# LinuxQMISDK

SLQS04.00.15

Generated by Doxygen 1.8.6

Fri Jun 22 2018 02:27:55

# 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** **35**

5.1 File List . . . 35

**6 Module Documentation** **39**

6.1 Device Connectivity Service (DCS) . . . 39

6.1.1 Detailed Description . . . 39

6.2 Wireless Data Service (WDS) . . . 40

6.2.1 Detailed Description . . . 40

6.3 Device Management Service (DMS) . . . 41

6.3.1 Detailed Description . . . 41

6.4 Network Access Service (NAS) . . . 42

6.4.1 Detailed Description . . . 42

6.5 CallBacK registration (CBK) . . . 43

6.5.1 Detailed Description . . . 43

6.6 Short Message Service (SMS) . . . 44

6.6.1 Detailed Description . . . 44

6.7 Position Determination Service (PDS) . . . 45
6.7.1 Detailed Description . . . 45
6.8 Card Application Toolkit (CAT) . . . 46
6.8.1 Detailed Description . . . 46
6.9 Remote Management Service (RMS) . . . 47
6.9.1 Detailed Description . . . 47
6.10 Firmware Management Service (FMS) . . . 48
6.10.1 Detailed Description . . . 48
6.11 Open Mobile Alliance Service (OMA) . . . 49
6.11.1 Detailed Description . . . 49
6.12 Specific Absorption Rate (SAR) . . . 50
6.12.1 Detailed Description . . . 50
6.13 SWI Open Mobile Alliance Service (SWIOMA) . . . 51
6.13.1 Detailed Description . . . 51
6.14 Voice Service (VOICE) . . . 52
6.14.1 Detailed Description . . . 52
6.15 Non-service specific APIs (SWI) . . . 53
6.15.1 Detailed Description . . . 53
6.16 User Identity Module Service (UIM) . . . 54
6.16.1 Detailed Description . . . 54
6.17 Audio Service (AUDIO) . . . 55
6.17.1 Detailed Description . . . 55
6.18 Quality of Service (QOS) . . . 56
6.18.1 Detailed Description . . . 56
6.19 IMS Service (IMS) . . . 57
6.19.1 Detailed Description . . . 57
6.20 SWI Audio Service(SWIAUDIO) . . . 58
6.20.1 Detailed Description . . . 58
6.21 Location Service(LOC) . . . 59
6.21.1 Detailed Description . . . 59
6.22 Thermal Mitigation Device(TMD) . . . 60
6.22.1 Detailed Description . . . 60
6.23 AVMS (SWIAVMS) . . . 61
6.23.1 Detailed Description . . . 61
6.24 SWIDMS . . . 62
6.24.1 Detailed Description . . . 62

**7 Namespace Documentation** **63**

7.1 Tables Namespace Reference . . . 63
7.1.1 Detailed Description . . . 63

**8 Data Structure Documentation** **65**

8.1 _getIndicationRegResp Struct Reference . . . 65
8.1.1 Detailed Description . . . 65
8.1.2 Field Documentation . . . 66
8.1.2.1 pRegCallStatInfoEvt . . . 66
8.1.2.2 pRegTransLayerInfoEvt . . . 66
8.1.2.3 pRegTransNWRegInfoEvt . . . 66
8.2 _GetProfileSettingIn Struct Reference . . . 66
8.2.1 Detailed Description . . . 66
8.2.2 Field Documentation . . . 66
8.2.2.1 ProfileID . . . 66
8.2.2.2 ProfileType . . . 66
8.3 _GetProfileSettingOut Struct Reference . . . 66
8.3.1 Detailed Description . . . 66
8.3.2 Field Documentation . . . 67
8.3.2.1 curProfile . . . 67
8.3.2.2 pExtErrCode . . . 67
8.4 _getResetInfoNotification Struct Reference . . . 67
8.4.1 Detailed Description . . . 67
8.4.2 Field Documentation . . . 68
8.4.2.1 source . . . 68
8.4.2.2 type . . . 68
8.5 _getTransLayerInfoResp Struct Reference . . . 68
8.5.1 Detailed Description . . . 68
8.5.2 Field Documentation . . . 68
8.5.2.1 pRegInd . . . 68
8.5.2.2 pTransLayerInfo . . . 68
8.6 _getTransNWRegInfoResp Struct Reference . . . 68
8.6.1 Detailed Description . . . 68
8.6.2 Field Documentation . . . 69
8.6.2.1 pRegStatus . . . 69
8.7 _MitigationDevInfo Struct Reference . . . 69
8.7.1 Detailed Description . . . 69
8.7.2 Field Documentation . . . 69
8.7.2.1 deviceID . . . 69
8.7.2.2 deviceIdLen . . . 69
8.8 _modemTempNotification Struct Reference . . . 69
8.8.1 Detailed Description . . . 70
8.8.2 Field Documentation . . . 70
8.8.2.1 ModemTemperature . . . 70

8.8.2.2 ModemTempState . . . 70
8.9 _packetSrvStatus Struct Reference . . . 70
8.9.1 Detailed Description . . . 70
8.9.2 Field Documentation . . . 72
8.9.2.1 bearerID . . . 72
8.9.2.2 connStatus . . . 72
8.9.2.3 ipFamily . . . 72
8.9.2.4 pQmiInterfaceInfo . . . 72
8.9.2.5 reconfigReqd . . . 72
8.9.2.6 sessionEndReason . . . 72
8.9.2.7 techName . . . 72
8.9.2.8 verboseSessnEndReason . . . 72
8.9.2.9 verboseSessnEndReasonType . . . 72
8.10 _qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . 72
8.10.1 Detailed Description . . . 72
8.10.2 Field Documentation . . . 73
8.10.2.1 activated_ind . . . 73
8.10.2.2 CDMABroadcastConfig . . . 73
8.10.2.3 num_instances . . . 73
8.11 _qaQmi3GPPBroadcastCfgInfo Struct Reference . . . 73
8.11.1 Detailed Description . . . 73
8.11.2 Field Documentation . . . 73
8.11.2.1 activated_ind . . . 73
8.11.2.2 broadcastConfig . . . 73
8.11.2.3 num_instances . . . 73
8.12 _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp Struct Reference . . . 73
8.12.1 Detailed Description . . . 74
8.12.2 Field Documentation . . . 75
8.12.2.1 bBinaryType . . . 75
8.12.2.2 bSerity . . . 75
8.12.2.3 bState . . . 75
8.12.2.4 bUserInputRequest . . . 75
8.12.2.5 szDescription . . . 75
8.12.2.6 szName . . . 75
8.12.2.7 szVersion . . . 75
8.12.2.8 TlvPresent . . . 75
8.12.2.9 ulPkgDownloadComplete . . . 75
8.12.2.10 ulPkgDownloadSize . . . 75
8.12.2.11 wDescriptionLength . . . 75
8.12.2.12 wNameLength . . . 75

8.12.2.13 wUpdateCompeteStatus . . . 75
8.12.2.14 wUserInputTimeout . . . 75
8.12.2.15 wVersionLength . . . 75
8.13 _QmiSwiAvmsEventReportConfigResp Struct Reference . . . 75
8.13.1 Detailed Description . . . 75
8.13.2 Field Documentation . . . 76
8.13.2.1 bState . . . 76
8.13.2.2 bUserInputRequest . . . 76
8.13.2.3 szAlertMsg . . . 76
8.13.2.4 TlvPresent . . . 76
8.13.2.5 wAlertMsgLength . . . 76
8.13.2.6 wUserInputTimeout . . . 76
8.14 _QmiSwiAvmsEventReportConnectionRequestResp Struct Reference . . . 76
8.14.1 Detailed Description . . . 76
8.14.2 Field Documentation . . . 77
8.14.2.1 bUserInputRequest . . . 77
8.14.2.2 TlvPresent . . . 77
8.14.2.3 wUserInputTimeout . . . 77
8.15 _QmiSwiAvmsEventReportDataSessionStatusResp Struct Reference . . . 77
8.15.1 Detailed Description . . . 77
8.15.2 Field Documentation . . . 77
8.15.2.1 bType . . . 77
8.15.2.2 TlvPresent . . . 77
8.15.2.3 wErrorCode . . . 77
8.16 _QmiSwiAvmsEventReportHTTPStatusResp Struct Reference . . . 77
8.16.1 Detailed Description . . . 77
8.16.2 Field Documentation . . . 78
8.16.2.1 TlvPresent . . . 78
8.16.2.2 wHTTPStatus . . . 79
8.17 _QmiSwiAvmsEventReportNotificationResp Struct Reference . . . 79
8.17.1 Detailed Description . . . 79
8.17.2 Field Documentation . . . 79
8.17.2.1 bNotification . . . 79
8.17.2.2 TlvPresent . . . 80
8.17.2.3 wSessionStatus . . . 80
8.18 _QmiSwiAvmsEventReportPackageIDResp Struct Reference . . . 80
8.18.1 Detailed Description . . . 80
8.18.2 Field Documentation . . . 80
8.18.2.1 bPackageID . . . 80
8.18.2.2 TlvPresent . . . 80

8.19 _QmiSwiAvmsEventReportRegStatusResp Struct Reference . . . 80
8.19.1 Detailed Description . . . 80
8.19.2 Field Documentation . . . 80
8.19.2.1 bRegStatus . . . 80
8.19.2.2 TlvPresent . . . 80
8.20 _QmiSwiAvmsEventReportSessionTypeResp Struct Reference . . . 80
8.20.1 Detailed Description . . . 81
8.20.2 Field Documentation . . . 81
8.20.2.1 bType . . . 81
8.20.2.2 TlvPresent . . . 81
8.21 _QmiSwiAvmsEventReportWAMSParamChangeResp Struct Reference . . . 81
8.21.1 Detailed Description . . . 81
8.21.2 Field Documentation . . . 81
8.21.2.1 TlvPresent . . . 81
8.21.2.2 wWamsChangeMask . . . 81
8.22 _QmiSwiAvmsSetSettingsAPNInfo Struct Reference . . . 81
8.22.1 Field Documentation . . . 82
8.22.1.1 bAPNLength . . . 82
8.22.1.2 bPWDLength . . . 82
8.22.1.3 bUnameLength . . . 82
8.22.1.4 szAPN . . . 82
8.22.1.5 szPWD . . . 82
8.22.1.6 szUname . . . 82
8.23 _QmiSwiAvmsSetSettingsConnectionRetryTimers Struct Reference . . . 82
8.23.1 Field Documentation . . . 82
8.23.1.1 Timers . . . 82
8.24 _QmiSwiAvmsSetSettingsPeriodInfo Struct Reference . . . 82
8.24.1 Field Documentation . . . 82
8.24.1.1 ulMax . . . 82
8.24.1.2 ulMin . . . 82
8.25 _setIndicationRegReq Struct Reference . . . 82
8.25.1 Detailed Description . . . 83
8.25.2 Field Documentation . . . 83
8.25.2.1 pRegCallStatInfoEvt . . . 83
8.25.2.2 pRegTransLayerInfoEvt . . . 83
8.25.2.3 pRegTransNWRegInfoEvt . . . 83
8.26 _slqs3GPPConfigItem Struct Reference . . . 83
8.26.1 Detailed Description . . . 84
8.26.2 Field Documentation . . . 85
8.26.2.1 LTEAttachProfileListLen . . . 85

8.26.2.2 p3gppRelease . . . 85
8.26.2.3 pDefaultPDNEnabled . . . 85
8.26.2.4 pLTEAttachProfile . . . 85
8.26.2.5 pLTEAttachProfileList . . . 85
8.26.2.6 pProfileList . . . 85
8.27 _SLQSAVMSSetSettingsReqPara Struct Reference . . . 85
8.27.1 Detailed Description . . . 85
8.27.2 Field Documentation . . . 87
8.27.2.1 pAPNInfo . . . 87
8.27.2.2 pAutoConnect . . . 87
8.27.2.3 pAutoReboot . . . 87
8.27.2.4 pConnectionRetryTimers . . . 87
8.27.2.5 pFwAutoSDM . . . 87
8.27.2.6 pFwPromptdownload . . . 87
8.27.2.7 pFwPromptUpdate . . . 87
8.27.2.8 pNotifStore . . . 87
8.27.2.9 pPeroidInfo . . . 87
8.27.2.10 pPollingTimer . . . 87
8.28 _SLQSAVMSSettings Struct Reference . . . 87
8.28.1 Field Documentation . . . 87
8.28.1.1 pAPNInfo . . . 87
8.28.1.2 pAutoConnect . . . 87
8.28.1.3 pAutoReboot . . . 87
8.28.1.4 pConnectionRetryTimers . . . 87
8.28.1.5 pFwAutoSDM . . . 87
8.28.1.6 pFwPromptdownload . . . 87
8.28.1.7 pFwPromptUpdate . . . 87
8.28.1.8 pNotificationStore . . . 88
8.28.1.9 pOMADMEnabled . . . 88
8.28.1.10 pPeroidsInfo . . . 88
8.28.1.11 pPollingTimer . . . 88
8.29 _SLQSAVMSSettingsAPNInfo Struct Reference . . . 88
8.29.1 Field Documentation . . . 88
8.29.1.1 bAPNLength . . . 88
8.29.1.2 bPWDLength . . . 88
8.29.1.3 bUnameLength . . . 88
8.29.1.4 pAPN . . . 88
8.29.1.5 pPWD . . . 88
8.29.1.6 pUname . . . 88
8.30 _SLQSAVMSSettingsConnectionRetryTimers Struct Reference . . . 88

8.30.1 Field Documentation . . . 88
8.30.1.1 Timers . . . 88
8.31 _SLQSAVMSSettingsPeriodsInfo Struct Reference . . . 88
8.31.1 Field Documentation . . . 89
8.31.1.1 max . . . 89
8.31.1.2 min . . . 89
8.32 _SlqsNas3GppNetworkRAT_ Struct Reference . . . 89
8.32.1 Detailed Description . . . 89
8.32.2 Field Documentation . . . 89
8.32.2.1 MCC . . . 89
8.32.2.2 MNC . . . 89
8.32.2.3 RAT . . . 89
8.33 _slqsNetworkScanInfo Struct Reference . . . 89
8.33.1 Detailed Description . . . 90
8.33.2 Field Documentation . . . 90
8.33.2.1 pNetworkInfo . . . 90
8.33.2.2 pNetworkInfoInstances . . . 90
8.33.2.3 pPCSDigitInfo . . . 90
8.33.2.4 pPCSDigitInstances . . . 90
8.33.2.5 pRATInfo . . . 90
8.33.2.6 pRATInstances . . . 90
8.33.2.7 pScanResult . . . 90
8.34 _SLQSOMADMSessionInfo Struct Reference . . . 91
8.34.1 Detailed Description . . . 91
8.34.2 Field Documentation . . . 93
8.34.2.1 pDate . . . 93
8.34.2.2 pDateLength . . . 93
8.34.2.3 pPkgDescLength . . . 93
8.34.2.4 pPkgDescription . . . 93
8.34.2.5 pPkgName . . . 93
8.34.2.6 pPkgNameLength . . . 93
8.34.2.7 pRetryCount . . . 93
8.34.2.8 pSessionState . . . 93
8.34.2.9 pSessionType . . . 93
8.34.2.10 pSeverity . . . 93
8.34.2.11 pSource . . . 93
8.34.2.12 pSourceLength . . . 93
8.34.2.13 pStatus . . . 93
8.34.2.14 pTime . . . 93
8.34.2.15 pTimeLength . . . 93

8.34.2.16 pUpdateCompleteStatus . . . 93
8.35 _SLQSOMADMSettings Struct Reference . . . 93
8.35.1 Detailed Description . . . 93
8.35.2 Field Documentation . . . 95
8.35.2.1 pAutosdm . . . 95
8.35.2.2 pFOTAdownload . . . 95
8.35.2.3 pFOTAUpdate . . . 95
8.35.2.4 pFwAutoCheck . . . 95
8.35.2.5 pOMADMEnabled . . . 95
8.36 _SLQSOMADMSettingsReqParams Struct Reference . . . 95
8.36.1 Detailed Description . . . 95
8.36.2 Field Documentation . . . 95
8.36.2.1 FOTAdownload . . . 96
8.36.2.2 FOTAUpdate . . . 96
8.36.2.3 pAutosdm . . . 96
8.37 _SLQSOMADMSettingsReqParams3 Struct Reference . . . 96
8.37.1 Detailed Description . . . 96
8.37.2 Field Documentation . . . 96
8.37.2.1 FOTAdownload . . . 97
8.37.2.2 FOTAUpdate . . . 97
8.37.2.3 pAutosdm . . . 97
8.37.2.4 pFwAutoCheck . . . 97
8.38 _SLQSSwiGetHostDevInfoParams Struct Reference . . . 97
8.38.1 Detailed Description . . . 97
8.38.2 Field Documentation . . . 98
8.38.2.1 bManSize . . . 98
8.38.2.2 bModelSize . . . 98
8.38.2.3 bPlasmaIDSize . . . 98
8.38.2.4 bSWVerSize . . . 98
8.38.2.5 pManString . . . 98
8.38.2.6 pModelString . . . 98
8.38.2.7 pPlasmaIDString . . . 98
8.38.2.8 pSWVerString . . . 98
8.39 _SLQSSwiGetOSInfoParams Struct Reference . . . 98
8.39.1 Detailed Description . . . 98
8.39.2 Field Documentation . . . 98
8.39.2.1 bNameSize . . . 98
8.39.2.2 bVersionSize . . . 98
8.39.2.3 pNameString . . . 99
8.39.2.4 pVersionString . . . 99

8.40 _SLQSSwiGetSerialNoExtParams Struct Reference . . . 99
8.40.1 Detailed Description . . . 99
8.40.2 Field Documentation . . . 99
8.40.2.1 meidLength . . . 99
8.40.2.2 pMeidString . . . 99
8.41 _SLQSSwiSetHostDevInfoParams Struct Reference . . . 99
8.41.1 Detailed Description . . . 99
8.41.2 Field Documentation . . . 100
8.41.2.1 bManSize . . . 100
8.41.2.2 bModelSize . . . 100
8.41.2.3 bPlasmaIDSize . . . 100
8.41.2.4 bSWVerSize . . . 100
8.41.2.5 pManString . . . 100
8.41.2.6 pModelString . . . 100
8.41.2.7 pPlasmaIDString . . . 100
8.41.2.8 pSWVerString . . . 100
8.42 _SLQSSwiSetOSInfoParams Struct Reference . . . 100
8.42.1 Detailed Description . . . 101
8.42.2 Field Documentation . . . 101
8.42.2.1 bNameSize . . . 101
8.42.2.2 bVersionSize . . . 101
8.42.2.3 pNameString . . . 101
8.42.2.4 pVersionString . . . 101
8.43 _SwiAvmsEventInd Struct Reference . . . 101
8.43.1 Field Documentation . . . 101
8.43.1.1 BinaryUpdateSessionInfo . . . 101
8.43.1.2 Config . . . 101
8.43.1.3 ConnectionRequest . . . 101
8.43.1.4 DataSessionStatus . . . 102
8.43.1.5 HTTPStatus . . . 102
8.43.1.6 Notification . . . 102
8.43.1.7 PackageID . . . 102
8.43.1.8 RegStatus . . . 102
8.43.1.9 SessionType . . . 102
8.43.1.10 WAMSParaChanged . . . 102
8.44 _sysSelectPrefInfo Struct Reference . . . 102
8.44.1 Detailed Description . . . 102
8.44.2 Field Documentation . . . 106
8.44.2.1 pBandPref . . . 106
8.44.2.2 pEmerMode . . . 106

8.44.2.3 pGWAcqOrderPref . . . 106
8.44.2.4 pLTEBandPref . . . 106
8.44.2.5 pModePref . . . 106
8.44.2.6 pNetSelPref . . . 106
8.44.2.7 pPRLPref . . . 106
8.44.2.8 pRoamPref . . . 106
8.44.2.9 pSrvDomainPref . . . 106
8.45 _sysSelectPrefParams Struct Reference . . . 107
8.45.1 Detailed Description . . . 107
8.45.2 Field Documentation . . . 112
8.45.2.1 pAcqOrderPref . . . 112
8.45.2.2 pBandPref . . . 112
8.45.2.3 pChgDuration . . . 112
8.45.2.4 pCSGID . . . 112
8.45.2.5 pEmerMode . . . 112
8.45.2.6 pGWAcqOrderPref . . . 112
8.45.2.7 pLTEBandPref . . . 112
8.45.2.8 pMNCIncPCSDigStat . . . 112
8.45.2.9 pModePref . . . 112
8.45.2.10 pNetSelPref . . . 112
8.45.2.11 pPRLPref . . . 113
8.45.2.12 pRAT . . . 113
8.45.2.13 pRoamPref . . . 113
8.45.2.14 pSrvDomainPref . . . 113
8.45.2.15 pSrvRegRestriction . . . 113
8.45.2.16 pTdscdmaBandPref . . . 113
8.46 _transLayerinfo Struct Reference . . . 113
8.46.1 Detailed Description . . . 113
8.46.2 Field Documentation . . . 113
8.46.2.1 TransCap . . . 113
8.46.2.2 TransType . . . 113
8.47 _transLayerInfoNotification Struct Reference . . . 113
8.47.1 Detailed Description . . . 113
8.47.2 Field Documentation . . . 114
8.47.2.1 pTransLayerInfo . . . 114
8.47.2.2 regInd . . . 114
8.48 _transNWRegInfoNotification Struct Reference . . . 114
8.48.1 Detailed Description . . . 114
8.48.2 Field Documentation . . . 114
8.48.2.1 NWRegStat . . . 114

8.49 accelAcceptReady_s Struct Reference . . . 115
8.49.1 Detailed Description . . . 115
8.49.2 Field Documentation . . . 115
8.49.2.1 batchPerSec . . . 115
8.49.2.2 injectEnable . . . 115
8.49.2.3 samplesPerBatch . . . 115
8.50 accelTempAcceptReady_s Struct Reference . . . 115
8.50.1 Detailed Description . . . 115
8.50.2 Field Documentation . . . 116
8.50.2.1 batchPerSec . . . 116
8.50.2.2 injectEnable . . . 116
8.50.2.3 samplesPerBatch . . . 116
8.51 acqOrderPref Struct Reference . . . 116
8.51.1 Detailed Description . . . 116
8.51.2 Field Documentation . . . 117
8.51.2.1 acqOrdeLen . . . 117
8.51.2.2 pAcqOrder . . . 117
8.52 ActPilotPNElement Struct Reference . . . 117
8.52.1 Detailed Description . . . 117
8.52.2 Field Documentation . . . 117
8.52.2.1 ActSetPilotPN . . . 117
8.52.2.2 ActSetPilotPNStrength . . . 117
8.53 AddCDMASysInfo Struct Reference . . . 117
8.53.1 Detailed Description . . . 117
8.53.2 Field Documentation . . . 118
8.53.2.1 geoSysIdx . . . 118
8.53.2.2 regPrd . . . 118
8.54 AddSysInfo Struct Reference . . . 118
8.54.1 Detailed Description . . . 118
8.54.2 Field Documentation . . . 118
8.54.2.1 cellBroadcastCap . . . 118
8.54.2.2 geoSysIdx . . . 118
8.55 airTimer Struct Reference . . . 118
8.55.1 Detailed Description . . . 118
8.55.2 Field Documentation . . . 119
8.55.2.1 airTimerValue . . . 119
8.55.2.2 namID . . . 119
8.56 allCallsAlphaIDInfo Struct Reference . . . 119
8.56.1 Detailed Description . . . 119
8.56.2 Field Documentation . . . 119

8.56.2.1 AlphaIDInfo . . . 119
8.56.2.2 callID . . . 119
8.57 allCallsDiagInfo Struct Reference . . . 119
8.57.1 Detailed Description . . . 120
8.57.2 Field Documentation . . . 120
8.57.2.1 callID . . . 120
8.57.2.2 DiagInfo . . . 120
8.58 allCallsUUSInfo Struct Reference . . . 120
8.58.1 Detailed Description . . . 120
8.58.2 Field Documentation . . . 120
8.58.2.1 callID . . . 120
8.58.2.2 uusInfo . . . 120
8.59 alphaIDInfo Struct Reference . . . 120
8.59.1 Detailed Description . . . 121
8.59.2 Field Documentation . . . 121
8.59.2.1 alphaDcs . . . 121
8.59.2.2 alphaLen . . . 121
8.59.2.3 alphaText . . . 121
8.60 altitudeSrcInfo Struct Reference . . . 121
8.60.1 Detailed Description . . . 121
8.60.2 Field Documentation . . . 122
8.60.2.1 coverage . . . 122
8.60.2.2 linkage . . . 122
8.60.2.3 source . . . 122
8.61 altSrcInfo_t Struct Reference . . . 122
8.61.1 Detailed Description . . . 122
8.61.2 Field Documentation . . . 123
8.61.2.1 coverage . . . 123
8.61.2.2 linkage . . . 123
8.61.2.3 source . . . 123
8.62 appStats Struct Reference . . . 123
8.62.1 Detailed Description . . . 124
8.62.2 Field Documentation . . . 126
8.62.2.1 aidLength . . . 126
8.62.2.2 aidVal . . . 126
8.62.2.3 appState . . . 126
8.62.2.4 appType . . . 126
8.62.2.5 persoFeature . . . 126
8.62.2.6 persoRetries . . . 126
8.62.2.7 persoState . . . 126

8.62.2.8 persoUnblockRetries . . . 126
8.62.2.9 pin1Retries . . . 126
8.62.2.10 pin1State . . . 126
8.62.2.11 pin2Retries . . . 126
8.62.2.12 pin2State . . . 126
8.62.2.13 puk1Retries . . . 126
8.62.2.14 puk2Retries . . . 126
8.62.2.15 univPin . . . 126
8.63 appStatus Struct Reference . . . 126
8.63.1 Detailed Description . . . 127
8.63.2 Field Documentation . . . 129
8.63.2.1 aidLength . . . 129
8.63.2.2 aidVal . . . 129
8.63.2.3 appState . . . 129
8.63.2.4 appType . . . 129
8.63.2.5 persoFeature . . . 129
8.63.2.6 persoRetries . . . 129
8.63.2.7 persoState . . . 129
8.63.2.8 persoUnblockRetries . . . 129
8.63.2.9 pin1Retries . . . 129
8.63.2.10 pin1State . . . 129
8.63.2.11 pin2Retries . . . 129
8.63.2.12 pin2State . . . 129
8.63.2.13 puk1Retries . . . 129
8.63.2.14 puk2Retries . . . 129
8.63.2.15 univPin . . . 129
8.64 arrAlertingPattern Struct Reference . . . 129
8.64.1 Detailed Description . . . 130
8.64.2 Field Documentation . . . 130
8.64.2.1 alertingPattern . . . 130
8.64.2.2 callID . . . 130
8.64.2.3 numInstances . . . 130
8.65 arrAlertingType Struct Reference . . . 130
8.65.1 Detailed Description . . . 130
8.65.2 Field Documentation . . . 131
8.65.2.1 AlertingType . . . 131
8.65.2.2 callID . . . 131
8.65.2.3 numInstances . . . 131
8.66 arrAlphaID Struct Reference . . . 131
8.66.1 Detailed Description . . . 131

8.66.2 Field Documentation . . . 131
8.66.2.1 allCallsAlphaIDInfoArr . . . 131
8.66.2.2 numInstances . . . 131
8.67 arrCalledPartyNum Struct Reference . . . 131
8.67.1 Detailed Description . . . 132
8.67.2 Field Documentation . . . 132
8.67.2.1 CalledPartyNum . . . 132
8.67.2.2 numInstances . . . 132
8.68 arrCallEndReason Struct Reference . . . 132
8.68.1 Detailed Description . . . 132
8.68.2 Field Documentation . . . 133
8.68.2.1 callEndReason . . . 133
8.68.2.2 callID . . . 133
8.68.2.3 numInstances . . . 133
8.69 arrCallInfo Struct Reference . . . 133
8.69.1 Detailed Description . . . 133
8.69.2 Field Documentation . . . 133
8.69.2.1 getAllCallInfo . . . 133
8.69.2.2 numInstances . . . 133
8.70 arrConnectPartyNum Struct Reference . . . 133
8.70.1 Detailed Description . . . 133
8.70.2 Field Documentation . . . 134
8.70.2.1 ConnectedPartyNum . . . 134
8.70.2.2 numInstances . . . 134
8.71 arrDiagInfo Struct Reference . . . 134
8.71.1 Detailed Description . . . 134
8.71.2 Field Documentation . . . 134
8.71.2.1 DiagInfo . . . 134
8.71.2.2 numInstances . . . 134
8.72 arrRedirPartyNum Struct Reference . . . 134
8.72.1 Detailed Description . . . 134
8.72.2 Field Documentation . . . 135
8.72.2.1 numInstances . . . 135
8.72.2.2 RedirPartyNum . . . 135
8.73 arrRemotePartyName Struct Reference . . . 135
8.73.1 Detailed Description . . . 135
8.73.2 Field Documentation . . . 135
8.73.2.1 GetAllCallRmtPtyName . . . 135
8.73.2.2 numInstances . . . 135
8.74 arrRemotePartyNum Struct Reference . . . 135

8.74.1 Detailed Description . . . 136
8.74.2 Field Documentation . . . 136
8.74.2.1 numInstances . . . 136
8.74.2.2 RmtPtyNum . . . 136
8.75 arrSvcOption Struct Reference . . . 136
8.75.1 Detailed Description . . . 136
8.75.2 Field Documentation . . . 136
8.75.2.1 callID . . . 136
8.75.2.2 numInstances . . . 136
8.75.2.3 srvOption . . . 137
8.76 arrUUSInfo Struct Reference . . . 137
8.76.1 Detailed Description . . . 137
8.76.2 Field Documentation . . . 137
8.76.2.1 AllCallsUUSInfo . . . 137
8.76.2.2 numInstances . . . 137
8.77 authenticateResult Struct Reference . . . 137
8.77.1 Detailed Description . . . 137
8.77.2 Field Documentation . . . 138
8.77.2.1 content . . . 138
8.77.2.2 contentLen . . . 138
8.78 authenticationData Struct Reference . . . 138
8.78.1 Detailed Description . . . 138
8.78.2 Field Documentation . . . 138
8.78.2.1 context . . . 139
8.78.2.2 data . . . 139
8.78.2.3 dataLen . . . 139
8.79 BandCapabilityResp Struct Reference . . . 139
8.79.1 Detailed Description . . . 139
8.79.2 Field Documentation . . . 142
8.79.2.1 bandCapability . . . 142
8.79.2.2 pLteBandCapability . . . 142
8.79.2.3 pTdsBandCapability . . . 142
8.80 BdsSV Struct Reference . . . 142
8.80.1 Detailed Description . . . 142
8.80.2 Field Documentation . . . 142
8.80.2.1 id . . . 142
8.80.2.2 mask . . . 142
8.81 BdsSVInfo Struct Reference . . . 142
8.81.1 Detailed Description . . . 143
8.81.2 Field Documentation . . . 143

8.81.2.1 len . . . 143
8.81.2.2 pSV . . . 143
8.82 BroadcastConfig Struct Reference . . . 143
8.82.1 Detailed Description . . . 143
8.82.2 Field Documentation . . . 144
8.82.2.1 fromServiceId . . . 144
8.82.2.2 selected . . . 144
8.82.2.3 toServiceId . . . 144
8.83 burstDTMFInfo Struct Reference . . . 144
8.83.1 Detailed Description . . . 144
8.83.2 Field Documentation . . . 144
8.83.2.1 digitCnt . . . 144
8.83.2.2 pCallID . . . 144
8.83.2.3 pDigitBuff . . . 144
8.84 CallBarringSysInfo Struct Reference . . . 144
8.84.1 Detailed Description . . . 145
8.84.2 Field Documentation . . . 145
8.84.2.1 csBarStatus . . . 145
8.84.2.2 psBarStatus . . . 145
8.85 callBarStatus Struct Reference . . . 145
8.85.1 Detailed Description . . . 145
8.85.2 Field Documentation . . . 146
8.85.2.1 csBarStatus . . . 146
8.85.2.2 psBarStatus . . . 146
8.86 calledPartyInfo Struct Reference . . . 146
8.86.1 Detailed Description . . . 146
8.86.2 Field Documentation . . . 147
8.86.2.1 number . . . 147
8.86.2.2 numLen . . . 147
8.86.2.3 numPlan . . . 147
8.86.2.4 numType . . . 148
8.86.2.5 PI . . . 148
8.86.2.6 SI . . . 148
8.87 calledPartySubAdd Struct Reference . . . 148
8.87.1 Detailed Description . . . 148
8.87.2 Field Documentation . . . 148
8.87.2.1 extBit . . . 148
8.87.2.2 oddEvenInd . . . 148
8.87.2.3 subAddr . . . 148
8.87.2.4 subAddrLen . . . 148

8.87.2.5 subAddrType . . . 149
8.88 callerIDInfo Struct Reference . . . 149
8.88.1 Detailed Description . . . 149
8.88.2 Field Documentation . . . 149
8.88.2.1 callerID . . . 149
8.88.2.2 callerIDLen . . . 149
8.88.2.3 PI . . . 149
8.89 callFwdTypeAndPlan Struct Reference . . . 149
8.89.1 Detailed Description . . . 149
8.89.2 Field Documentation . . . 150
8.89.2.1 numberPlan . . . 150
8.89.2.2 numberType . . . 150
8.90 callFWExtInfo Struct Reference . . . 150
8.90.1 Detailed Description . . . 150
8.90.2 Field Documentation . . . 152
8.90.2.1 noReplyTimer . . . 152
8.90.2.2 number . . . 152
8.90.2.3 numLen . . . 152
8.90.2.4 numPlan . . . 152
8.90.2.5 numType . . . 152
8.90.2.6 PI . . . 152
8.90.2.7 SI . . . 152
8.90.2.8 SvcClass . . . 152
8.90.2.9 SvcStatus . . . 152
8.91 callFWInfo Struct Reference . . . 152
8.91.1 Detailed Description . . . 152
8.91.2 Field Documentation . . . 153
8.91.2.1 noReplyTimer . . . 153
8.91.2.2 number . . . 153
8.91.2.3 numLen . . . 153
8.91.2.4 SvcClass . . . 153
8.91.2.5 SvcStatus . . . 153
8.92 callInfo Struct Reference . . . 153
8.92.1 Detailed Description . . . 153
8.92.2 Field Documentation . . . 154
8.92.2.1 callID . . . 154
8.92.2.2 callState . . . 155
8.92.2.3 callType . . . 155
8.92.2.4 direction . . . 155
8.92.2.5 mode . . . 155

8.93 callingPartyInfo Struct Reference . . . 155
8.93.1 Detailed Description . . . 155
8.93.2 Field Documentation . . . 156
8.93.2.1 number . . . 156
8.93.2.2 numLen . . . 156
8.93.2.3 numPlan . . . 156
8.93.2.4 numType . . . 156
8.93.2.5 PI . . . 156
8.93.2.6 SI . . . 156
8.94 cardResult Struct Reference . . . 156
8.94.1 Detailed Description . . . 157
8.94.2 Field Documentation . . . 157
8.94.2.1 sw1 . . . 157
8.94.2.2 sw2 . . . 157
8.95 cardStatus Struct Reference . . . 157
8.95.1 Detailed Description . . . 157
8.95.2 Field Documentation . . . 158
8.95.2.1 index1xPri . . . 158
8.95.2.2 index1xSec . . . 158
8.95.2.3 indexGwPri . . . 158
8.95.2.4 indexGwSec . . . 158
8.95.2.5 numSlot . . . 158
8.95.2.6 SlotInfo . . . 158
8.96 CarrierImage_t Struct Reference . . . 158
8.96.1 Detailed Description . . . 158
8.96.2 Field Documentation . . . 159
8.96.2.1 m_FwBuildId . . . 159
8.96.2.2 m_FwImageId . . . 159
8.96.2.3 m_nCarrierId . . . 159
8.96.2.4 m_nFolderId . . . 159
8.96.2.5 m_nStorage . . . 159
8.96.2.6 m_PriBuildId . . . 159
8.96.2.7 m_PriImageId . . . 159
8.97 CatAlPhaIdentifierTlv Struct Reference . . . 159
8.97.1 Detailed Description . . . 160
8.97.2 Field Documentation . . . 160
8.97.2.1 AlphaID . . . 160
8.97.2.2 AlphaIDLength . . . 160
8.97.2.3 ReferenceID . . . 160
8.98 CatCommonEventTlv Struct Reference . . . 160

8.98.1 Field Documentation . . . 160
8.98.1.1 CatEvent . . . 160
8.98.1.2 EventID . . . 160
8.98.1.3 EventLength . . . 160
8.98.1.4 TlvPresent . . . 160
8.99 CatEndProactiveSessionTlv Struct Reference . . . 160
8.99.1 Detailed Description . . . 160
8.99.2 Field Documentation . . . 161
8.99.2.1 EndProactiveSession . . . 161
8.100 CATEventDataType Struct Reference . . . 161
8.100.1 Field Documentation . . . 161
8.100.1.1 eventMask . . . 161
8.100.1.2 pErrorMask . . . 161
8.101 CatEventIDDataTlv Struct Reference . . . 161
8.101.1 Detailed Description . . . 161
8.101.2 Field Documentation . . . 161
8.101.2.1 Data . . . 161
8.101.2.2 DataLength . . . 161
8.101.2.3 ReferenceID . . . 161
8.102 CatEventListTlv Struct Reference . . . 161
8.102.1 Detailed Description . . . 162
8.102.2 Field Documentation . . . 162
8.102.2.1 SetupEventList . . . 162
8.103 CatRefreshTlv Struct Reference . . . 162
8.103.1 Detailed Description . . . 162
8.103.2 Field Documentation . . . 162
8.103.2.1 RefreshMode . . . 162
8.103.2.2 RefreshStage . . . 162
8.104 ccSUPSType Struct Reference . . . 162
8.104.1 Detailed Description . . . 163
8.104.2 Field Documentation . . . 163
8.104.2.1 reason . . . 163
8.104.2.2 svcType . . . 163
8.105 CDMABroadcastConfig Struct Reference . . . 163
8.105.1 Detailed Description . . . 163
8.105.2 Field Documentation . . . 164
8.105.2.1 language . . . 164
8.105.2.2 selected . . . 164
8.105.2.3 serviceCategory . . . 164
8.106 CDMAChannel Struct Reference . . . 164

8.106.1 Detailed Description . . . 164
8.106.2 Field Documentation . . . 164
8.106.2.1 priChA . . . 164
8.106.2.2 priChB . . . 164
8.106.2.3 secChA . . . 165
8.106.2.4 secChB . . . 165
8.107 CDMAECIOThresh Struct Reference . . . 165
8.107.1 Detailed Description . . . 165
8.107.2 Field Documentation . . . 165
8.107.2.1 CDMAECIOThreshListLen . . . 165
8.107.2.2 pCDMAECIOThreshList . . . 165
8.108 CDMAInfo Struct Reference . . . 165
8.108.1 Detailed Description . . . 165
8.108.2 Field Documentation . . . 166
8.108.2.1 baseId . . . 166
8.108.2.2 baseLat . . . 166
8.108.2.3 baseLong . . . 166
8.108.2.4 nid . . . 166
8.108.2.5 refpn . . . 166
8.108.2.6 sid . . . 166
8.109 cdmaMsgDecodingParams Struct Reference . . . 166
8.109.1 Detailed Description . . . 167
8.109.2 Field Documentation . . . 168
8.109.2.1 absoluteValidity . . . 168
8.109.2.2 mcTimeStamp . . . 168
8.109.2.3 messageLength . . . 168
8.109.2.4 pAlertPriority . . . 168
8.109.2.5 pCallbkAddr . . . 168
8.109.2.6 pCallbkAddrLength . . . 168
8.109.2.7 pDisplayMode . . . 168
8.109.2.8 pLanguage . . . 169
8.109.2.9 pMessage . . . 169
8.109.2.10 pMessageID . . . 169
8.109.2.11 pPriority . . . 169
8.109.2.12 pPrivacy . . . 169
8.109.2.13 pReadAcknowledgementReq . . . 169
8.109.2.14 pRelativeValidity . . . 169
8.109.2.15 pSenderAddr . . . 169
8.109.2.16 pSenderAddrLength . . . 169
8.109.2.17 pTextMsg . . . 169

8.109.2.18 pTextMsgLength . . . 169
8.109.2.19 pUserAcknowledgementReq . . . 169
8.110 cdmaMsgDecodingParamsExt Struct Reference . . . 169
8.110.1 Detailed Description . . . 170
8.110.2 Field Documentation . . . 171
8.110.2.1 absoluteValidity . . . 171
8.110.2.2 mcTimeStamp . . . 171
8.110.2.3 messageLength . . . 172
8.110.2.4 pAlertPriority . . . 172
8.110.2.5 pCallbkAddr . . . 172
8.110.2.6 pCallbkAddrLength . . . 172
8.110.2.7 pDisplayMode . . . 172
8.110.2.8 pIsUDHPresent . . . 172
8.110.2.9 pLanguage . . . 172
8.110.2.10 pMessage . . . 172
8.110.2.11 pMessageID . . . 172
8.110.2.12 pPartNum . . . 172
8.110.2.13 pPriority . . . 172
8.110.2.14 pPrivacy . . . 172
8.110.2.15 pReadAcknowledgementReq . . . 172
8.110.2.16 pReferenceNum . . . 172
8.110.2.17 pRelativeValidity . . . 172
8.110.2.18 pSenderAddr . . . 172
8.110.2.19 pSenderAddrLength . . . 172
8.110.2.20 pTextMsg . . . 172
8.110.2.21 pTextMsgLength . . . 172
8.110.2.22 pTotalNum . . . 172
8.110.2.23 pUserAcknowledgementReq . . . 172
8.111 cdmaMsgEncodingParams Struct Reference . . . 172
8.111.1 Detailed Description . . . 173
8.111.2 Field Documentation . . . 174
8.111.2.1 messageId . . . 174
8.111.2.2 pCallbackAddr . . . 174
8.111.2.3 pDestAddr . . . 174
8.111.2.4 pEncodingAlphabet . . . 174
8.111.2.5 pMessage . . . 174
8.111.2.6 pMessageSize . . . 174
8.111.2.7 pPriority . . . 174
8.111.2.8 pRelValidity . . . 174
8.111.2.9 pTextMsg . . . 174

8.111.2.10 textMsgLength . . . 174
8.112 CDMARSSIThresh Struct Reference . . . 174
8.112.1 Detailed Description . . . 174
8.112.2 Field Documentation . . . 174
8.112.2.1 CDMARSSIThreshListLen . . . 174
8.112.2.2 pCDMARSSIThreshList . . . 174
8.113 CDMASSInfo Struct Reference . . . 174
8.113.1 Detailed Description . . . 175
8.113.2 Field Documentation . . . 175
8.113.2.1 ecio . . . 175
8.113.2.2 rssi . . . 175
8.114 cdmaSSInfo Struct Reference . . . 175
8.114.1 Detailed Description . . . 175
8.114.2 Field Documentation . . . 175
8.114.2.1 ecio . . . 175
8.114.2.2 rssi . . . 176
8.115 CDMASysInfo Struct Reference . . . 176
8.115.1 Detailed Description . . . 176
8.115.2 Field Documentation . . . 178
8.115.2.1 baseId . . . 178
8.115.2.2 baseLat . . . 179
8.115.2.3 baseLong . . . 179
8.115.2.4 bsInfoValid . . . 179
8.115.2.5 bsPRev . . . 179
8.115.2.6 bsPRevValid . . . 179
8.115.2.7 ccsSupported . . . 179
8.115.2.8 ccsSupportedValid . . . 179
8.115.2.9 cdmaSysIdValid . . . 179
8.115.2.10 isSysPrlMatch . . . 179
8.115.2.11 isSysPrlMatchValid . . . 179
8.115.2.12 MCC . . . 179
8.115.2.13 MNC . . . 179
8.115.2.14 networkID . . . 179
8.115.2.15 networkIdValid . . . 179
8.115.2.16 packetZone . . . 179
8.115.2.17 packetZoneValid . . . 179
8.115.2.18 pRevInUse . . . 179
8.115.2.19 pRevInUseValid . . . 179
8.115.2.20 sysInfoCDMA . . . 179
8.115.2.21 systemID . . . 179

8.116 CDMASysInfoExt Struct Reference . . . 179
8.116.1 Detailed Description . . . 179
8.116.2 Field Documentation . . . 180
8.116.2.1 imsi_11_12 . . . 180
8.116.2.2 MCC . . . 180
8.117 CellDb Struct Reference . . . 180
8.117.1 Detailed Description . . . 180
8.117.2 Field Documentation . . . 180
8.117.2.1 mask . . . 180
8.118 cellParams Struct Reference . . . 180
8.118.1 Detailed Description . . . 181
8.118.2 Field Documentation . . . 181
8.118.2.1 pci . . . 181
8.118.2.2 rsrp . . . 181
8.118.2.3 rsrq . . . 181
8.118.2.4 rssi . . . 181
8.118.2.5 srxlev . . . 181
8.119 changeUIMPIN Struct Reference . . . 181
8.119.1 Detailed Description . . . 182
8.119.2 Field Documentation . . . 182
8.119.2.1 oldPINLen . . . 182
8.119.2.2 oldPINVal . . . 182
8.119.2.3 pinID . . . 182
8.119.2.4 pinLen . . . 182
8.119.2.5 pinVal . . . 182
8.120 ChannelRate Struct Reference . . . 182
8.120.1 Detailed Description . . . 182
8.120.2 Field Documentation . . . 183
8.120.2.1 CurrChanRxRate . . . 183
8.120.2.2 CurrChanTxRate . . . 183
8.120.2.3 MaxChanRxRate . . . 183
8.120.2.4 MaxChanTxRate . . . 183
8.121 channelRate Struct Reference . . . 183
8.121.1 Detailed Description . . . 183
8.121.2 Field Documentation . . . 183
8.121.2.1 CurrChanRxRate . . . 183
8.121.2.2 CurrChanTxRate . . . 183
8.122 ciotAcqOrderPref Struct Reference . . . 183
8.122.1 Detailed Description . . . 184
8.122.2 Field Documentation . . . 184

8.122.2.1 ciotAcqOrderLen . . . 184
8.122.2.2 pCiotAcqOrder . . . 184
8.123 CLIPResp Struct Reference . . . 184
8.123.1 Detailed Description . . . 184
8.123.2 Field Documentation . . . 185
8.123.2.1 ActiveStatus . . . 185
8.123.2.2 ProvisionStatus . . . 185
8.124 CLIRResp Struct Reference . . . 185
8.124.1 Detailed Description . . . 185
8.124.2 Field Documentation . . . 186
8.124.2.1 ActiveStatus . . . 186
8.124.2.2 ProvisionStatus . . . 186
8.125 ClkInfo Struct Reference . . . 186
8.125.1 Detailed Description . . . 186
8.125.2 Field Documentation . . . 187
8.125.2.1 mask . . . 187
8.126 CNAPResp Struct Reference . . . 187
8.126.1 Detailed Description . . . 188
8.126.2 Field Documentation . . . 188
8.126.2.1 ActiveStatus . . . 188
8.126.2.2 ProvisionStatus . . . 188
8.127 COLPResp Struct Reference . . . 188
8.127.1 Detailed Description . . . 188
8.127.2 Field Documentation . . . 189
8.127.2.1 ActiveStatus . . . 189
8.127.2.2 ProvisionStatus . . . 189
8.128 COLRResp Struct Reference . . . 189
8.128.1 Detailed Description . . . 189
8.128.2 Field Documentation . . . 189
8.128.2.1 ActiveStatus . . . 189
8.128.2.2 ProvisionStatus . . . 189
8.129 CommInfo Struct Reference . . . 189
8.129.1 Detailed Description . . . 190
8.129.2 Field Documentation . . . 191
8.129.2.1 imsRegState . . . 191
8.129.2.2 modemMode . . . 191
8.129.2.3 psState . . . 191
8.129.2.4 systemMode . . . 191
8.129.2.5 temperature . . . 191
8.130 ConnectionStatus Struct Reference . . . 191

8.130.1 Detailed Description . . . 191
8.130.2 Field Documentation . . . 192
8.130.2.1 MDMCallDuration . . . 192
8.130.2.2 MDMConnStatus . . . 192
8.131 connectionStatus Struct Reference . . . 192
8.131.1 Detailed Description . . . 192
8.131.2 Field Documentation . . . 192
8.131.2.1 MDMCallDuration . . . 193
8.131.2.2 MDMConnStatus . . . 193
8.132 connectNumInfo Struct Reference . . . 193
8.132.1 Detailed Description . . . 193
8.132.2 Field Documentation . . . 194
8.132.2.1 callerID . . . 194
8.132.2.2 callerIDLen . . . 194
8.132.2.3 numPlan . . . 194
8.132.2.4 numPresInd . . . 194
8.132.2.5 numType . . . 194
8.132.2.6 screeningInd . . . 194
8.133 CrashInfo Struct Reference . . . 194
8.133.1 Detailed Description . . . 194
8.133.2 Field Documentation . . . 195
8.133.2.1 crashData . . . 195
8.133.2.2 crashId . . . 195
8.133.2.3 crashStrLen . . . 195
8.133.2.4 gcDumpStrLen . . . 195
8.133.2.5 numCrashes . . . 195
8.133.2.6 pCrashString . . . 195
8.133.2.7 pGCDumpString . . . 195
8.134 CrashInfoParams Struct Reference . . . 195
8.134.1 Detailed Description . . . 195
8.134.2 Field Documentation . . . 196
8.134.2.1 pCrashInfo . . . 196
8.134.2.2 pDevCrashStatus . . . 196
8.135 crashInfoParams Struct Reference . . . 196
8.135.1 Detailed Description . . . 196
8.135.2 Field Documentation . . . 196
8.135.2.1 crashInfo . . . 196
8.135.2.2 crashStatus . . . 196
8.136 crashInformation Struct Reference . . . 196
8.136.1 Detailed Description . . . 197

8.136.2 Field Documentation . . . 197
8.136.2.1 crashData . . . 197
8.136.2.2 crashId . . . 197
8.136.2.3 crashString . . . 197
8.136.2.4 crashStrlen . . . 197
8.136.2.5 gcdumpString . . . 197
8.136.2.6 gcdumpStrlen . . . 197
8.136.2.7 numCrashes . . . 197
8.137 CreateProfileIn Struct Reference . . . 197
8.137.1 Detailed Description . . . 198
8.137.2 Field Documentation . . . 198
8.137.2.1 curProfile . . . 198
8.137.2.2 pProfileID . . . 198
8.137.2.3 pProfileType . . . 198
8.138 CreateProfileOut Struct Reference . . . 198
8.138.1 Detailed Description . . . 198
8.138.2 Field Documentation . . . 199
8.138.2.1 pExtErrorCode . . . 199
8.138.2.2 pProfileIndex . . . 199
8.138.2.3 pProfileType . . . 199
8.139 CsgId Struct Reference . . . 199
8.139.1 Detailed Description . . . 199
8.139.2 Field Documentation . . . 199
8.139.2.1 csgId . . . 199
8.139.2.2 TlvPresent . . . 199
8.140 CSGID Struct Reference . . . 199
8.140.1 Detailed Description . . . 200
8.140.2 Field Documentation . . . 200
8.140.2.1 id . . . 200
8.140.2.2 mcc . . . 200
8.140.2.3 mnc . . . 200
8.140.2.4 mncPcsDigits . . . 200
8.140.2.5 rat . . . 200
8.141 CUGInfo Struct Reference . . . 200
8.141.1 Detailed Description . . . 201
8.141.2 Field Documentation . . . 201
8.141.2.1 CUGIndex . . . 201
8.141.2.2 SuppOA . . . 201
8.141.2.3 SuppPrefCUG . . . 201
8.142 curAMRConfig Struct Reference . . . 201

8.142.1 Detailed Description . . . 201
8.142.2 Field Documentation . . . 202
8.142.2.1 gsmAmrStat . . . 202
8.142.2.2 wcdmaAmrStat . . . 202
8.143 CurrDataSysStat Struct Reference . . . 202
8.143.1 Detailed Description . . . 202
8.143.2 Field Documentation . . . 203
8.143.2.1 pCurrNetworkInfo . . . 203
8.143.2.2 pNetworkInfoLen . . . 203
8.143.2.3 pPrefNetwork . . . 203
8.144 currentCatEvent Union Reference . . . 203
8.144.1 Detailed Description . . . 203
8.144.2 Field Documentation . . . 203
8.144.2.1 CatAlphaIdtfr . . . 203
8.144.2.2 CatEndPS . . . 203
8.144.2.3 CatEventLst . . . 203
8.144.2.4 CatEvIDData . . . 203
8.144.2.5 CatRefresh . . . 203
8.145 CurrentImgList Struct Reference . . . 203
8.145.1 Detailed Description . . . 204
8.145.2 Field Documentation . . . 204
8.145.2.1 carrier . . . 204
8.145.2.2 fwvers . . . 204
8.145.2.3 numEntries . . . 204
8.145.2.4 pCurrImgInfo . . . 204
8.145.2.5 pkgver . . . 204
8.145.2.6 priver . . . 204
8.146 currentPLMN Struct Reference . . . 204
8.146.1 Detailed Description . . . 205
8.146.2 Field Documentation . . . 205
8.146.2.1 MCC . . . 205
8.146.2.2 MNC . . . 205
8.146.2.3 netDescr . . . 205
8.146.2.4 netDescrLength . . . 205
8.147 CurrImageInfo Struct Reference . . . 205
8.147.1 Detailed Description . . . 206
8.147.2 Field Documentation . . . 206
8.147.2.1 buildID . . . 206
8.147.2.2 buildIDLen . . . 206
8.147.2.3 imageType . . . 206

8.147.2.4 uniqueID . . . 206
8.148 CurrNetworkInfo Struct Reference . . . 206
8.148.1 Detailed Description . . . 206
8.148.2 Field Documentation . . . 207
8.148.2.1 NetworkType . . . 208
8.148.2.2 RATMask . . . 208
8.148.2.3 SOMask . . . 208
8.149 currNetworkInfo Struct Reference . . . 208
8.149.1 Detailed Description . . . 208
8.149.2 Field Documentation . . . 209
8.149.2.1 NetworkType . . . 209
8.149.2.2 RATMask . . . 209
8.149.2.3 SOMask . . . 209
8.150 custFeaturesInfo Struct Reference . . . 209
8.150.1 Detailed Description . . . 209
8.150.2 Field Documentation . . . 211
8.150.2.1 GpsEnable . . . 211
8.150.2.2 pDHCPRelayEnabled . . . 211
8.150.2.3 pDisableIMSI . . . 211
8.150.2.4 pGPSLPM . . . 211
8.150.2.5 pGPSSel . . . 211
8.150.2.6 pIPFamSupport . . . 211
8.150.2.7 pIsVoiceEnabled . . . 211
8.150.2.8 pRMAutoConnect . . . 211
8.150.2.9 pSMSSupport . . . 212
8.151 custFeaturesSetting Struct Reference . . . 212
8.151.1 Detailed Description . . . 212
8.151.2 Field Documentation . . . 213
8.151.2.1 pDHCPRelayEnabled . . . 213
8.151.2.2 pGPSEnable . . . 213
8.151.2.3 pGPSLPM . . . 213
8.151.2.4 pGPSSel . . . 213
8.151.2.5 pIsVoiceEnabled . . . 213
8.152 custSettingInfo Struct Reference . . . 213
8.152.1 Detailed Description . . . 213
8.152.2 Field Documentation . . . 214
8.152.2.1 cust_attr . . . 214
8.152.2.2 cust_id . . . 214
8.152.2.3 cust_value . . . 214
8.152.2.4 id_length . . . 214

8.152.2.5 value_length . . . 214
8.153 custSettingList Struct Reference . . . 214
8.153.1 Detailed Description . . . 214
8.153.2 Field Documentation . . . 214
8.153.2.1 custSetting . . . 214
8.153.2.2 list_type . . . 215
8.153.2.3 num_instances . . . 215
8.154 dataBearers Struct Reference . . . 215
8.154.1 Detailed Description . . . 215
8.154.2 Field Documentation . . . 215
8.154.2.1 dataBearerMask . . . 215
8.154.2.2 pCurDataBearerTechnology . . . 215
8.154.2.3 pLastCallDataBearerTechnology . . . 215
8.155 DataBearerTech Struct Reference . . . 215
8.155.1 Detailed Description . . . 215
8.155.2 Field Documentation . . . 217
8.155.2.1 ratValue . . . 217
8.155.2.2 soMask . . . 217
8.155.2.3 techType . . . 217
8.156 DataBearerTechExt Struct Reference . . . 217
8.156.1 Detailed Description . . . 218
8.156.2 Field Documentation . . . 218
8.156.2.1 pBearerTech . . . 218
8.156.2.2 pLastBearerTech . . . 218
8.157 dataBearerTechnology Struct Reference . . . 218
8.157.1 Detailed Description . . . 218
8.157.2 Field Documentation . . . 219
8.157.2.1 currentNetwork . . . 219
8.157.2.2 ratMask . . . 219
8.157.2.3 soMask . . . 219
8.158 dataRate Struct Reference . . . 219
8.158.1 Detailed Description . . . 219
8.158.2 Field Documentation . . . 220
8.158.2.1 dataRateMax . . . 220
8.158.2.2 guaranteedRate . . . 220
8.159 dataSrvCapabilities Struct Reference . . . 220
8.159.1 Detailed Description . . . 220
8.159.2 Field Documentation . . . 220
8.159.2.1 dataCapabilities . . . 220
8.159.2.2 dataCapabilitiesLen . . . 221

8.160 DataStatusDetail Struct Reference . . . 221
8.160.1 Detailed Description . . . 221
8.160.2 Field Documentation . . . 222
8.160.2.1 IPAddress . . . 222
8.160.2.2 LastErrCode . . . 222
8.161 DataULongLongTlv Struct Reference . . . 223
8.161.1 Field Documentation . . . 223
8.161.1.1 TlvPresent . . . 223
8.161.1.2 ullData . . . 223
8.162 DataULongTlv Struct Reference . . . 223
8.162.1 Field Documentation . . . 223
8.162.1.1 TlvPresent . . . 223
8.162.1.2 ulData . . . 223
8.163 DcsUsbPortNames Struct Reference . . . 223
8.163.1 Field Documentation . . . 223
8.163.1.1 AtCmdPort . . . 223
8.163.1.2 DmPort . . . 223
8.163.1.3 NmeaPort . . . 223
8.164 delAssistDataStatus Struct Reference . . . 223
8.164.1 Detailed Description . . . 224
8.164.2 Field Documentation . . . 224
8.164.2.1 status . . . 224
8.165 depersonalizationInformation Struct Reference . . . 224
8.165.1 Detailed Description . . . 224
8.165.2 Field Documentation . . . 225
8.165.2.1 ckLen . . . 225
8.165.2.2 ckVal . . . 225
8.165.2.3 feature . . . 225
8.165.2.4 operation . . . 225
8.166 detailSvcInfo Struct Reference . . . 225
8.166.1 Detailed Description . . . 225
8.166.2 Field Documentation . . . 226
8.166.2.1 hdrHybrid . . . 226
8.166.2.2 hdrSrvStatus . . . 226
8.166.2.3 isSysForbidden . . . 226
8.166.2.4 srvCapability . . . 227
8.166.2.5 srvStatus . . . 227
8.167 DeviceConfigDetail Struct Reference . . . 227
8.167.1 Detailed Description . . . 227
8.167.2 Field Documentation . . . 227

8.167.2.1 Chipset . . . 227
8.167.2.2 HWVersion . . . 228
8.167.2.3 QLIC . . . 228
8.167.2.4 Technology . . . 228
8.168 DHCPOption Struct Reference . . . 228
8.168.1 Detailed Description . . . 228
8.168.2 Field Documentation . . . 228
8.168.2.1 optCode . . . 228
8.168.2.2 optValLen . . . 228
8.168.2.3 pOptVal . . . 228
8.169 DHCPOptionList Struct Reference . . . 228
8.169.1 Detailed Description . . . 228
8.169.2 Field Documentation . . . 229
8.169.2.1 numOpt . . . 229
8.169.2.2 pOptions . . . 229
8.170 diagInfo Struct Reference . . . 229
8.170.1 Detailed Description . . . 229
8.170.2 Field Documentation . . . 229
8.170.2.1 diagInfoLen . . . 229
8.170.2.2 diagnosticInfo . . . 229
8.171 dirNum Struct Reference . . . 229
8.171.1 Detailed Description . . . 230
8.171.2 Field Documentation . . . 230
8.171.2.1 dirNum . . . 230
8.171.2.2 dirNumLen . . . 230
8.172 dms_ActivationStatusTlv Struct Reference . . . 230
8.172.1 Detailed Description . . . 230
8.172.2 Field Documentation . . . 231
8.172.2.1 activationStatus . . . 231
8.172.2.2 TlvPresent . . . 231
8.173 dms_OperatingModeTlv Struct Reference . . . 231
8.173.1 Detailed Description . . . 231
8.173.2 Field Documentation . . . 231
8.173.2.1 operatingMode . . . 231
8.173.2.2 TlvPresent . . . 231
8.174 dms_PSMActiveTimerIndTlv Struct Reference . . . 231
8.174.1 Detailed Description . . . 231
8.174.2 Field Documentation . . . 232
8.174.2.1 ActiveTimerInd . . . 232
8.174.2.2 TlvPresent . . . 232

8.175dms_PSMActiveTimerTlv Struct Reference . . . 232
8.175.1 Detailed Description . . . 232
8.175.2 Field Documentation . . . 232
8.175.2.1 activeTimer . . . 232
8.175.2.2 TlvPresent . . . 232
8.176dms_PSMDurationDueToOOSTlv Struct Reference . . . 232
8.176.1 Detailed Description . . . 232
8.176.2 Field Documentation . . . 233
8.176.2.1 durationDueToOOS . . . 233
8.176.2.2 TlvPresent . . . 233
8.177dms_PSMDurationThresholdTlv Struct Reference . . . 233
8.177.1 Detailed Description . . . 233
8.177.2 Field Documentation . . . 233
8.177.2.1 durationThreshold . . . 233
8.177.2.2 TlvPresent . . . 233
8.178dms_PSMEarlyWakeupTimeTlv Struct Reference . . . 233
8.178.1 Detailed Description . . . 233
8.178.2 Field Documentation . . . 234
8.178.2.1 earlyWakeupTime . . . 234
8.178.2.2 TlvPresent . . . 234
8.179dms_PSMEnableStateIndTlv Struct Reference . . . 234
8.179.1 Detailed Description . . . 234
8.179.2 Field Documentation . . . 234
8.179.2.1 EnableStateInd . . . 234
8.179.2.2 TlvPresent . . . 234
8.180dms_PSMEnableStateTlv Struct Reference . . . 234
8.180.1 Detailed Description . . . 235
8.180.2 Field Documentation . . . 235
8.180.2.1 enableState . . . 235
8.180.2.2 TlvPresent . . . 235
8.181dms_PSMPeriodicUpdateTimerIndTlv Struct Reference . . . 235
8.181.1 Detailed Description . . . 235
8.181.2 Field Documentation . . . 235
8.181.2.1 PeriodicUpdateTimerInd . . . 235
8.181.2.2 TlvPresent . . . 235
8.182dms_PSMPeriodicUpdateTimerTlv Struct Reference . . . 235
8.182.1 Detailed Description . . . 236
8.182.2 Field Documentation . . . 236
8.182.2.1 periodicUpdateTimer . . . 236
8.182.2.2 TlvPresent . . . 236

8.183 dms_PSMRandomizationWindowTlv Struct Reference . . . 236
8.183.1 Detailed Description . . . 236
8.183.2 Field Documentation . . . 236
8.183.2.1 randomizationWindow . . . 236
8.183.2.2 TlvPresent . . . 236
8.184 dms_TemperatureTlv Struct Reference . . . 236
8.184.1 Detailed Description . . . 237
8.184.2 Field Documentation . . . 237
8.184.2.1 Temperature . . . 237
8.184.2.2 TempStat . . . 237
8.184.2.3 TlvPresent . . . 237
8.185 dms_VoltageTlv Struct Reference . . . 237
8.185.1 Detailed Description . . . 237
8.185.2 Field Documentation . . . 238
8.185.2.1 TlvPresent . . . 238
8.185.2.2 Voltage . . . 238
8.185.2.3 VoltStat . . . 238
8.186 dmsCurrentPRLInfo Struct Reference . . . 238
8.186.1 Detailed Description . . . 238
8.186.2 Field Documentation . . . 238
8.186.2.1 pPRLPreference . . . 238
8.186.2.2 pPRLVersion . . . 238
8.187 DMScustSettingInfo Struct Reference . . . 238
8.187.1 Detailed Description . . . 239
8.187.2 Field Documentation . . . 239
8.187.2.1 cust_attr . . . 239
8.187.2.2 cust_id . . . 239
8.187.2.3 cust_value . . . 239
8.187.2.4 id_length . . . 239
8.187.2.5 value_length . . . 239
8.188 DMScustSettingList Struct Reference . . . 239
8.188.1 Detailed Description . . . 239
8.188.2 Field Documentation . . . 240
8.188.2.1 custSetting . . . 240
8.188.2.2 list_type . . . 240
8.188.2.3 num_instances . . . 240
8.189 DMSgetCustomFeatureV2 Struct Reference . . . 240
8.189.1 Detailed Description . . . 240
8.189.2 Field Documentation . . . 240
8.189.2.1 pCustSettingInfo . . . 240

8.189.2.2 pCustSettingList . . . 240
8.189.2.3 pGetCustomInput . . . 240
8.190 DMSgetCustomInput Struct Reference . . . 241
8.190.1 Detailed Description . . . 241
8.190.2 Field Documentation . . . 241
8.190.2.1 cust_id . . . 241
8.190.2.2 list_type . . . 241
8.191 dmsIndicationRegisterReq Struct Reference . . . 241
8.191.1 Detailed Description . . . 241
8.191.2 Field Documentation . . . 241
8.191.2.1 pSwiGetResetInd . . . 242
8.192 dmsPsmCfgParamChange Struct Reference . . . 242
8.192.1 Detailed Description . . . 242
8.192.2 Field Documentation . . . 242
8.192.2.1 pActiveTimer . . . 242
8.192.2.2 pEnableState . . . 242
8.192.2.3 pPeriodicUpdateTimer . . . 242
8.193 dmsSwiGetPCInfo Struct Reference . . . 242
8.193.1 Detailed Description . . . 242
8.193.2 Field Documentation . . . 243
8.193.2.1 opMode . . . 244
8.193.2.2 pLpmFlag . . . 244
8.193.2.3 pPersistentLpm . . . 244
8.193.2.4 pPowerOffMode . . . 244
8.193.2.5 pWdisable . . . 244
8.194 dmsSwiGetResetInfo Struct Reference . . . 244
8.194.1 Detailed Description . . . 244
8.194.2 Field Documentation . . . 244
8.194.2.1 source . . . 244
8.194.2.2 type . . . 244
8.195 Domain Struct Reference . . . 244
8.195.1 Detailed Description . . . 245
8.195.2 Field Documentation . . . 245
8.195.2.1 domainLen . . . 245
8.195.2.2 domainName . . . 245
8.196 DomainNameList Struct Reference . . . 245
8.196.1 Detailed Description . . . 245
8.196.2 Field Documentation . . . 245
8.196.2.1 domain . . . 245
8.196.2.2 numInstances . . . 245

8.197 DRCParams Struct Reference . . . 245
8.197.1 Detailed Description . . . 246
8.197.2 Field Documentation . . . 246
8.197.2.1 DRCCover . . . 246
8.197.2.2 DRCValue . . . 246
8.198 DTMFInfo Struct Reference . . . 246
8.198.1 Detailed Description . . . 246
8.198.2 Field Documentation . . . 247
8.198.2.1 callID . . . 247
8.198.2.2 digitBuff . . . 247
8.198.2.3 digitCnt . . . 247
8.198.2.4 DTMFEvent . . . 247
8.199 DTMFLengths Struct Reference . . . 247
8.199.1 Detailed Description . . . 247
8.199.2 Field Documentation . . . 247
8.199.2.1 DTMFInterdigitInterval . . . 248
8.199.2.2 DTMFPulseWidth . . . 248
8.200 DUNCallInfoInd Struct Reference . . . 248
8.200.1 Field Documentation . . . 248
8.200.1.1 CallEndReason . . . 248
8.200.1.2 ChannelRate . . . 248
8.200.1.3 DataBearerTech . . . 248
8.200.1.4 DormancyStatus . . . 248
8.200.1.5 MdmConnStatus . . . 248
8.200.1.6 RXOKBytesCount . . . 248
8.200.1.7 TXOKBytesCount . . . 248
8.201 dunchannelRate Struct Reference . . . 248
8.201.1 Detailed Description . . . 248
8.201.2 Field Documentation . . . 249
8.201.2.1 CurrChanRxRate . . . 249
8.201.2.2 CurrChanTxRate . . . 249
8.201.2.3 MaxChanRxRate . . . 249
8.201.2.4 MaxChanTxRate . . . 249
8.202 ecioListElement Struct Reference . . . 249
8.202.1 Detailed Description . . . 249
8.202.2 Field Documentation . . . 249
8.202.2.1 ecio . . . 250
8.202.2.2 radioIf . . . 250
8.203 ECIOThresh Struct Reference . . . 250
8.203.1 Detailed Description . . . 250

8.203.2 Field Documentation . . . 250
8.203.2.1 ECIOThresListLen . . . 250
8.203.2.2 pECIOThresList . . . 250
8.204 ECTNum Struct Reference . . . 250
8.204.1 Detailed Description . . . 250
8.204.2 Field Documentation . . . 251
8.204.2.1 ECTCallState . . . 251
8.204.2.2 number . . . 251
8.204.2.3 presentationInd . . . 251
8.205 ehrpdMTUSizeTlv Struct Reference . . . 251
8.205.1 Detailed Description . . . 251
8.205.2 Field Documentation . . . 251
8.205.2.1 ehrpdMTUSize . . . 251
8.205.2.2 TlvPresent . . . 251
8.206 encryptedPIN1 Struct Reference . . . 252
8.206.1 Detailed Description . . . 252
8.206.2 Field Documentation . . . 252
8.206.2.1 pin1Len . . . 252
8.206.2.2 pin1Val . . . 252
8.207 eriDataparams Struct Reference . . . 252
8.207.1 Detailed Description . . . 252
8.207.2 Field Documentation . . . 253
8.207.2.1 eriData . . . 253
8.207.2.2 eriDataLen . . . 253
8.208 ERIFileparams Struct Reference . . . 253
8.208.1 Detailed Description . . . 253
8.208.2 Field Documentation . . . 253
8.208.2.1 pFile . . . 253
8.208.2.2 pFileSize . . . 253
8.209 errorRateListElement Struct Reference . . . 253
8.209.1 Detailed Description . . . 253
8.209.2 Field Documentation . . . 254
8.209.2.1 errorRate . . . 254
8.209.2.2 radioIf . . . 254
8.210 eTWSPLMNInfoTlv Struct Reference . . . 254
8.210.1 Detailed Description . . . 254
8.210.2 Field Documentation . . . 255
8.210.2.1 ETWSPLMNInfo . . . 255
8.210.2.2 TlvPresent . . . 255
8.211 extDispRecInfo Struct Reference . . . 255

8.211.1 Detailed Description . . . 255
8.211.2 Field Documentation . . . 255
8.211.2.1 dispType . . . 255
8.211.2.2 extDispInfo . . . 255
8.211.2.3 extDispInfoLen . . . 255
8.212 FactorySequenceNumber Struct Reference . . . 255
8.212.1 Detailed Description . . . 256
8.212.2 Field Documentation . . . 256
8.212.2.1 FSNumber . . . 256
8.213 fileAttributes Struct Reference . . . 256
8.213.1 Detailed Description . . . 256
8.213.2 Field Documentation . . . 259
8.213.2.1 fileID . . . 259
8.213.2.2 fileSize . . . 259
8.213.2.3 fileType . . . 259
8.213.2.4 rawLen . . . 259
8.213.2.5 rawValue . . . 259
8.213.2.6 recordCount . . . 259
8.213.2.7 recordSize . . . 259
8.213.2.8 secActivate . . . 259
8.213.2.9 secActivateMask . . . 259
8.213.2.10 secDeactivate . . . 259
8.213.2.11 secDeactivateMask . . . 259
8.213.2.12 secIncrease . . . 259
8.213.2.13 secIncreaseMask . . . 259
8.213.2.14 secRead . . . 259
8.213.2.15 secReadMask . . . 259
8.213.2.16 secWrite . . . 259
8.213.2.17 secWriteMask . . . 259
8.214 fileInfo Struct Reference . . . 259
8.214.1 Detailed Description . . . 259
8.214.2 Field Documentation . . . 260
8.214.2.1 fileID . . . 260
8.214.2.2 path . . . 260
8.214.2.3 pathLen . . . 260
8.215 FirmwareUpdatStat Struct Reference . . . 260
8.215.1 Detailed Description . . . 260
8.215.2 Field Documentation . . . 261
8.215.2.1 pImgType . . . 261
8.215.2.2 pLogString . . . 262

8.215.2.3 pLogStringLen . . . 262
8.215.2.4 pRefData . . . 262
8.215.2.5 pRefString . . . 262
8.215.2.6 pRefStringLen . . . 262
8.215.2.7 ResCode . . . 262
8.216 FMSImageElement Struct Reference . . . 262
8.216.1 Detailed Description . . . 262
8.216.2 Field Documentation . . . 262
8.216.2.1 buildId . . . 262
8.216.2.2 buildIdLength . . . 262
8.216.2.3 imageId . . . 262
8.216.2.4 imageType . . . 262
8.217 FMSImageIdElement Struct Reference . . . 262
8.217.1 Detailed Description . . . 263
8.217.2 Field Documentation . . . 263
8.217.2.1 buildID . . . 263
8.217.2.2 buildIDLength . . . 263
8.217.2.3 failureCount . . . 263
8.217.2.4 imageID . . . 263
8.217.2.5 storageIndex . . . 263
8.218 FMSImageIDEntries Struct Reference . . . 263
8.218.1 Detailed Description . . . 264
8.218.2 Field Documentation . . . 264
8.218.2.1 executingImage . . . 264
8.218.2.2 imageIDElement . . . 264
8.218.2.3 imageIDSize . . . 264
8.218.2.4 imageType . . . 264
8.218.2.5 maxImages . . . 264
8.219 FMSImageList Struct Reference . . . 264
8.219.1 Detailed Description . . . 264
8.219.2 Field Documentation . . . 265
8.219.2.1 imageIDEntries . . . 265
8.219.2.2 listSize . . . 265
8.220 FMSPrefImageList Struct Reference . . . 265
8.220.1 Detailed Description . . . 265
8.220.2 Field Documentation . . . 265
8.220.2.1 listEntries . . . 265
8.220.2.2 listSize . . . 265
8.221 fwinfo_s Struct Reference . . . 265
8.221.1 Detailed Description . . . 265

8.221.2 Field Documentation . . . 266
8.221.2.1 Carrier . . . 266
8.221.2.2 FirmwareID . . . 266
8.221.2.3 GPSCapability . . . 266
8.221.2.4 Region . . . 266
8.221.2.5 Technology . . . 266
8.222GERANInfo Struct Reference . . . 266
8.222.1 Detailed Description . . . 266
8.222.2 Field Documentation . . . 267
8.222.2.1 arfcn . . . 267
8.222.2.2 bsic . . . 267
8.222.2.3 cellID . . . 267
8.222.2.4 insNmrCellInfo . . . 268
8.222.2.5 lac . . . 268
8.222.2.6 nmrInst . . . 268
8.222.2.7 plmn . . . 268
8.222.2.8 rxLev . . . 268
8.222.2.9 timingAdvance . . . 268
8.223geranInstInfo Struct Reference . . . 268
8.223.1 Detailed Description . . . 268
8.223.2 Field Documentation . . . 268
8.223.2.1 geranArfcn . . . 268
8.223.2.2 geranBsicBcc . . . 268
8.223.2.3 geranBsicNcc . . . 268
8.223.2.4 geranRssi . . . 268
8.224getAllCallInformation Struct Reference . . . 269
8.224.1 Detailed Description . . . 269
8.224.2 Field Documentation . . . 269
8.224.2.1 ALS . . . 269
8.224.2.2 Callinfo . . . 269
8.224.2.3 isEmpty . . . 269
8.225getAllCallRmtPtyName Struct Reference . . . 269
8.225.1 Detailed Description . . . 269
8.225.2 Field Documentation . . . 270
8.225.2.1 callID . . . 270
8.225.2.2 RemotePartyName . . . 270
8.226getAllCallRmtPtyNum Struct Reference . . . 270
8.226.1 Detailed Description . . . 270
8.226.2 Field Documentation . . . 270
8.226.2.1 callID . . . 270

8.226.2.2 RemotePartyNum . . . 270
8.227 GetAudioPathConfigReq Struct Reference . . . 270
8.227.1 Detailed Description . . . 270
8.227.2 Field Documentation . . . 271
8.227.2.1 Item . . . 271
8.227.2.2 Profile . . . 271
8.228 GetAudioPathConfigResp Struct Reference . . . 271
8.228.1 Detailed Description . . . 271
8.228.2 Field Documentation . . . 272
8.228.2.1 pCodecSTGain . . . 273
8.228.2.2 pDTMFTXGain . . . 273
8.228.2.3 pECMode . . . 273
8.228.2.4 pMICGainSelect . . . 273
8.228.2.5 pNSEnable . . . 273
8.228.2.6 pRXAGCList . . . 273
8.228.2.7 pRXAVCAGCSwitch . . . 273
8.228.2.8 pRXAVCList . . . 273
8.228.2.9 pRXPCMIIRFltr . . . 273
8.228.2.10 pTXAGCList . . . 273
8.228.2.11 pTXAVCSwitch . . . 273
8.228.2.12 pTXGain . . . 273
8.228.2.13 pTXPCMIIRFltr . . . 273
8.229 GetAudioProfileReq Struct Reference . . . 273
8.229.1 Detailed Description . . . 273
8.229.2 Field Documentation . . . 273
8.229.2.1 Generator . . . 273
8.230 GetAudioProfileResp Struct Reference . . . 273
8.230.1 Detailed Description . . . 274
8.230.2 Field Documentation . . . 274
8.230.2.1 EarMute . . . 274
8.230.2.2 MicMute . . . 274
8.230.2.3 Profile . . . 274
8.230.2.4 Volume . . . 274
8.231 GetAudioVolTLBConfigReq Struct Reference . . . 275
8.231.1 Detailed Description . . . 275
8.231.2 Field Documentation . . . 275
8.231.2.1 Generator . . . 275
8.231.2.2 Item . . . 275
8.231.2.3 Profile . . . 275
8.231.2.4 Volume . . . 275

8.232GetAudioVolTLBConfigResp Struct Reference . . . 275
8.232.1 Detailed Description . . . 276
8.232.2 Field Documentation . . . 276
8.232.2.1 ResCode . . . 276
8.233getCallFWExtInfo Struct Reference . . . 276
8.233.1 Detailed Description . . . 276
8.233.2 Field Documentation . . . 276
8.233.2.1 CallFWExtInfo . . . 276
8.233.2.2 numInstances . . . 276
8.234getCallFWInfo Struct Reference . . . 276
8.234.1 Detailed Description . . . 276
8.234.2 Field Documentation . . . 277
8.234.2.1 CallFWInfo . . . 277
8.234.2.2 numInstances . . . 277
8.235getCustomFeatureV2 Struct Reference . . . 277
8.235.1 Detailed Description . . . 277
8.235.2 Field Documentation . . . 277
8.235.2.1 pCustSettingInfo . . . 277
8.235.2.2 pCustSettingList . . . 277
8.235.2.3 pGetCustomInput . . . 277
8.236getCustomInput Struct Reference . . . 278
8.236.1 Detailed Description . . . 278
8.236.2 Field Documentation . . . 278
8.236.2.1 cust_id . . . 278
8.236.2.2 list_type . . . 278
8.237getDUNCallInfoReq Struct Reference . . . 278
8.237.1 Detailed Description . . . 278
8.237.2 Field Documentation . . . 279
8.237.2.1 Mask . . . 279
8.237.2.2 pReportChannelRate . . . 279
8.237.2.3 pReportConnStatus . . . 279
8.237.2.4 pReportDataBearerTech . . . 279
8.237.2.5 pReportDormStatus . . . 279
8.237.2.6 pTransferStatInd . . . 279
8.238getDUNCallInfoResp Struct Reference . . . 280
8.238.1 Detailed Description . . . 280
8.238.2 Field Documentation . . . 282
8.238.2.1 pCallEndReason . . . 282
8.238.2.2 pChannelRate . . . 282
8.238.2.3 pConnectionStatus . . . 282

8.238.2.4 pDataBearerTech . . . 282
8.238.2.5 pDormancyStatus . . . 282
8.238.2.6 pLastCallDataBearerTech . . . 282
8.238.2.7 pLastCallRXOKBytesCnt . . . 282
8.238.2.8 pLastCallTXOKBytesCnt . . . 282
8.238.2.9 pMdmCallDurationActive . . . 282
8.238.2.10 pRXOKBytesCount . . . 282
8.238.2.11 pTXOKBytesCount . . . 282
8.239 getDyingGaspCfg Struct Reference . . . 282
8.239.1 Detailed Description . . . 283
8.239.2 Field Documentation . . . 283
8.239.2.1 pDestSMSContent . . . 283
8.239.2.2 pDestSMSNum . . . 283
8.240 getDyingGaspStatistics Struct Reference . . . 283
8.240.1 Detailed Description . . . 283
8.240.2 Field Documentation . . . 283
8.240.2.1 pSMSAttemptedFlag . . . 284
8.240.2.2 pTimeStamp . . . 284
8.241 GetErrRateResp Struct Reference . . . 284
8.241.1 Detailed Description . . . 284
8.241.2 Field Documentation . . . 284
8.241.2.1 pCDMAFrameErrRate . . . 284
8.241.2.2 pGSMBER . . . 284
8.241.2.3 pHDRPackErrRate . . . 284
8.241.2.4 pWCDMABER . . . 284
8.242 GetHRPDStatsResp Struct Reference . . . 285
8.242.1 Detailed Description . . . 285
8.242.2 Field Documentation . . . 285
8.242.2.1 pDRCParams . . . 285
8.242.2.2 pPilotSetData . . . 285
8.242.2.3 pUATI . . . 285
8.243 GetIMSSMSConfigParams Struct Reference . . . 285
8.243.1 Detailed Description . . . 285
8.243.2 Field Documentation . . . 286
8.243.2.1 pPhoneCtxtURI . . . 286
8.243.2.2 pPhoneCtxtURILen . . . 286
8.243.2.3 pSettingResp . . . 286
8.243.2.4 pSMSFormat . . . 286
8.243.2.5 pSMSOverIPNwInd . . . 286
8.244 GetIMSUserConfigParams Struct Reference . . . 286

8.244.1 Detailed Description . . . 286
8.244.2 Field Documentation . . . 287
8.244.2.1 pIMSDomain . . . 287
8.244.2.2 pIMSDomainLen . . . 287
8.244.2.3 pSettingResp . . . 287
8.245 GetIMSVoIPConfigResp Struct Reference . . . 287
8.245.1 Detailed Description . . . 287
8.245.2 Field Documentation . . . 289
8.245.2.1 pAmrMode . . . 289
8.245.2.2 pAmrOctetAligned . . . 289
8.245.2.3 pAmrWbEnable . . . 289
8.245.2.4 pAmrWBMode . . . 289
8.245.2.5 pAmrWBOctetAligned . . . 289
8.245.2.6 pMinSessionExpiryTimer . . . 289
8.245.2.7 pRingBackTimer . . . 289
8.245.2.8 pRingingTimer . . . 289
8.245.2.9 pRTPRTCPInactTimer . . . 289
8.245.2.10 pScrAmrEnable . . . 289
8.245.2.11 pScrAmrWbEnable . . . 289
8.245.2.12 pSessionExpiryTimer . . . 289
8.245.2.13 pSettingResp . . . 289
8.246 GetInstIDResp Struct Reference . . . 289
8.246.1 Field Documentation . . . 289
8.246.1.1 pInstanceID . . . 289
8.246.1.2 pIPFamily . . . 290
8.247 GetM2MAudioProfileReq Struct Reference . . . 290
8.247.1 Detailed Description . . . 290
8.247.2 Field Documentation . . . 290
8.247.2.1 pGenerator . . . 290
8.248 GetM2MAudioProfileResp Struct Reference . . . 290
8.248.1 Detailed Description . . . 290
8.248.2 Field Documentation . . . 291
8.248.2.1 CwtMute . . . 291
8.248.2.2 EarMute . . . 291
8.248.2.3 Generator . . . 291
8.248.2.4 MicMute . . . 291
8.248.2.5 Profile . . . 291
8.248.2.6 Volume . . . 291
8.249 GetM2MAudioVolumeReq Struct Reference . . . 291
8.249.1 Detailed Description . . . 291

8.249.2 Field Documentation . . . 292
8.249.2.1 Generator . . . 292
8.249.2.2 Profile . . . 292
8.250 GetM2MAudioVolumeResp Struct Reference . . . 292
8.250.1 Detailed Description . . . 292
8.250.2 Field Documentation . . . 292
8.250.2.1 Level . . . 292
8.251 GetM2MAVMuteReq Struct Reference . . . 292
8.251.1 Detailed Description . . . 292
8.251.2 Field Documentation . . . 292
8.251.2.1 Profile . . . 293
8.252 GetM2MAVMuteResp Struct Reference . . . 293
8.252.1 Detailed Description . . . 293
8.252.2 Field Documentation . . . 293
8.252.2.1 CwtMute . . . 293
8.252.2.2 EarMute . . . 293
8.252.2.3 MicMute . . . 293
8.253 GetM2MSpkrGainReq Struct Reference . . . 293
8.253.1 Detailed Description . . . 293
8.253.2 Field Documentation . . . 294
8.253.2.1 Profile . . . 294
8.254 GetM2MSpkrGainResp Struct Reference . . . 294
8.254.1 Detailed Description . . . 294
8.254.2 Field Documentation . . . 294
8.254.2.1 Value . . . 294
8.255 getMsgWaitingInfo Struct Reference . . . 294
8.255.1 Detailed Description . . . 294
8.255.2 Field Documentation . . . 295
8.255.2.1 msgWaitInfo . . . 295
8.255.2.2 numInstances . . . 295
8.256 GetNetworkTimeResp Struct Reference . . . 295
8.256.1 Detailed Description . . . 295
8.256.2 Field Documentation . . . 295
8.256.2.1 p3GPP2TimeInfo . . . 295
8.256.2.2 p3GPPTimeInfo . . . 295
8.257 GetRegMgrConfigParams Struct Reference . . . 295
8.257.1 Detailed Description . . . 295
8.257.2 Field Documentation . . . 296
8.257.2.1 pIMSTestMode . . . 296
8.257.2.2 pPCSCFPort . . . 296

8.257.2.3 pPriCSCFPortName . . . 296
8.257.2.4 pPriCSCFPortNameLen . . . 296
8.257.2.5 pSettingResp . . . 296
8.258 GetSessionIDResp Struct Reference . . . 296
8.258.1 Field Documentation . . . 296
8.258.1.1 pSessionIDv4 . . . 296
8.258.1.2 pSessionIDv6 . . . 296
8.259 GetSIPConfigResp Struct Reference . . . 296
8.259.1 Detailed Description . . . 297
8.259.2 Field Documentation . . . 297
8.259.2.1 pSettingResp . . . 297
8.259.2.2 pSigCompEnabled . . . 297
8.259.2.3 pSIPLocalPort . . . 297
8.259.2.4 pSubscribeTimer . . . 297
8.259.2.5 pTimerSIPReg . . . 297
8.259.2.6 pTimerT1 . . . 297
8.259.2.7 pTimerT2 . . . 297
8.259.2.8 pTimerTf . . . 297
8.260 GnssData Struct Reference . . . 298
8.260.1 Detailed Description . . . 298
8.260.2 Field Documentation . . . 299
8.260.2.1 mask . . . 299
8.261 gnssSvInfoNotification Struct Reference . . . 299
8.261.1 Detailed Description . . . 300
8.261.2 Field Documentation . . . 300
8.261.2.1 bAltitudeAssumed . . . 300
8.261.2.2 pSatelliteInfo . . . 300
8.262 GPRSQoS Struct Reference . . . 300
8.262.1 Detailed Description . . . 300
8.262.2 Field Documentation . . . 301
8.262.2.1 delayClass . . . 301
8.262.2.2 meanThroughputClass . . . 301
8.262.2.3 peakThroughputClass . . . 301
8.262.2.4 precedenceClass . . . 301
8.262.2.5 reliabilityClass . . . 301
8.263 GPRSRequestedQoS Struct Reference . . . 301
8.263.1 Detailed Description . . . 301
8.263.2 Field Documentation . . . 302
8.263.2.1 delayClass . . . 302
8.263.2.2 meanThroughputClass . . . 302

8.263.2.3 peakThroughputClass . . . 302
8.263.2.4 precedenceClass . . . 302
8.263.2.5 reliabilityClass . . . 302
8.264 GPSStateInfo Struct Reference . . . 302
8.264.1 Detailed Description . . . 302
8.264.2 Field Documentation . . . 305
8.264.2.1 Altitude . . . 305
8.264.2.2 EngineState . . . 305
8.264.2.3 glo_almanac_sv_msk . . . 305
8.264.2.4 glo_ephemeris_sv_msk . . . 305
8.264.2.5 glo_health_sv_msk . . . 305
8.264.2.6 glo_visible_sv_msk . . . 305
8.264.2.7 gps_almanac_sv_msk . . . 305
8.264.2.8 gps_ephemeris_sv_msk . . . 305
8.264.2.9 gps_health_sv_msk . . . 305
8.264.2.10 gps_visible_sv_msk . . . 305
8.264.2.11 HorizontalUncertainty . . . 305
8.264.2.12 iono_valid . . . 305
8.264.2.13 Latitude . . . 305
8.264.2.14 Longitude . . . 305
8.264.2.15 sbas_almanac_sv_msk . . . 305
8.264.2.16 sbas_ephemeris_sv_msk . . . 305
8.264.2.17 sbas_health_sv_msk . . . 306
8.264.2.18 sbas_visible_sv_msk . . . 306
8.264.2.19 Time_uncert_ms . . . 306
8.264.2.20 TimeStmp_gps_week . . . 306
8.264.2.21 TimeStmp_tow_ms . . . 306
8.264.2.22 ValidMask . . . 306
8.264.2.23 VerticalUncertainty . . . 306
8.264.2.24 xtra_start_gps_minutes . . . 306
8.264.2.25 xtra_start_gps_week . . . 306
8.264.2.26 xtra_valid_duration_hours . . . 306
8.265 gpsTime_s Struct Reference . . . 306
8.265.1 Detailed Description . . . 306
8.265.2 Field Documentation . . . 306
8.265.2.1 gpsTimeOfWeekMs . . . 306
8.265.2.2 gpsWeek . . . 306
8.266 gsmCellInfo Struct Reference . . . 306
8.266.1 Detailed Description . . . 307
8.266.2 Field Documentation . . . 307

8.266.2.1 arfcn . . . 307
8.266.2.2 band1900 . . . 307
8.266.2.3 bsicId . . . 307
8.266.2.4 cellIdValid . . . 307
8.266.2.5 rssi . . . 307
8.266.2.6 srxlev . . . 307
8.267 GSMRSSIThresh Struct Reference . . . 308
8.267.1 Detailed Description . . . 308
8.267.2 Field Documentation . . . 308
8.267.2.1 GSMRSSIThreshListLen . . . 308
8.267.2.2 pGSMRSSIThreshList . . . 308
8.268 GSMSrvStatusInfo Struct Reference . . . 308
8.268.1 Detailed Description . . . 308
8.268.2 Field Documentation . . . 309
8.268.2.1 isPrefDataPath . . . 309
8.268.2.2 srvStatus . . . 309
8.268.2.3 trueSrvStatus . . . 309
8.269 GSMSysInfo Struct Reference . . . 309
8.269.1 Detailed Description . . . 309
8.269.2 Field Documentation . . . 311
8.269.2.1 cellId . . . 311
8.269.2.2 cellIdValid . . . 311
8.269.2.3 dtmSupp . . . 311
8.269.2.4 dtmSuppValid . . . 311
8.269.2.5 egprsSupp . . . 311
8.269.2.6 egprsSuppValid . . . 311
8.269.2.7 lac . . . 312
8.269.2.8 lacValid . . . 312
8.269.2.9 MCC . . . 312
8.269.2.10 MNC . . . 312
8.269.2.11 networkIdValid . . . 312
8.269.2.12 regRejectInfoValid . . . 312
8.269.2.13 rejCause . . . 312
8.269.2.14 rejectSrvDomain . . . 312
8.269.2.15 sysInfoGSM . . . 312
8.270 gyroAcceptReady_s Struct Reference . . . 312
8.270.1 Detailed Description . . . 312
8.270.2 Field Documentation . . . 312
8.270.2.1 batchPerSec . . . 312
8.270.2.2 injectEnable . . . 313

8.270.2.3 samplesPerBatch . . . 313
8.271 gyroTempAcceptReady_s Struct Reference . . . 313
8.271.1 Detailed Description . . . 313
8.271.2 Field Documentation . . . 313
8.271.2.1 batchPerSec . . . 313
8.271.2.2 injectEnable . . . 313
8.271.2.3 samplesPerBatch . . . 313
8.272 HDRECIOThresh Struct Reference . . . 313
8.272.1 Detailed Description . . . 314
8.272.2 Field Documentation . . . 314
8.272.2.1 HDRECIOThreshListLen . . . 314
8.272.2.2 pHDRECIOThreshList . . . 314
8.273 HDRIOThresh Struct Reference . . . 314
8.273.1 Detailed Description . . . 314
8.273.2 Field Documentation . . . 314
8.273.2.1 HDRIOThreshListLen . . . 314
8.273.2.2 pHDRIOThreshList . . . 314
8.274 HDRPersonalityInd Struct Reference . . . 314
8.274.1 Field Documentation . . . 315
8.274.1.1 pCurrentPersonality . . . 315
8.274.1.2 pPersonalityListLength . . . 315
8.274.1.3 pProtocolSubtypeElement . . . 315
8.275 HDRPersonalityResp Struct Reference . . . 315
8.275.1 Detailed Description . . . 315
8.275.2 Field Documentation . . . 315
8.275.2.1 pCurrentPersonality . . . 315
8.275.2.2 pPersonalityListLength . . . 315
8.275.2.3 pProtocolSubtypeElement . . . 315
8.276 HDRProtSubtypResp Struct Reference . . . 315
8.276.1 Detailed Description . . . 316
8.276.2 Field Documentation . . . 316
8.276.2.1 pAppSubType . . . 316
8.276.2.2 pCurrentPrsnlty . . . 316
8.276.2.3 pPersonalityListLength . . . 316
8.276.2.4 pProtoSubTypElmnt . . . 316
8.277 HDRRSSIThresh Struct Reference . . . 316
8.277.1 Detailed Description . . . 316
8.277.2 Field Documentation . . . 317
8.277.2.1 HDRRSSIThreshListLen . . . 317
8.277.2.2 pHDRRSSIThreshList . . . 317

8.278 HDRSINRThresh Struct Reference . . . 317
8.278.1 Detailed Description . . . 317
8.278.2 Field Documentation . . . 317
8.278.2.1 HDRSINRThresListLen . . . 317
8.278.2.2 pHDRSINRThresList . . . 317
8.279 HDRSINRThreshold Struct Reference . . . 317
8.279.1 Detailed Description . . . 318
8.279.2 Field Documentation . . . 318
8.279.2.1 HDRSINRThreshListLen . . . 318
8.279.2.2 pHDRSINRThreshList . . . 318
8.280 HDRSSInfo Struct Reference . . . 318
8.280.1 Detailed Description . . . 318
8.280.2 Field Documentation . . . 319
8.280.2.1 ecio . . . 319
8.280.2.2 io . . . 319
8.280.2.3 rssi . . . 319
8.280.2.4 sinr . . . 319
8.281 hdrSSInfo Struct Reference . . . 319
8.281.1 Detailed Description . . . 319
8.281.2 Field Documentation . . . 320
8.281.2.1 ecio . . . 320
8.281.2.2 io . . . 320
8.281.2.3 rssi . . . 320
8.281.2.4 sinr . . . 320
8.282 HDRSysInfo Struct Reference . . . 320
8.282.1 Detailed Description . . . 321
8.282.2 Field Documentation . . . 322
8.282.2.1 hdrActiveProt . . . 322
8.282.2.2 hdrActiveProtValid . . . 322
8.282.2.3 hdrPersonality . . . 322
8.282.2.4 hdrPersonalityValid . . . 322
8.282.2.5 is856SysId . . . 322
8.282.2.6 is856SysIdValid . . . 322
8.282.2.7 isSysPrlMatch . . . 322
8.282.2.8 isSysPrlMatchValid . . . 322
8.282.2.9 sysInfoHDR . . . 322
8.283 homeSIDNID Struct Reference . . . 322
8.283.1 Detailed Description . . . 322
8.283.2 Field Documentation . . . 323
8.283.2.1 numInstances . . . 323

8.283.2.2 SidNid . . . 323
8.284 hotSwapStatus Struct Reference . . . 323
8.284.1 Detailed Description . . . 323
8.284.2 Field Documentation . . . 323
8.284.2.1 hotSwap . . . 323
8.284.2.2 hotSwapLength . . . 323
8.285 hrpdMTUSizeTlv Struct Reference . . . 323
8.285.1 Detailed Description . . . 323
8.285.2 Field Documentation . . . 324
8.285.2.1 hrpdMTUSize . . . 324
8.285.2.2 TlvPresent . . . 324
8.286 image_info_t Struct Reference . . . 324
8.286.1 Detailed Description . . . 324
8.286.2 Field Documentation . . . 324
8.286.2.1 buildID . . . 324
8.286.2.2 buildIDLen . . . 324
8.286.2.3 imageType . . . 324
8.286.2.4 uniqueID . . . 325
8.287 ImageElement Struct Reference . . . 325
8.287.1 Detailed Description . . . 325
8.287.2 Field Documentation . . . 325
8.287.2.1 buildId . . . 325
8.287.2.2 buildIdLength . . . 325
8.287.2.3 imageId . . . 325
8.287.2.4 imageType . . . 325
8.288 ImageIdElement Struct Reference . . . 325
8.288.1 Detailed Description . . . 326
8.288.2 Field Documentation . . . 326
8.288.2.1 buildID . . . 326
8.288.2.2 buildIDLength . . . 326
8.288.2.3 failureCount . . . 326
8.288.2.4 imageID . . . 326
8.288.2.5 storageIndex . . . 326
8.289 ImageIDEntries Struct Reference . . . 326
8.289.1 Detailed Description . . . 326
8.289.2 Field Documentation . . . 327
8.289.2.1 executingImage . . . 327
8.289.2.2 imageIDElement . . . 327
8.289.2.3 imageIDSize . . . 327
8.289.2.4 imageType . . . 327

8.289.2.5 maxImages . . . 327
8.290 ImageList Struct Reference . . . 327
8.290.1 Detailed Description . . . 327
8.290.2 Field Documentation . . . 327
8.290.2.1 imageIDEntries . . . 328
8.290.2.2 listSize . . . 328
8.291 IMSAIndRegisterInfo Struct Reference . . . 328
8.291.1 Detailed Description . . . 328
8.291.2 Field Documentation . . . 328
8.291.2.1 pPdpStatusConfig . . . 328
8.291.2.2 pRatHandoverStatusConfig . . . 329
8.291.2.3 pRegStatusConfig . . . 329
8.291.2.4 pServiceStatusConfig . . . 329
8.292 imsaPdpStatusInfo Struct Reference . . . 329
8.292.1 Detailed Description . . . 329
8.292.2 Field Documentation . . . 329
8.292.2.1 connetionState . . . 329
8.292.2.2 pFailErrorCode . . . 329
8.293 imsaRatStatusInfo Struct Reference . . . 329
8.293.1 Detailed Description . . . 329
8.293.2 Field Documentation . . . 330
8.293.2.1 pErrorCodeStr . . . 330
8.293.2.2 pRATStatus . . . 330
8.293.2.3 pSrcRAT . . . 330
8.293.2.4 pTgtRAT . . . 330
8.294 IMSARegistrationStatus Struct Reference . . . 330
8.294.1 Detailed Description . . . 330
8.294.2 Field Documentation . . . 331
8.294.2.1 pImsRegErrCode . . . 331
8.294.2.2 pImsRegStatus . . . 331
8.294.2.3 pNewImsRegStatus . . . 331
8.295 imsaRegStatusInfo Struct Reference . . . 331
8.295.1 Detailed Description . . . 331
8.295.2 Field Documentation . . . 331
8.295.2.1 pbIMSRegistered . . . 331
8.295.2.2 pImsRegStatus . . . 331
8.295.2.3 pRegStatusErrorCode . . . 331
8.296 IMSAServiceStatus Struct Reference . . . 331
8.296.1 Detailed Description . . . 332
8.296.2 Field Documentation . . . 333

8.296.2.1 pSmsServiceRat . . . 333
8.296.2.2 pSmsServiceStatus . . . 333
8.296.2.3 pUtServiceRat . . . 333
8.296.2.4 pUtServiceStatus . . . 333
8.296.2.5 pVoipServiceRat . . . 333
8.296.2.6 pVoipServiceStatus . . . 333
8.296.2.7 pVsServiceRat . . . 333
8.296.2.8 pVsServiceStatus . . . 333
8.296.2.9 pVtServiceRat . . . 334
8.296.2.10 pVtServiceStatus . . . 334
8.297 IMSASupportedFieldsResp Struct Reference . . . 334
8.297.1 Detailed Description . . . 334
8.297.2 Field Documentation . . . 334
8.297.2.1 pIndFieldsList . . . 334
8.297.2.2 pReqFieldsList . . . 334
8.297.2.3 pRespFieldsList . . . 334
8.298 IMSASupportedMsgInfo Struct Reference . . . 334
8.298.1 Detailed Description . . . 334
8.298.2 Field Documentation . . . 335
8.298.2.1 pSupportedMsgList . . . 335
8.299 imsaSvcStatusInfo Struct Reference . . . 335
8.299.1 Detailed Description . . . 335
8.299.2 Field Documentation . . . 335
8.299.2.1 pSMSSvcRAT . . . 335
8.299.2.2 pSMSSvcStatus . . . 335
8.299.2.3 pUTSvcRAT . . . 335
8.299.2.4 pUTSvcStatus . . . 335
8.299.2.5 pVOIPSvcRAT . . . 336
8.299.2.6 pVOIPSvcStatus . . . 336
8.299.2.7 pVTSvcRAT . . . 336
8.299.2.8 pVTSvcStatus . . . 336
8.300 imsCfgIndRegisterInfo Struct Reference . . . 336
8.300.1 Detailed Description . . . 336
8.300.2 Field Documentation . . . 337
8.300.2.1 pRegMgrConfigEvents . . . 337
8.300.2.2 pSIPConfigEvents . . . 337
8.300.2.3 pSMSConfigEvents . . . 337
8.300.2.4 pUserConfigEvents . . . 337
8.300.2.5 pVoIPConfigEvents . . . 337
8.301 imsRegMgrConfigInfo Struct Reference . . . 337

8.301.1 Detailed Description . . . 337
8.301.2 Field Documentation . . . 338
8.301.2.1 pCSCFPortName . . . 338
8.301.2.2 pIMSTestMode . . . 338
8.301.2.3 pPriCSCFPort . . . 338
8.302imsSIPConfigInfo Struct Reference . . . 338
8.302.1 Detailed Description . . . 338
8.302.2 Field Documentation . . . 339
8.302.2.1 pSigCompEnabled . . . 339
8.302.2.2 pSIPLocalPort . . . 339
8.302.2.3 pSubscribeTimer . . . 339
8.302.2.4 pTimerSIPReg . . . 339
8.302.2.5 pTimerT1 . . . 339
8.302.2.6 pTimerT2 . . . 339
8.302.2.7 pTimerTf . . . 339
8.303imsSMSConfigInfo Struct Reference . . . 339
8.303.1 Detailed Description . . . 339
8.303.2 Field Documentation . . . 339
8.303.2.1 pPhoneCtxtURI . . . 339
8.303.2.2 pSMSFormat . . . 339
8.303.2.3 pSMSOverIPNwInd . . . 339
8.304imsUserConfigInfo Struct Reference . . . 340
8.304.1 Detailed Description . . . 340
8.304.2 Field Documentation . . . 340
8.304.2.1 pIMSDomain . . . 340
8.305imsVoIPConfigInfo Struct Reference . . . 340
8.305.1 Detailed Description . . . 340
8.305.2 Field Documentation . . . 342
8.305.2.1 pAmrMode . . . 342
8.305.2.2 pAmrOctetAligned . . . 342
8.305.2.3 pAmrWbEnable . . . 342
8.305.2.4 pAmrWBMode . . . 342
8.305.2.5 pAmrWBOctetAligned . . . 342
8.305.2.6 pMinSessionExpiryTimer . . . 342
8.305.2.7 pRingBackTimer . . . 342
8.305.2.8 pRingingTimer . . . 342
8.305.2.9 pRTPRTCPInactTimer . . . 342
8.305.2.10 pScrAmrEnable . . . 342
8.305.2.11 pScrAmrWbEnable . . . 342
8.305.2.12 pSessionExpiryTimer . . . 342

8.306 IndFieldsList Struct Reference . . . 342
8.306.1 Detailed Description . . . 343
8.306.2 Field Documentation . . . 343
8.306.2.1 indicationFields . . . 343
8.306.2.2 indicationFieldsLen . . . 343
8.307 infoInterFreq Struct Reference . . . 343
8.307.1 Detailed Description . . . 343
8.307.2 Field Documentation . . . 344
8.307.2.1 cell_resel_priority . . . 344
8.307.2.2 cellInterFreqParams . . . 344
8.307.2.3 cells_len . . . 344
8.307.2.4 earfcn . . . 344
8.307.2.5 threshXHigh . . . 344
8.307.2.6 threshXLow . . . 344
8.308 IOThresh Struct Reference . . . 344
8.308.1 Detailed Description . . . 344
8.308.2 Field Documentation . . . 345
8.308.2.1 IOThresListLen . . . 345
8.308.2.2 pIOThresList . . . 345
8.309 IPv4Addr Struct Reference . . . 345
8.309.1 Detailed Description . . . 345
8.309.2 Field Documentation . . . 345
8.309.2.1 addr . . . 345
8.309.2.2 subnetMask . . . 345
8.310 IPv4Config Struct Reference . . . 345
8.310.1 Detailed Description . . . 345
8.310.2 Field Documentation . . . 346
8.310.2.1 IPv4Addr . . . 346
8.310.2.2 IPv4Port . . . 346
8.311 IPv4Info Struct Reference . . . 346
8.311.1 Detailed Description . . . 346
8.311.2 Field Documentation . . . 346
8.311.2.1 address . . . 346
8.311.2.2 port . . . 346
8.311.2.3 TlvPresent . . . 346
8.312 IPv6Addr Struct Reference . . . 346
8.312.1 Detailed Description . . . 346
8.312.2 Field Documentation . . . 347
8.312.2.1 addr . . . 347
8.312.2.2 prefixLen . . . 347

8.313 IPV6AddressInfo Struct Reference . . . 347
8.313.1 Detailed Description . . . 347
8.313.2 Field Documentation . . . 347
8.313.2.1 IPAddressV6 . . . 347
8.313.2.2 IPV6PrefixLen . . . 347
8.314 ipv6AddressInfo Struct Reference . . . 347
8.314.1 Detailed Description . . . 347
8.314.2 Field Documentation . . . 348
8.314.2.1 IPAddressV6 . . . 348
8.314.2.2 IPV6PrefixLen . . . 348
8.315 IPv6Config Struct Reference . . . 348
8.315.1 Detailed Description . . . 348
8.315.2 Field Documentation . . . 348
8.315.2.1 IPv6Addr . . . 348
8.315.2.2 IPv6Port . . . 348
8.316 IPV6GWAddressInfo Struct Reference . . . 348
8.316.1 Detailed Description . . . 348
8.316.2 Field Documentation . . . 349
8.316.2.1 gwAddressV6 . . . 349
8.316.2.2 gwV6PrefixLen . . . 349
8.317 IPv6Info Struct Reference . . . 349
8.317.1 Detailed Description . . . 349
8.317.2 Field Documentation . . . 349
8.317.2.1 address . . . 349
8.317.2.2 port . . . 349
8.317.2.3 TlvPresent . . . 349
8.318 IPv6TrafCls Struct Reference . . . 349
8.318.1 Detailed Description . . . 350
8.318.2 Field Documentation . . . 350
8.318.2.1 mask . . . 350
8.318.2.2 val . . . 350
8.319 LibPackGPRSRequestedQoS Struct Reference . . . 350
8.319.1 Detailed Description . . . 350
8.319.2 Field Documentation . . . 351
8.319.2.1 delayClass . . . 351
8.319.2.2 meanThroughputClass . . . 351
8.319.2.3 peakThroughputClass . . . 351
8.319.2.4 precedenceClass . . . 351
8.319.2.5 reliabilityClass . . . 351
8.320 LibpackProfile3GPP Struct Reference . . . 351

8.320.1 Detailed Description . . . 352
8.320.2 Field Documentation . . . 356
8.320.2.1 pAddrAllocPref . . . 356
8.320.2.2 pAPNClass . . . 356
8.320.2.3 pAPNDisabledFlag . . . 356
8.320.2.4 pAPNName . . . 356
8.320.2.5 pAPNnameSize . . . 356
8.320.2.6 pAuthenticationPref . . . 356
8.320.2.7 pGPRSMinimumQoS . . . 356
8.320.2.8 pGPRSRequestedQos . . . 357
8.320.2.9 pImCnFlag . . . 357
8.320.2.10 pIPv4AddrPref . . . 357
8.320.2.11 pIPv6AddPref . . . 357
8.320.2.12 pPassword . . . 357
8.320.2.13 pPasswordSize . . . 357
8.320.2.14 pPcscfAddrUsingDhcp . . . 357
8.320.2.15 pPcscfAddrUsingPCO . . . 357
8.320.2.16 pPDNInactivTimeout . . . 357
8.320.2.17 pPdpAccessConFlag . . . 357
8.320.2.18 pPdpContext . . . 357
8.320.2.19 pPdpDataCompType . . . 357
8.320.2.20 pPdpHdrCompType . . . 357
8.320.2.21 pPDPtype . . . 357
8.320.2.22 pPriDNSIPv4AddPref . . . 357
8.320.2.23 pPriDNSIPv6addpref . . . 357
8.320.2.24 pPrimaryID . . . 357
8.320.2.25 pProfilename . . . 357
8.320.2.26 pProfilenameSize . . . 357
8.320.2.27 pQosClassID . . . 357
8.320.2.28 pSecDNSIPv4AddPref . . . 357
8.320.2.29 pSecDNSIPv6addpref . . . 357
8.320.2.30 pSecondaryFlag . . . 357
8.320.2.31 pSupportEmergencyCalls . . . 357
8.320.2.32 pTFTID1Params . . . 357
8.320.2.33 pTFTID2Params . . . 357
8.320.2.34 pUMTSMinQoS . . . 357
8.320.2.35 pUMTSMinQosSigInd . . . 357
8.320.2.36 pUMTSReqQoS . . . 358
8.320.2.37 pUMTSReqQoSSigInd . . . 358
8.320.2.38 pUsername . . . 358

8.320.2.39 pUsernameSize . . . 358
8.321 LibpackProfile3GPP2 Struct Reference . . . 358
8.321.1 Detailed Description . . . 358
8.321.2 Field Documentation . . . 362
8.321.2.1 pAllowLinger . . . 362
8.321.2.2 pAPNClass3GPP2 . . . 362
8.321.2.3 pAPNEnabled3GPP2 . . . 362
8.321.2.4 pApnString . . . 362
8.321.2.5 pApnStringSize . . . 362
8.321.2.6 pAppPriority . . . 362
8.321.2.7 pAppType . . . 362
8.321.2.8 pAuthPassword . . . 362
8.321.2.9 pAuthPasswordSize . . . 362
8.321.2.10 pAuthProtocol . . . 362
8.321.2.11 pAuthRetryCount . . . 362
8.321.2.12 pAuthTimeout . . . 362
8.321.2.13 pDataMode . . . 362
8.321.2.14 pDataRate . . . 362
8.321.2.15 pIpcpAckTimeout . . . 362
8.321.2.16 pIpcpCreqRetryCount . . . 362
8.321.2.17 pIsPcscfAddressNedded . . . 362
8.321.2.18 pLcpAckTimeout . . . 362
8.321.2.19 pLcpCreqRetryCount . . . 363
8.321.2.20 pNegoDnsSrvrPref . . . 363
8.321.2.21 pPDNInactivTimeout3GPP2 . . . 363
8.321.2.22 pPdnType . . . 363
8.321.2.23 pPppSessCloseTimer1x . . . 363
8.321.2.24 pPppSessCloseTimerDO . . . 363
8.321.2.25 pPrimaryV4DnsAddress . . . 363
8.321.2.26 pPriV6DnsAddress . . . 363
8.321.2.27 pRATType . . . 363
8.321.2.28 pSecondaryV4DnsAddress . . . 363
8.321.2.29 pSecV6DnsAddress . . . 363
8.321.2.30 pUserId . . . 363
8.321.2.31 pUserIdSize . . . 363
8.322 LibPackprofile_3GPP Struct Reference . . . 363
8.322.1 Detailed Description . . . 364
8.322.2 Field Documentation . . . 367
8.322.2.1 pAddrAllocPref . . . 368
8.322.2.2 pAPNClass . . . 368

8.322.2.3 pAPNDisabledFlag . . . 368
8.322.2.4 pAPNName . . . 368
8.322.2.5 pAPNnameSize . . . 368
8.322.2.6 pAuthenticationPref . . . 368
8.322.2.7 pGPRSMinimumQoS . . . 368
8.322.2.8 pGPRSRequestedQos . . . 368
8.322.2.9 pImCnFlag . . . 368
8.322.2.10 pIPv4AddrPref . . . 368
8.322.2.11 pIPv6AddPref . . . 368
8.322.2.12 pPassword . . . 368
8.322.2.13 pPasswordSize . . . 368
8.322.2.14 pPcscfAddrUsingDhcp . . . 368
8.322.2.15 pPcscfAddrUsingPCO . . . 368
8.322.2.16 pPDNInactivTimeout . . . 368
8.322.2.17 pPdpAccessConFlag . . . 368
8.322.2.18 pPdpContext . . . 368
8.322.2.19 pPdpDataCompType . . . 368
8.322.2.20 pPdpHdrCompType . . . 368
8.322.2.21 pPDPtype . . . 368
8.322.2.22 pPriDNSIPv4AddPref . . . 368
8.322.2.23 pPriDNSIPv6addpref . . . 368
8.322.2.24 pPrimaryID . . . 368
8.322.2.25 pProfilename . . . 368
8.322.2.26 pProfilenameSize . . . 368
8.322.2.27 pQosClassID . . . 368
8.322.2.28 pSecDNSIPv4AddPref . . . 368
8.322.2.29 pSecDNSIPv6addpref . . . 369
8.322.2.30 pSecondaryFlag . . . 369
8.322.2.31 pSupportEmergencyCalls . . . 369
8.322.2.32 pTFTID1Params . . . 369
8.322.2.33 pTFTID2Params . . . 369
8.322.2.34 pUMTSMinQoS . . . 369
8.322.2.35 pUMTSMinQosSigInd . . . 369
8.322.2.36 pUMTSReqQoS . . . 369
8.322.2.37 pUMTSReqQoSSigInd . . . 369
8.322.2.38 pUsername . . . 369
8.322.2.39 pUsernameSize . . . 369
8.323 LibPackprofile_3GPP2 Struct Reference . . . 369
8.323.1 Detailed Description . . . 370
8.323.2 Field Documentation . . . 373

8.323.2.1 pAllowLinger . . . 373
8.323.2.2 pAPNClass3GPP2 . . . 373
8.323.2.3 pAPNEnabled3GPP2 . . . 373
8.323.2.4 pApnString . . . 373
8.323.2.5 pApnStringSize . . . 373
8.323.2.6 pAppPriority . . . 373
8.323.2.7 pAppType . . . 373
8.323.2.8 pAuthPassword . . . 373
8.323.2.9 pAuthPassword_tSize . . . 373
8.323.2.10 pAuthProtocol . . . 373
8.323.2.11 pAuthRetryCount . . . 373
8.323.2.12 pAuthTimeout . . . 373
8.323.2.13 pDataMode . . . 373
8.323.2.14 pDataRate . . . 373
8.323.2.15 pIpcpAckTimeout . . . 374
8.323.2.16 pIpcpCreqRetryCount . . . 374
8.323.2.17 pIsPcscfAddressNedded . . . 374
8.323.2.18 pLcpAckTimeout . . . 374
8.323.2.19 pLcpCreqRetryCount . . . 374
8.323.2.20 pNegoDnsSrvrPref . . . 374
8.323.2.21 pPDNInactivTimeout3GPP2 . . . 374
8.323.2.22 pPdnType . . . 374
8.323.2.23 pPppSessCloseTimer1x . . . 374
8.323.2.24 pPppSessCloseTimerDO . . . 374
8.323.2.25 pPrimaryV4DnsAddress . . . 374
8.323.2.26 pPriV6DnsAddress . . . 374
8.323.2.27 pRATType . . . 374
8.323.2.28 pSecondaryV4DnsAddress . . . 374
8.323.2.29 pSecV6DnsAddress . . . 374
8.323.2.30 pUserId . . . 374
8.323.2.31 pUserIdSize . . . 374
8.324 LibPackQosClassID Struct Reference . . . 374
8.324.1 Detailed Description . . . 374
8.324.2 Field Documentation . . . 375
8.324.2.1 gDlBitRate . . . 375
8.324.2.2 gUlBitRate . . . 375
8.324.2.3 maxDlBitRate . . . 375
8.324.2.4 maxUlBitRate . . . 375
8.324.2.5 QCI . . . 375
8.325 LibPackTFTIDParams Struct Reference . . . 375

8.325.1 Detailed Description . . . 375
8.325.2 Field Documentation . . . 376
8.325.2.1 destPortRangeEnd . . . 376
8.325.2.2 destPortRangeStart . . . 376
8.325.2.3 eValid . . . 377
8.325.2.4 filterId . . . 377
8.325.2.5 flowLabel . . . 377
8.325.2.6 IPSECSPI . . . 377
8.325.2.7 ipVersion . . . 377
8.325.2.8 nextHeader . . . 377
8.325.2.9 pSourceIP . . . 377
8.325.2.10 sourceIPMask . . . 377
8.325.2.11 srcPortRangeEnd . . . 377
8.325.2.12 srcPortRangeStart . . . 377
8.325.2.13 tosMask . . . 377
8.326 LibPackUMTSQoS Struct Reference . . . 377
8.326.1 Detailed Description . . . 377
8.326.2 Field Documentation . . . 379
8.326.2.1 deliveryErrSDU . . . 379
8.326.2.2 grntDownlinkBitrate . . . 379
8.326.2.3 grntUplinkBitrate . . . 379
8.326.2.4 maxDownlinkBitrate . . . 379
8.326.2.5 maxSDUSize . . . 379
8.326.2.6 maxUplinkBitrate . . . 379
8.326.2.7 qosDeliveryOrder . . . 379
8.326.2.8 resBerRatio . . . 379
8.326.2.9 sduErrorRatio . . . 379
8.326.2.10 trafficClass . . . 379
8.326.2.11 trafficPriority . . . 379
8.326.2.12 transferDelay . . . 379
8.327 LibPackUMTSReqQoSSigInd Struct Reference . . . 380
8.327.1 Detailed Description . . . 380
8.327.2 Field Documentation . . . 380
8.327.2.1 SigInd . . . 380
8.327.2.2 UMTSReqQoS . . . 380
8.328 lineCtrlInfo Struct Reference . . . 380
8.328.1 Detailed Description . . . 380
8.328.2 Field Documentation . . . 381
8.328.2.1 polarityIncluded . . . 381
8.328.2.2 pwrDenialTime . . . 381

8.328.2.3 revPolarity . . . 381
8.328.2.4 toggleMode . . . 381
8.329loc_accelAcceptReady Struct Reference . . . 381
8.329.1 Detailed Description . . . 381
8.329.2 Field Documentation . . . 381
8.329.2.1 batchPerSec . . . 381
8.329.2.2 injectEnable . . . 382
8.329.2.3 samplesPerBatch . . . 382
8.330loc_accelTempAcceptReady Struct Reference . . . 382
8.330.1 Detailed Description . . . 382
8.330.2 Field Documentation . . . 382
8.330.2.1 batchPerSec . . . 382
8.330.2.2 injectEnable . . . 382
8.330.2.3 samplesPerBatch . . . 382
8.331loc_BdsSV Struct Reference . . . 382
8.331.1 Detailed Description . . . 383
8.331.2 Field Documentation . . . 383
8.331.2.1 id . . . 383
8.331.2.2 mask . . . 383
8.332loc_BdsSVInfo Struct Reference . . . 383
8.332.1 Detailed Description . . . 383
8.332.2 Field Documentation . . . 383
8.332.2.1 len . . . 383
8.332.2.2 pSV . . . 383
8.333loc_CellDb Struct Reference . . . 383
8.333.1 Detailed Description . . . 384
8.333.2 Field Documentation . . . 384
8.333.2.1 mask . . . 384
8.334loc_ClkInfo Struct Reference . . . 384
8.334.1 Detailed Description . . . 384
8.334.2 Field Documentation . . . 385
8.334.2.1 mask . . . 385
8.335loc_GnssData Struct Reference . . . 385
8.335.1 Detailed Description . . . 386
8.335.2 Field Documentation . . . 386
8.335.2.1 mask . . . 387
8.336loc_gpsTime Struct Reference . . . 387
8.336.1 Detailed Description . . . 387
8.336.2 Field Documentation . . . 387
8.336.2.1 gpsTimeOfWeekMs . . . 387

8.336.2.2 gpsWeek . . . 387
8.337 loc_gyroAcceptReady Struct Reference . . . 387
8.337.1 Detailed Description . . . 387
8.337.2 Field Documentation . . . 388
8.337.2.1 batchPerSec . . . 388
8.337.2.2 injectEnable . . . 388
8.337.2.3 samplesPerBatch . . . 388
8.338 loc_gyroTempAcceptReady Struct Reference . . . 388
8.338.1 Detailed Description . . . 388
8.338.2 Field Documentation . . . 389
8.338.2.1 batchPerSec . . . 389
8.338.2.2 injectEnable . . . 389
8.338.2.3 samplesPerBatch . . . 389
8.339 loc_IPv4Config Struct Reference . . . 389
8.339.1 Detailed Description . . . 389
8.339.2 Field Documentation . . . 389
8.339.2.1 IPv4Addr . . . 389
8.339.2.2 IPv4Port . . . 389
8.340 loc_IPv4Info Struct Reference . . . 389
8.340.1 Detailed Description . . . 389
8.340.2 Field Documentation . . . 390
8.340.2.1 address . . . 390
8.340.2.2 port . . . 390
8.340.2.3 TlvPresent . . . 390
8.341 loc_IPv6Config Struct Reference . . . 390
8.341.1 Detailed Description . . . 390
8.341.2 Field Documentation . . . 390
8.341.2.1 IPv6Addr . . . 390
8.341.2.2 IPv6Port . . . 390
8.342 loc_IPv6Info Struct Reference . . . 390
8.342.1 Detailed Description . . . 390
8.342.2 Field Documentation . . . 391
8.342.2.1 address . . . 391
8.342.2.2 port . . . 391
8.342.2.3 TlvPresent . . . 391
8.343 loc_LocApplicationInfo Struct Reference . . . 391
8.343.1 Detailed Description . . . 391
8.343.2 Field Documentation . . . 392
8.343.2.1 appNameLength . . . 392
8.343.2.2 appProviderLength . . . 392

8.343.2.3 appVersionLength . . . 392
8.343.2.4 appVersionValid . . . 392
8.343.2.5 pAppName . . . 392
8.343.2.6 pAppProvider . . . 392
8.343.2.7 pAppVersion . . . 392
8.344loc_precisionDilution Struct Reference . . . 392
8.344.1 Detailed Description . . . 392
8.344.2 Field Documentation . . . 393
8.344.2.1 HDOP . . . 393
8.344.2.2 PDOP . . . 393
8.344.2.3 VDOP . . . 393
8.345loc_satelliteInfo Struct Reference . . . 393
8.345.1 Detailed Description . . . 393
8.345.2 Field Documentation . . . 395
8.345.2.1 azimuth . . . 395
8.345.2.2 elevation . . . 395
8.345.2.3 gnssSvId . . . 395
8.345.2.4 healthStatus . . . 395
8.345.2.5 snr . . . 395
8.345.2.6 svInfoMask . . . 395
8.345.2.7 svListLen . . . 395
8.345.2.8 svStatus . . . 395
8.345.2.9 system . . . 395
8.345.2.10validMask . . . 395
8.346loc_sensorDataUsage Struct Reference . . . 395
8.346.1 Detailed Description . . . 395
8.346.2 Field Documentation . . . 396
8.346.2.1 aidingIndicatorMask . . . 396
8.346.2.2 usageMask . . . 396
8.347loc_SV Struct Reference . . . 396
8.347.1 Detailed Description . . . 396
8.347.2 Field Documentation . . . 397
8.347.2.1 id . . . 397
8.347.2.2 mask . . . 397
8.347.2.3 system . . . 397
8.348loc_SVInfo Struct Reference . . . 397
8.348.1 Detailed Description . . . 397
8.348.2 Field Documentation . . . 397
8.348.2.1 len . . . 397
8.348.2.2 pSV . . . 397

8.349loc_svUsedforFix Struct Reference . . . 397
8.349.1 Detailed Description . . . 397
8.349.2 Field Documentation . . . 398
8.349.2.1 gnssSvUsedList . . . 398
8.349.2.2 gnssSvUsedList_len . . . 398
8.350loc_urlAddr Struct Reference . . . 398
8.350.1 Detailed Description . . . 398
8.350.2 Field Documentation . . . 398
8.350.2.1 address . . . 398
8.350.2.2 TlvPresent . . . 398
8.351loc_URLAddrInfo Struct Reference . . . 398
8.351.1 Detailed Description . . . 399
8.351.2 Field Documentation . . . 399
8.351.2.1 urlAddr . . . 399
8.352LocApplicationInfo Struct Reference . . . 399
8.352.1 Detailed Description . . . 399
8.352.2 Field Documentation . . . 400
8.352.2.1 appNameLength . . . 400
8.352.2.2 appProviderLength . . . 400
8.352.2.3 appVersionLength . . . 400
8.352.2.4 appVersionValid . . . 400
8.352.2.5 pAppName . . . 400
8.352.2.6 pAppProvider . . . 400
8.352.2.7 pAppVersion . . . 400
8.353LocDelAssDataReq Struct Reference . . . 400
8.353.1 Detailed Description . . . 400
8.353.2 Field Documentation . . . 401
8.353.2.1 pBdsSVInfo . . . 401
8.353.2.2 pCellDb . . . 401
8.353.2.3 pClkInfo . . . 401
8.353.2.4 pGnssData . . . 401
8.353.2.5 pSVInfo . . . 401
8.354LOCEventRegisterReqResp Struct Reference . . . 401
8.354.1 Detailed Description . . . 401
8.354.2 Field Documentation . . . 403
8.354.2.1 eventRegister . . . 403
8.355LOCExtPowerStateReqResp Struct Reference . . . 403
8.355.1 Detailed Description . . . 403
8.355.2 Field Documentation . . . 403
8.355.2.1 extPowerState . . . 403

8.356 LocGetServerReq Struct Reference . . . 403
8.356.1 Detailed Description . . . 403
8.356.2 Field Documentation . . . 404
8.356.2.1 pServerAddrTypeMask . . . 404
8.356.2.2 serverType . . . 404
8.357 LocInjectPositionReq Struct Reference . . . 404
8.357.1 Detailed Description . . . 404
8.357.2 Field Documentation . . . 407
8.357.2.1 pAltitudeSrcInfo . . . 407
8.357.2.2 pAltitudeWrtEllipsoid . . . 407
8.357.2.3 pAltitudeWrtMeanSeaLevel . . . 407
8.357.2.4 pHorConfidence . . . 408
8.357.2.5 pHorReliability . . . 408
8.357.2.6 pHorUncCircular . . . 408
8.357.2.7 pLatitude . . . 408
8.357.2.8 pLongitude . . . 408
8.357.2.9 pPositionSrc . . . 408
8.357.2.10 pRawHorConfidence . . . 408
8.357.2.11 pRawHorUncCircular . . . 408
8.357.2.12 pTimestampAge . . . 408
8.357.2.13 pTimestampUtc . . . 408
8.357.2.14 pVertConfidence . . . 408
8.357.2.15 pVertReliability . . . 408
8.357.2.16 pVertUnc . . . 408
8.358 LocInjectSensorDataReq Struct Reference . . . 408
8.358.1 Detailed Description . . . 408
8.358.2 Field Documentation . . . 409
8.358.2.1 pAcceleroData . . . 409
8.358.2.2 pAcceleroTempData . . . 409
8.358.2.3 pAcceleroTimeSrc . . . 409
8.358.2.4 pGyroData . . . 409
8.358.2.5 pGyroTempData . . . 409
8.358.2.6 pGyroTimeSrc . . . 409
8.358.2.7 pOpaqueIdentifier . . . 409
8.359 LocSetCradleMountReq Struct Reference . . . 409
8.359.1 Detailed Description . . . 410
8.359.2 Field Documentation . . . 410
8.359.2.1 pConfidence . . . 410
8.359.2.2 state . . . 410
8.360 LocSetServerReq Struct Reference . . . 410

8.360.1 Detailed Description . . . 410
8.360.2 Field Documentation . . . 411
8.360.2.1 pIPv4Config . . . 411
8.360.2.2 pIPv6Config . . . 411
8.360.2.3 pURLAddr . . . 411
8.360.2.4 serverType . . . 411
8.361 LOCStartReq Struct Reference . . . 411
8.361.1 Detailed Description . . . 411
8.361.2 Field Documentation . . . 412
8.361.2.1 pApplicationInfo . . . 412
8.361.2.2 pConfigAltitudeAssumed . . . 412
8.361.2.3 pHorizontalAccuracyLvl . . . 412
8.361.2.4 pIntermediateReportState . . . 412
8.361.2.5 pMinIntervalTime . . . 412
8.361.2.6 pRecurrenceType . . . 412
8.361.2.7 SessionId . . . 413
8.362 LOCStopReq Struct Reference . . . 413
8.362.1 Detailed Description . . . 413
8.362.2 Field Documentation . . . 413
8.362.2.1 sessionId . . . 413
8.363 LteCQIParm Struct Reference . . . 413
8.363.1 Detailed Description . . . 413
8.363.2 Field Documentation . . . 414
8.363.2.1 CQIValueCW0 . . . 414
8.363.2.2 CQIValueCW1 . . . 414
8.363.2.3 ValidityCW0 . . . 414
8.363.2.4 ValidityCW1 . . . 414
8.364 lteEARFCN Struct Reference . . . 414
8.364.1 Detailed Description . . . 414
8.364.2 Field Documentation . . . 414
8.364.2.1 earfcn0 . . . 414
8.364.2.2 earfcn1 . . . 414
8.364.2.3 status . . . 414
8.365 lteGsmCellInfo Struct Reference . . . 414
8.365.1 Detailed Description . . . 415
8.365.2 Field Documentation . . . 415
8.365.2.1 cellReselPriority . . . 415
8.365.2.2 cells_len . . . 415
8.365.2.3 GsmCellInfo . . . 415
8.365.2.4 nccPermitted . . . 415

8.365.2.5 threshGsmHigh . . . 416
8.365.2.6 threshGsmLow . . . 416
8.366 LTEInfo Struct Reference . . . 416
8.366.1 Detailed Description . . . 416
8.366.2 Field Documentation . . . 417
8.366.2.1 band . . . 417
8.366.2.2 bandwidth . . . 418
8.366.2.3 emmConnState . . . 418
8.366.2.4 emmState . . . 418
8.366.2.5 emmSubState . . . 418
8.366.2.6 RXChan . . . 418
8.366.2.7 TXChan . . . 418
8.367 LTEInfoInterfreq Struct Reference . . . 418
8.367.1 Detailed Description . . . 418
8.367.2 Field Documentation . . . 418
8.367.2.1 freqsLen . . . 418
8.367.2.2 InfoInterfreq . . . 418
8.367.2.3 ueInIdle . . . 418
8.368 LTEInfoIntrafreq Struct Reference . . . 418
8.368.1 Detailed Description . . . 419
8.368.2 Field Documentation . . . 420
8.368.2.1 CellParams . . . 420
8.368.2.2 cellReselPriority . . . 420
8.368.2.3 cellsLen . . . 420
8.368.2.4 earfcn . . . 420
8.368.2.5 globalCellId . . . 420
8.368.2.6 plmn . . . 420
8.368.2.7 servingCellId . . . 420
8.368.2.8 sIntraSearch . . . 420
8.368.2.9 sNonIntraSearch . . . 420
8.368.2.10 tac . . . 420
8.368.2.11 threshServingLow . . . 420
8.368.2.12 ueInIdle . . . 420
8.369 LTEInfoNeighboringGSM Struct Reference . . . 420
8.369.1 Detailed Description . . . 421
8.369.2 Field Documentation . . . 421
8.369.2.1 freqsLen . . . 421
8.369.2.2 LteGsmCellInfo . . . 421
8.369.2.3 ueInIdle . . . 421
8.370 LTEInfoNeighboringWCDMA Struct Reference . . . 421

8.370.1 Detailed Description . . . 421
8.370.2 Field Documentation . . . 422
8.370.2.1 freqsLen . . . 422
8.370.2.2 LTEWCDMACellInfo . . . 422
8.370.2.3 ueInIdle . . . 422
8.371 LteNasReleaseInfo_s Struct Reference . . . 422
8.371.1 Detailed Description . . . 422
8.371.2 Field Documentation . . . 422
8.371.2.1 nas_major . . . 422
8.371.2.2 nas_minor . . . 422
8.371.2.3 nas_release . . . 422
8.372 LTEOperationMode Struct Reference . . . 422
8.372.1 Detailed Description . . . 423
8.372.2 Field Documentation . . . 423
8.372.2.1 pLTEOperationMode . . . 423
8.372.2.2 TlvPresent . . . 423
8.373 ltePCI Struct Reference . . . 423
8.373.1 Detailed Description . . . 423
8.373.2 Field Documentation . . . 423
8.373.2.1 earfcn . . . 423
8.373.2.2 pci . . . 423
8.373.2.3 status . . . 423
8.374 lteRsrpinformation Struct Reference . . . 424
8.374.1 Detailed Description . . . 424
8.374.2 Field Documentation . . . 424
8.374.2.1 rsrplevel . . . 424
8.375 LTERSRPThresh Struct Reference . . . 424
8.375.1 Detailed Description . . . 424
8.375.2 Field Documentation . . . 424
8.375.2.1 LTERSRPThreshListLen . . . 424
8.375.2.2 pLTERSRPThreshList . . . 424
8.376 LTERSRQThresh Struct Reference . . . 424
8.376.1 Detailed Description . . . 425
8.376.2 Field Documentation . . . 425
8.376.2.1 LTERSRQThreshListLen . . . 425
8.376.2.2 pLTERSRQThreshList . . . 425
8.377 LTERSSIThresh Struct Reference . . . 425
8.377.1 Detailed Description . . . 425
8.377.2 Field Documentation . . . 425
8.377.2.1 LTERSSIThreshListLen . . . 425

8.377.2.2 pLTERSSIThreshList . . . 426
8.378 LteSccRxInfoResp Struct Reference . . . 426
8.378.1 Detailed Description . . . 426
8.378.2 Field Documentation . . . 426
8.378.2.1 pSccRxInfo . . . 426
8.379 LTESigRptCfg Struct Reference . . . 426
8.379.1 Detailed Description . . . 426
8.379.2 Field Documentation . . . 427
8.379.2.1 avgPeriod . . . 427
8.379.2.2 rptRate . . . 427
8.380 LTESigRptConfig Struct Reference . . . 427
8.380.1 Detailed Description . . . 427
8.380.2 Field Documentation . . . 428
8.380.2.1 avgPeriod . . . 428
8.380.2.2 rptRate . . . 428
8.381 lteSnrinformation Struct Reference . . . 428
8.381.1 Detailed Description . . . 428
8.381.2 Field Documentation . . . 428
8.381.2.1 snrlevel . . . 428
8.382 LTESNRThresh Struct Reference . . . 428
8.382.1 Detailed Description . . . 429
8.382.2 Field Documentation . . . 429
8.382.2.1 LTESNRThresListLen . . . 429
8.382.2.2 pLTESNRThresList . . . 429
8.383 LTESNRThreshold Struct Reference . . . 429
8.383.1 Detailed Description . . . 429
8.383.2 Field Documentation . . . 429
8.383.2.1 LTESNRThreshListLen . . . 429
8.383.2.2 pLTESNRThreshList . . . 430
8.384 LTESSInfo Struct Reference . . . 430
8.384.1 Detailed Description . . . 430
8.384.2 Field Documentation . . . 430
8.384.2.1 rsrp . . . 430
8.384.2.2 rsrq . . . 430
8.384.2.3 rssi . . . 430
8.384.2.4 snr . . . 430
8.385 lteSSInfo Struct Reference . . . 430
8.385.1 Detailed Description . . . 431
8.385.2 Field Documentation . . . 431
8.385.2.1 rsrp . . . 431

8.385.2.2 rsrq . . . 431
8.385.2.3 rssi . . . 431
8.385.2.4 snr . . . 431
8.386 LTESysInfo Struct Reference . . . 431
8.386.1 Detailed Description . . . 432
8.386.2 Field Documentation . . . 433
8.386.2.1 cellId . . . 433
8.386.2.2 cellIdValid . . . 433
8.386.2.3 lac . . . 433
8.386.2.4 lacValid . . . 433
8.386.2.5 MCC . . . 433
8.386.2.6 MNC . . . 433
8.386.2.7 networkIdValid . . . 433
8.386.2.8 regRejectInfoValid . . . 433
8.386.2.9 rejCause . . . 433
8.386.2.10 rejectSrvDomain . . . 433
8.386.2.11 sysInfoLTE . . . 433
8.386.2.12 tac . . . 434
8.386.2.13 tacValid . . . 434
8.387 lteWcdmaCellInfo Struct Reference . . . 434
8.387.1 Detailed Description . . . 434
8.387.2 Field Documentation . . . 434
8.387.2.1 cellReselPriority . . . 434
8.387.2.2 cellsLen . . . 434
8.387.2.3 threshXhigh . . . 435
8.387.2.4 threshXlow . . . 435
8.387.2.5 uarfcn . . . 435
8.387.2.6 WCDMACellInfo . . . 435
8.388 messageModeTlv Struct Reference . . . 435
8.388.1 Detailed Description . . . 435
8.388.2 Field Documentation . . . 435
8.388.2.1 MessageModeInfo . . . 435
8.388.2.2 TlvPresent . . . 435
8.389 messageWaitingInfoContent Struct Reference . . . 435
8.389.1 Detailed Description . . . 435
8.389.2 Field Documentation . . . 436
8.389.2.1 activeInd . . . 436
8.389.2.2 msgCount . . . 436
8.389.2.3 msgType . . . 436
8.390 minBasedIMSI Struct Reference . . . 436

8.390.1 Detailed Description . . . 436
8.390.2 Field Documentation . . . 437
8.390.2.1 imsiM1112 . . . 437
8.390.2.2 imsiMS1 . . . 437
8.390.2.3 imsiMS2 . . . 437
8.390.2.4 mccM . . . 437
8.391 mitigationDevList Struct Reference . . . 437
8.391.1 Detailed Description . . . 437
8.391.2 Field Documentation . . . 437
8.391.2.1 maxMitigationLevel . . . 437
8.391.2.2 mitigationDevId . . . 437
8.391.2.3 mitigationDevIdLen . . . 437
8.392 MNRInfo Struct Reference . . . 437
8.392.1 Detailed Description . . . 438
8.392.2 Field Documentation . . . 438
8.392.2.1 mcc . . . 438
8.392.2.2 mnc . . . 438
8.392.2.3 rat . . . 438
8.393 ModifyProfileIn Struct Reference . . . 438
8.393.1 Detailed Description . . . 438
8.393.2 Field Documentation . . . 439
8.393.2.1 curProfile . . . 439
8.393.2.2 pProfileID . . . 439
8.393.2.3 pProfileType . . . 439
8.394 ModifyProfileOut Struct Reference . . . 439
8.394.1 Detailed Description . . . 439
8.394.2 Field Documentation . . . 439
8.394.2.1 pExtErrorCode . . . 439
8.395 msgWaitingInfo Struct Reference . . . 439
8.395.1 Detailed Description . . . 439
8.395.2 Field Documentation . . . 440
8.395.2.1 msgWaitInfo . . . 440
8.395.2.2 numInstances . . . 440
8.396 mtuSize3gppTlv Struct Reference . . . 440
8.396.1 Detailed Description . . . 440
8.396.2 Field Documentation . . . 440
8.396.2.1 MTUSize3gpp . . . 440
8.396.2.2 TlvPresent . . . 440
8.397 MTUSizeReq Struct Reference . . . 440
8.397.1 Detailed Description . . . 440

8.397.2 Field Documentation . . . 441
8.397.2.1 MTUSize . . . 441
8.398 namName Struct Reference . . . 441
8.398.1 Detailed Description . . . 441
8.398.2 Field Documentation . . . 441
8.398.2.1 namName . . . 441
8.398.2.2 namNameLen . . . 441
8.399 nas_acqOrderPref Struct Reference . . . 441
8.399.1 Detailed Description . . . 441
8.399.2 Field Documentation . . . 442
8.399.2.1 acqOrdeLen . . . 442
8.399.2.2 pAcqOrder . . . 442
8.400 nas_ActPilotPNElement Struct Reference . . . 442
8.400.1 Detailed Description . . . 442
8.400.2 Field Documentation . . . 442
8.400.2.1 ActSetPilotPN . . . 442
8.400.2.2 ActSetPilotPNStrength . . . 442
8.401 nas_AddCDMASysInfo Struct Reference . . . 442
8.401.1 Detailed Description . . . 443
8.401.2 Field Documentation . . . 443
8.401.2.1 geoSysIdx . . . 443
8.401.2.2 regPrd . . . 443
8.402 nas_AddSysInfo Struct Reference . . . 443
8.402.1 Detailed Description . . . 443
8.402.2 Field Documentation . . . 443
8.402.2.1 cellBroadcastCap . . . 443
8.402.2.2 geoSysIdx . . . 444
8.403 nas_CallBarringSysInfo Struct Reference . . . 444
8.403.1 Detailed Description . . . 444
8.403.2 Field Documentation . . . 444
8.403.2.1 csBarStatus . . . 444
8.403.2.2 psBarStatus . . . 444
8.404 nas_callBarStatus Struct Reference . . . 444
8.404.1 Detailed Description . . . 444
8.404.2 Field Documentation . . . 445
8.404.2.1 csBarStatus . . . 445
8.404.2.2 psBarStatus . . . 445
8.405 nas_CDMAChannel Struct Reference . . . 445
8.405.1 Detailed Description . . . 445
8.405.2 Field Documentation . . . 446

8.405.2.1 priChA . . . 446
8.405.2.2 priChB . . . 446
8.405.2.3 secChA . . . 446
8.405.2.4 secChB . . . 446
8.406 nas_CDMAECIOThresh Struct Reference . . . 446
8.406.1 Detailed Description . . . 446
8.406.2 Field Documentation . . . 446
8.406.2.1 CDMAECIOThreshListLen . . . 446
8.406.2.2 pCDMAECIOThreshList . . . 446
8.407 nas_CDMAInfo Struct Reference . . . 446
8.407.1 Detailed Description . . . 447
8.407.2 Field Documentation . . . 447
8.407.2.1 baseId . . . 447
8.407.2.2 baseLat . . . 447
8.407.2.3 baseLong . . . 447
8.407.2.4 nid . . . 447
8.407.2.5 refpn . . . 447
8.407.2.6 sid . . . 447
8.408 nas_CDMARSSIThresh Struct Reference . . . 447
8.408.1 Detailed Description . . . 448
8.408.2 Field Documentation . . . 448
8.408.2.1 CDMARSSIThreshListLen . . . 448
8.408.2.2 pCDMARSSIThreshList . . . 448
8.409 nas_CDMASysInfo Struct Reference . . . 448
8.409.1 Detailed Description . . . 449
8.409.2 Field Documentation . . . 451
8.409.2.1 baseId . . . 451
8.409.2.2 baseLat . . . 451
8.409.2.3 baseLong . . . 451
8.409.2.4 bsInfoValid . . . 451
8.409.2.5 bsPRev . . . 451
8.409.2.6 bsPRevValid . . . 451
8.409.2.7 ccsSupported . . . 451
8.409.2.8 ccsSupportedValid . . . 451
8.409.2.9 cdmaSysIdValid . . . 451
8.409.2.10 isSysPrlMatch . . . 451
8.409.2.11 isSysPrlMatchValid . . . 451
8.409.2.12 MCC . . . 451
8.409.2.13 MNC . . . 451
8.409.2.14 networkID . . . 451

8.409.2.15 networkIdValid . . . 451
8.409.2.16 packetZone . . . 451
8.409.2.17 packetZoneValid . . . 451
8.409.2.18 pRevInUse . . . 452
8.409.2.19 pRevInUseValid . . . 452
8.409.2.20 sysInfoCDMA . . . 452
8.409.2.21 systemID . . . 452
8.410 nas_CDMASysInfoExt Struct Reference . . . 452
8.410.1 Detailed Description . . . 452
8.410.2 Field Documentation . . . 452
8.410.2.1 imsi_11_12 . . . 452
8.410.2.2 MCC . . . 452
8.411 nas_cellParams Struct Reference . . . 452
8.411.1 Detailed Description . . . 452
8.411.2 Field Documentation . . . 453
8.411.2.1 pci . . . 453
8.411.2.2 rsrp . . . 453
8.411.2.3 rsrq . . . 453
8.411.2.4 rssi . . . 453
8.411.2.5 srxlev . . . 453
8.412 nas_ciotAcqOrderPref Struct Reference . . . 453
8.412.1 Detailed Description . . . 453
8.412.2 Field Documentation . . . 454
8.412.2.1 ciotAcqOrderLen . . . 454
8.412.2.2 pCiotAcqOrder . . . 454
8.413 nas_CommInfo Struct Reference . . . 454
8.413.1 Detailed Description . . . 454
8.413.2 Field Documentation . . . 456
8.413.2.1 imsRegState . . . 456
8.413.2.2 modemMode . . . 456
8.413.2.3 psState . . . 456
8.413.2.4 systemMode . . . 456
8.413.2.5 temperature . . . 456
8.414 nas_CSGID Struct Reference . . . 456
8.414.1 Detailed Description . . . 456
8.414.2 Field Documentation . . . 456
8.414.2.1 id . . . 457
8.414.2.2 mcc . . . 457
8.414.2.3 mnc . . . 457
8.414.2.4 mncPcsDigits . . . 457

8.414.2.5 rat . . . 457
8.415 nas_currentPLMN Struct Reference . . . 457
8.415.1 Detailed Description . . . 457
8.415.2 Field Documentation . . . 457
8.415.2.1 MCC . . . 457
8.415.2.2 MNC . . . 457
8.415.2.3 netDescr . . . 458
8.415.2.4 netDescrLength . . . 458
8.416 nas_dataSrvCapabilities Struct Reference . . . 458
8.416.1 Detailed Description . . . 458
8.416.2 Field Documentation . . . 458
8.416.2.1 dataCapabilities . . . 458
8.416.2.2 dataCapabilitiesLen . . . 458
8.417 nas_DataStatusDetail Struct Reference . . . 458
8.417.1 Detailed Description . . . 459
8.417.2 Field Documentation . . . 460
8.417.2.1 IPAddress . . . 460
8.417.2.2 LastErrCode . . . 460
8.418 nas_detailSvcInfo Struct Reference . . . 461
8.418.1 Detailed Description . . . 461
8.418.2 Field Documentation . . . 462
8.418.2.1 hdrHybrid . . . 462
8.418.2.2 hdrSrvStatus . . . 462
8.418.2.3 isSysForbidden . . . 462
8.418.2.4 srvCapability . . . 462
8.418.2.5 srvStatus . . . 462
8.419 nas_DeviceConfigDetail Struct Reference . . . 462
8.419.1 Detailed Description . . . 462
8.419.2 Field Documentation . . . 463
8.419.2.1 Chipset . . . 463
8.419.2.2 HWVersion . . . 463
8.419.2.3 QLIC . . . 463
8.419.2.4 Technology . . . 463
8.420 nas_dirNum Struct Reference . . . 463
8.420.1 Detailed Description . . . 463
8.420.2 Field Documentation . . . 463
8.420.2.1 dirNum . . . 463
8.420.2.2 dirNumLen . . . 463
8.421 nas_DRCParams Struct Reference . . . 463
8.421.1 Detailed Description . . . 464

8.421.2 Field Documentation . . . 464
8.421.2.1 DRCCover . . . 464
8.421.2.2 DRCValue . . . 464
8.422 nas_ecioListElement Struct Reference . . . 464
8.422.1 Detailed Description . . . 464
8.422.2 Field Documentation . . . 464
8.422.2.1 ecio . . . 465
8.422.2.2 radioIf . . . 465
8.423 nas_ECIOThresh Struct Reference . . . 465
8.423.1 Detailed Description . . . 465
8.423.2 Field Documentation . . . 465
8.423.2.1 ECIOThresListLen . . . 465
8.423.2.2 pECIOThresList . . . 465
8.424 nas_errorRateListElement Struct Reference . . . 465
8.424.1 Detailed Description . . . 465
8.424.2 Field Documentation . . . 466
8.424.2.1 errorRate . . . 466
8.424.2.2 radioIf . . . 466
8.425 nas_GERANInfo Struct Reference . . . 466
8.425.1 Detailed Description . . . 467
8.425.2 Field Documentation . . . 468
8.425.2.1 arfcn . . . 468
8.425.2.2 bsic . . . 468
8.425.2.3 cellID . . . 468
8.425.2.4 insNmrCellInfo . . . 468
8.425.2.5 lac . . . 468
8.425.2.6 nmrInst . . . 468
8.425.2.7 plmn . . . 468
8.425.2.8 rxLev . . . 468
8.425.2.9 timingAdvance . . . 468
8.426 nas_geranInstInfo Struct Reference . . . 468
8.426.1 Detailed Description . . . 468
8.426.2 Field Documentation . . . 468
8.426.2.1 geranArfcn . . . 468
8.426.2.2 geranBsicBcc . . . 469
8.426.2.3 geranBsicNcc . . . 469
8.426.2.4 geranRssi . . . 469
8.427 nas_gsmCellInfo Struct Reference . . . 469
8.427.1 Detailed Description . . . 469
8.427.2 Field Documentation . . . 469

8.427.2.1 arfcn . . . 470
8.427.2.2 band1900 . . . 470
8.427.2.3 bsicId . . . 470
8.427.2.4 cellIdValid . . . 470
8.427.2.5 rssi . . . 470
8.427.2.6 srxlev . . . 470
8.428 nas_GSMRSSIThresh Struct Reference . . . 470
8.428.1 Detailed Description . . . 470
8.428.2 Field Documentation . . . 470
8.428.2.1 GSMRSSIThreshListLen . . . 470
8.428.2.2 pGSMRSSIThreshList . . . 470
8.429 nas_GSMSrvStatusInfo Struct Reference . . . 470
8.429.1 Detailed Description . . . 470
8.429.2 Field Documentation . . . 471
8.429.2.1 isPrefDataPath . . . 471
8.429.2.2 srvStatus . . . 471
8.429.2.3 trueSrvStatus . . . 471
8.430 nas_GSMSysInfo Struct Reference . . . 471
8.430.1 Detailed Description . . . 472
8.430.2 Field Documentation . . . 473
8.430.2.1 cellId . . . 473
8.430.2.2 cellIdValid . . . 473
8.430.2.3 dtmSupp . . . 473
8.430.2.4 dtmSuppValid . . . 474
8.430.2.5 egprsSupp . . . 474
8.430.2.6 egprsSuppValid . . . 474
8.430.2.7 lac . . . 474
8.430.2.8 lacValid . . . 474
8.430.2.9 MCC . . . 474
8.430.2.10 MNC . . . 474
8.430.2.11 networkIdValid . . . 474
8.430.2.12 regRejectInfoValid . . . 474
8.430.2.13 rejCause . . . 474
8.430.2.14 rejectSrvDomain . . . 474
8.430.2.15 sysInfoGSM . . . 474
8.431 nas_HDRECIOThresh Struct Reference . . . 474
8.431.1 Detailed Description . . . 474
8.431.2 Field Documentation . . . 474
8.431.2.1 HDRECIOThreshListLen . . . 474
8.431.2.2 pHDRECIOThreshList . . . 474

8.432nas_HDRIOThresh Struct Reference . . . 474
8.432.1 Detailed Description . . . 475
8.432.2 Field Documentation . . . 475
8.432.2.1 HDRIOThreshListLen . . . 475
8.432.2.2 pHDRIOThreshList . . . 475
8.433nas_HDRPersonality_Ind_Data Struct Reference . . . 475
8.433.1 Detailed Description . . . 475
8.433.2 Field Documentation . . . 475
8.433.2.1 pCurrentPersonality . . . 476
8.433.2.2 pPersonalityListLength . . . 476
8.433.2.3 pProtocolSubtypeElement . . . 476
8.434nas_HDRRSSIThresh Struct Reference . . . 476
8.434.1 Detailed Description . . . 476
8.434.2 Field Documentation . . . 476
8.434.2.1 HDRRSSIThreshListLen . . . 476
8.434.2.2 pHDRRSSIThreshList . . . 476
8.435nas_HDRSINRThresh Struct Reference . . . 476
8.435.1 Detailed Description . . . 476
8.435.2 Field Documentation . . . 477
8.435.2.1 HDRSINRThresListLen . . . 477
8.435.2.2 pHDRSINRThresList . . . 477
8.436nas_HDRSINRThreshold Struct Reference . . . 477
8.436.1 Detailed Description . . . 477
8.436.2 Field Documentation . . . 478
8.436.2.1 HDRSINRThreshListLen . . . 478
8.436.2.2 pHDRSINRThreshList . . . 478
8.437nas_HDRSysInfo Struct Reference . . . 478
8.437.1 Detailed Description . . . 478
8.437.2 Field Documentation . . . 479
8.437.2.1 hdrActiveProt . . . 479
8.437.2.2 hdrActiveProtValid . . . 479
8.437.2.3 hdrPersonality . . . 479
8.437.2.4 hdrPersonalityValid . . . 479
8.437.2.5 is856SysId . . . 479
8.437.2.6 is856SysIdValid . . . 479
8.437.2.7 isSysPrlMatch . . . 479
8.437.2.8 isSysPrlMatchValid . . . 479
8.437.2.9 sysInfoHDR . . . 480
8.438nas_homeSIDNID Struct Reference . . . 480
8.438.1 Detailed Description . . . 480

8.438.2 Field Documentation . . . 480
8.438.2.1 numInstances . . . 480
8.438.2.2 SidNid . . . 480
8.439 nas_infoInterFreq Struct Reference . . . 480
8.439.1 Detailed Description . . . 480
8.439.2 Field Documentation . . . 481
8.439.2.1 cell_resel_priority . . . 481
8.439.2.2 cellInterFreqParams . . . 481
8.439.2.3 cells_len . . . 481
8.439.2.4 earfcn . . . 481
8.439.2.5 threshXHigh . . . 481
8.439.2.6 threshXLow . . . 481
8.440 nas_IOThresh Struct Reference . . . 481
8.440.1 Detailed Description . . . 481
8.440.2 Field Documentation . . . 482
8.440.2.1 IOThresListLen . . . 482
8.440.2.2 pIOThresList . . . 482
8.441 nas_lteEARFCN Struct Reference . . . 482
8.441.1 Detailed Description . . . 482
8.441.2 Field Documentation . . . 482
8.441.2.1 earfcn0 . . . 482
8.441.2.2 earfcn1 . . . 482
8.441.2.3 status . . . 483
8.442 nas_lteGsmCellInfo Struct Reference . . . 483
8.442.1 Detailed Description . . . 483
8.442.2 Field Documentation . . . 483
8.442.2.1 cellReselPriority . . . 483
8.442.2.2 cells_len . . . 484
8.442.2.3 GsmCellInfo . . . 484
8.442.2.4 nccPermitted . . . 484
8.442.2.5 threshGsmHigh . . . 484
8.442.2.6 threshGsmLow . . . 484
8.443 nas_LTEInfo Struct Reference . . . 484
8.443.1 Detailed Description . . . 484
8.443.2 Field Documentation . . . 485
8.443.2.1 band . . . 485
8.443.2.2 bandwidth . . . 486
8.443.2.3 emmConnState . . . 486
8.443.2.4 emmState . . . 486
8.443.2.5 emmSubState . . . 486

8.443.2.6 RXChan . . . 486
8.443.2.7 TXChan . . . 486
8.444 nas_LTEInfoInterfreq Struct Reference . . . 486
8.444.1 Detailed Description . . . 486
8.444.2 Field Documentation . . . 486
8.444.2.1 freqsLen . . . 486
8.444.2.2 InfoInterfreq . . . 486
8.444.2.3 ueInIdle . . . 486
8.445 nas_LTEInfoIntrafreq Struct Reference . . . 486
8.445.1 Detailed Description . . . 487
8.445.2 Field Documentation . . . 488
8.445.2.1 CellParams . . . 488
8.445.2.2 cellReselPriority . . . 488
8.445.2.3 cellsLen . . . 488
8.445.2.4 earfcn . . . 488
8.445.2.5 globalCellId . . . 488
8.445.2.6 plmn . . . 488
8.445.2.7 servingCellId . . . 488
8.445.2.8 sIntraSearch . . . 488
8.445.2.9 sNonIntraSearch . . . 488
8.445.2.10 tac . . . 488
8.445.2.11 threshServingLow . . . 488
8.445.2.12 ueInIdle . . . 488
8.446 nas_LTEInfoNeighboringGSM Struct Reference . . . 488
8.446.1 Detailed Description . . . 489
8.446.2 Field Documentation . . . 489
8.446.2.1 freqsLen . . . 489
8.446.2.2 LteGsmCellInfo . . . 489
8.446.2.3 ueInIdle . . . 489
8.447 nas_LTEInfoNeighboringWCDMA Struct Reference . . . 489
8.447.1 Detailed Description . . . 489
8.447.2 Field Documentation . . . 490
8.447.2.1 freqsLen . . . 490
8.447.2.2 LTEWCDMACellInfo . . . 490
8.447.2.3 ueInIdle . . . 490
8.448 nas_LTEOperationMode Struct Reference . . . 490
8.448.1 Detailed Description . . . 490
8.448.2 Field Documentation . . . 490
8.448.2.1 pLTEOperationMode . . . 490
8.448.2.2 TlvPresent . . . 490

8.449 nas_ltePCI Struct Reference . . . 490
8.449.1 Detailed Description . . . 491
8.449.2 Field Documentation . . . 491
8.449.2.1 earfcn . . . 491
8.449.2.2 pci . . . 491
8.449.2.3 status . . . 491
8.450 nas_lteRsrpinformation Struct Reference . . . 491
8.450.1 Detailed Description . . . 491
8.450.2 Field Documentation . . . 491
8.450.2.1 rsrplevel . . . 491
8.451 nas_LTERSRPThresh Struct Reference . . . 491
8.451.1 Detailed Description . . . 492
8.451.2 Field Documentation . . . 492
8.451.2.1 LTERSRPThreshListLen . . . 492
8.451.2.2 pLTERSRPThreshList . . . 492
8.452 nas_LTERSRQThresh Struct Reference . . . 492
8.452.1 Detailed Description . . . 492
8.452.2 Field Documentation . . . 492
8.452.2.1 LTERSRQThreshListLen . . . 492
8.452.2.2 pLTERSRQThreshList . . . 492
8.453 nas_LTERSSIThresh Struct Reference . . . 492
8.453.1 Detailed Description . . . 493
8.453.2 Field Documentation . . . 493
8.453.2.1 LTERSSIThreshListLen . . . 493
8.453.2.2 pLTERSSIThreshList . . . 493
8.454 nas_LTESigRptCfg Struct Reference . . . 493
8.454.1 Detailed Description . . . 493
8.454.2 Field Documentation . . . 494
8.454.2.1 avgPeriod . . . 494
8.454.2.2 rptRate . . . 494
8.455 nas_LTESigRptConfig Struct Reference . . . 494
8.455.1 Detailed Description . . . 494
8.455.2 Field Documentation . . . 495
8.455.2.1 avgPeriod . . . 495
8.455.2.2 rptRate . . . 495
8.456 nas_lteSnrinformation Struct Reference . . . 495
8.456.1 Detailed Description . . . 495
8.456.2 Field Documentation . . . 495
8.456.2.1 snrlevel . . . 495
8.457 nas_LTESNRThresh Struct Reference . . . 495

8.457.1 Detailed Description . . . 496
8.457.2 Field Documentation . . . 496
8.457.2.1 LTESNRThresListLen . . . 496
8.457.2.2 pLTESNRThresList . . . 496
8.458 nas_LTESNRThreshold Struct Reference . . . 496
8.458.1 Detailed Description . . . 496
8.458.2 Field Documentation . . . 496
8.458.2.1 LTESNRThreshListLen . . . 496
8.458.2.2 pLTESNRThreshList . . . 497
8.459 nas_LTESysInfo Struct Reference . . . 497
8.459.1 Detailed Description . . . 497
8.459.2 Field Documentation . . . 498
8.459.2.1 cellId . . . 499
8.459.2.2 cellIdValid . . . 499
8.459.2.3 lac . . . 499
8.459.2.4 lacValid . . . 499
8.459.2.5 MCC . . . 499
8.459.2.6 MNC . . . 499
8.459.2.7 networkIdValid . . . 499
8.459.2.8 regRejectInfoValid . . . 499
8.459.2.9 rejCause . . . 499
8.459.2.10 rejectSrvDomain . . . 499
8.459.2.11 sysInfoLTE . . . 499
8.459.2.12 tac . . . 499
8.459.2.13 tacValid . . . 499
8.460 nas_lteWcdmaCellInfo Struct Reference . . . 499
8.460.1 Detailed Description . . . 499
8.460.2 Field Documentation . . . 500
8.460.2.1 cellReselPriority . . . 500
8.460.2.2 cellsLen . . . 500
8.460.2.3 threshXhigh . . . 500
8.460.2.4 threshXlow . . . 500
8.460.2.5 uarfcn . . . 500
8.460.2.6 WCDMACellInfo . . . 500
8.461 nas_minBasedIMSI Struct Reference . . . 500
8.461.1 Detailed Description . . . 500
8.461.2 Field Documentation . . . 501
8.461.2.1 imsiM1112 . . . 501
8.461.2.2 imsiMS1 . . . 501
8.461.2.3 imsiMS2 . . . 501

8.461.2.4 mccM . . . 501
8.462 nas_MNRInfo Struct Reference . . . 501
8.462.1 Detailed Description . . . 501
8.462.2 Field Documentation . . . 501
8.462.2.1 mcc . . . 501
8.462.2.2 mnc . . . 501
8.462.2.3 rat . . . 501
8.463 nas_namName Struct Reference . . . 502
8.463.1 Detailed Description . . . 502
8.463.2 Field Documentation . . . 502
8.463.2.1 namName . . . 502
8.463.2.2 namNameLen . . . 502
8.464 nas_netSelectionPref Struct Reference . . . 502
8.464.1 Detailed Description . . . 502
8.464.2 Field Documentation . . . 503
8.464.2.1 mcc . . . 503
8.464.2.2 mnc . . . 503
8.464.2.3 netReg . . . 503
8.465 nas_NetworkStat1x Struct Reference . . . 503
8.465.1 Detailed Description . . . 503
8.465.2 Field Documentation . . . 505
8.465.2.1 ActSetCnt . . . 505
8.465.2.2 NeighborSetCnt . . . 505
8.465.2.3 pActPilotPNElements . . . 505
8.465.2.4 pNeighborSetPilotPN . . . 505
8.465.2.5 RX_EC_IO . . . 505
8.465.2.6 RX_PWR . . . 505
8.465.2.7 SO . . . 505
8.465.2.8 State . . . 505
8.465.2.9 TX_PWR . . . 505
8.466 nas_NetworkStatEVDO Struct Reference . . . 505
8.466.1 Detailed Description . . . 506
8.466.2 Field Documentation . . . 507
8.466.2.1 MACIndex . . . 507
8.466.2.2 PER . . . 507
8.466.2.3 PilotEnergy . . . 507
8.466.2.4 pSectorID . . . 507
8.466.2.5 RX_PWR . . . 507
8.466.2.6 SectorIDLen . . . 507
8.466.2.7 SNR . . . 507

8.466.2.8 State . . . 507
8.467 nas_nmrCellInfo Struct Reference . . . 507
8.467.1 Detailed Description . . . 507
8.467.2 Field Documentation . . . 508
8.467.2.1 nmrArfcn . . . 508
8.467.2.2 nmrBsic . . . 508
8.467.2.3 nmrCellID . . . 508
8.467.2.4 nmrLac . . . 508
8.467.2.5 nmrPlmn . . . 508
8.467.2.6 nmrRxLev . . . 508
8.468 nas_nr5gBandPref Struct Reference . . . 508
8.468.1 Detailed Description . . . 508
8.468.2 Field Documentation . . . 509
8.468.2.1 bits_129_192 . . . 509
8.468.2.2 bits_193_256 . . . 509
8.468.2.3 bits_1_64 . . . 509
8.468.2.4 bits_65_128 . . . 509
8.469 nas_operatorNameString Struct Reference . . . 509
8.469.1 Detailed Description . . . 509
8.469.2 Field Documentation . . . 509
8.469.2.1 PLMNName . . . 509
8.470 nas_OperatorPLMNData Struct Reference . . . 509
8.470.1 Detailed Description . . . 509
8.470.2 Field Documentation . . . 510
8.470.2.1 lac1 . . . 510
8.470.2.2 lac2 . . . 510
8.470.2.3 mcc . . . 510
8.470.2.4 mnc . . . 510
8.470.2.5 PLMNRecID . . . 510
8.471 nas_operatorPLMNList Struct Reference . . . 510
8.471.1 Detailed Description . . . 510
8.471.2 Field Documentation . . . 511
8.471.2.1 numInstance . . . 511
8.471.2.2 PLMNData . . . 511
8.472 nas_PhyCaAggPcellInfo Struct Reference . . . 511
8.472.1 Detailed Description . . . 511
8.472.2 Field Documentation . . . 511
8.472.2.1 dl_bw_value . . . 511
8.472.2.2 freq . . . 511
8.472.2.3 iLTEbandValue . . . 511

8.472.2.4 pci . . . 511
8.472.2.5 TlvPresent . . . 511
8.473nas_PhyCaAggScellDlBw Struct Reference . . . 512
8.473.1 Detailed Description . . . 512
8.473.2 Field Documentation . . . 512
8.473.2.1 dl_bw_value . . . 512
8.473.2.2 TlvPresent . . . 512
8.474nas_PhyCaAggScellIndex Struct Reference . . . 512
8.474.1 Detailed Description . . . 512
8.474.2 Field Documentation . . . 512
8.474.2.1 scell_idx . . . 512
8.474.2.2 TlvPresent . . . 512
8.475nas_PhyCaAggScellIndType Struct Reference . . . 513
8.475.1 Detailed Description . . . 513
8.475.2 Field Documentation . . . 513
8.475.2.1 freq . . . 513
8.475.2.2 pci . . . 513
8.475.2.3 scell_state . . . 513
8.475.2.4 TlvPresent . . . 513
8.476nas_PhyCaAggScellInfo Struct Reference . . . 513
8.476.1 Detailed Description . . . 514
8.476.2 Field Documentation . . . 516
8.476.2.1 dl_bw_value . . . 516
8.476.2.2 freq . . . 516
8.476.2.3 iLTEbandValue . . . 516
8.476.2.4 pci . . . 516
8.476.2.5 scell_state . . . 516
8.476.2.6 TlvPresent . . . 516
8.477nas_PilotSetData Struct Reference . . . 516
8.477.1 Detailed Description . . . 516
8.477.2 Field Documentation . . . 516
8.477.2.1 NumPilots . . . 516
8.477.2.2 pPilotSetInfo . . . 516
8.478nas_PilotSetParams Struct Reference . . . 517
8.478.1 Detailed Description . . . 517
8.478.2 Field Documentation . . . 517
8.478.2.1 PilotPN . . . 517
8.478.2.2 PilotStrength . . . 517
8.478.2.3 PilotType . . . 517
8.479nas_PLMNNetworkName Struct Reference . . . 517

8.479.1 Detailed Description . . . 517
8.479.2 Field Documentation . . . 518
8.479.2.1 numInstance . . . 518
8.479.2.2 PLMNNetName . . . 518
8.480 nas_PLMNNetworkNameData Struct Reference . . . 518
8.480.1 Detailed Description . . . 518
8.480.2 Field Documentation . . . 519
8.480.2.1 codingScheme . . . 519
8.480.2.2 countryInitials . . . 519
8.480.2.3 longName . . . 519
8.480.2.4 longNameLen . . . 519
8.480.2.5 longNameSpareBits . . . 519
8.480.2.6 shortName . . . 519
8.480.2.7 shortNameLen . . . 519
8.480.2.8 shortNameSpareBits . . . 519
8.481 nas_protocolSubtypeElement Struct Reference . . . 519
8.481.1 Detailed Description . . . 520
8.481.2 Field Documentation . . . 520
8.481.2.1 AccessMac . . . 521
8.481.2.2 AuthProt . . . 521
8.481.2.3 ControlMac . . . 521
8.481.2.4 EncryptProt . . . 521
8.481.2.5 ForwardMac . . . 521
8.481.2.6 IdleState . . . 521
8.481.2.7 KeyExchange . . . 521
8.481.2.8 MultDisc . . . 521
8.481.2.9 PhysicalLayer . . . 521
8.481.2.10 ReverseMac . . . 521
8.481.2.11 SecProt . . . 521
8.481.2.12 VirtStream . . . 521
8.482 nas_qaQmi3Gpp2TimeZone Struct Reference . . . 521
8.482.1 Detailed Description . . . 521
8.482.2 Field Documentation . . . 521
8.482.2.1 daylightSavings . . . 522
8.482.2.2 leapSeconds . . . 522
8.482.2.3 localTimeOffset . . . 522
8.483 nas_QmiNas3GppNetworkInfo Struct Reference . . . 522
8.483.1 Detailed Description . . . 522
8.483.2 Field Documentation . . . 523
8.483.2.1 Desription . . . 523

8.483.2.2 Forbidden . . . 523
8.483.2.3 InUse . . . 523
8.483.2.4 MCC . . . 523
8.483.2.5 MNC . . . 523
8.483.2.6 Preferred . . . 523
8.483.2.7 Roaming . . . 523
8.484 nas_QmiNas3GppNetworkRAT Struct Reference . . . 523
8.484.1 Detailed Description . . . 523
8.484.2 Field Documentation . . . 523
8.484.2.1 MCC . . . 524
8.484.2.2 MNC . . . 524
8.484.2.3 RAT . . . 524
8.485 nas_QmisNasPcsDigit Struct Reference . . . 524
8.485.1 Detailed Description . . . 524
8.485.2 Field Documentation . . . 524
8.485.2.1 includes_pcs_digit . . . 524
8.485.2.2 MCC . . . 524
8.485.2.3 MNC . . . 524
8.486 nas_RankIndicatorTlv Struct Reference . . . 524
8.486.1 Detailed Description . . . 524
8.486.2 Field Documentation . . . 525
8.486.2.1 count0 . . . 525
8.486.2.2 count1 . . . 525
8.486.2.3 TlvPresent . . . 525
8.487 nas_RejectReasonTlv Struct Reference . . . 525
8.487.1 Detailed Description . . . 525
8.487.2 Field Documentation . . . 525
8.487.2.1 rejectCause . . . 525
8.487.2.2 serviceDomain . . . 525
8.487.2.3 TlvPresent . . . 525
8.488 nas_RFBandInfoElements Struct Reference . . . 525
8.488.1 Detailed Description . . . 526
8.488.2 Field Documentation . . . 526
8.488.2.1 activeBandClass . . . 526
8.488.2.2 activeChannel . . . 526
8.488.2.3 radioInterface . . . 526
8.489 nas_RfBandInfoExtFormat Struct Reference . . . 526
8.489.1 Detailed Description . . . 526
8.489.2 Field Documentation . . . 527
8.489.2.1 pInstancesSize . . . 527

8.489.2.2 pRfBandInfoExtFormatParam . . . 527
8.489.2.3 TlvPresent . . . 527
8.490 nas_RfBandInfoExtFormatElements Struct Reference . . . 527
8.490.1 Detailed Description . . . 527
8.490.2 Field Documentation . . . 527
8.490.2.1 activeBand . . . 527
8.490.2.2 activeChannel . . . 527
8.490.2.3 radioInterface . . . 527
8.491 nas_RfBandInfoList Struct Reference . . . 527
8.491.1 Detailed Description . . . 527
8.491.2 Field Documentation . . . 528
8.491.2.1 pInstanceSize . . . 528
8.491.2.2 pRFBandInfoParam . . . 528
8.492 nas_RfBandwidthInfo Struct Reference . . . 528
8.492.1 Detailed Description . . . 528
8.492.2 Field Documentation . . . 528
8.492.2.1 pInstancesSize . . . 528
8.492.2.2 pRfBandwidthInfoParam . . . 528
8.492.2.3 TlvPresent . . . 528
8.493 nas_RfBandwidthInfoElements Struct Reference . . . 529
8.493.1 Detailed Description . . . 529
8.493.2 Field Documentation . . . 529
8.493.2.1 bandwidth . . . 529
8.493.2.2 radioInterface . . . 529
8.494 nas_RfDedicatedBandInfo Struct Reference . . . 529
8.494.1 Detailed Description . . . 529
8.494.2 Field Documentation . . . 530
8.494.2.1 pInstancesSize . . . 530
8.494.2.2 pRfDedicatedBandInfoParam . . . 530
8.494.2.3 TlvPresent . . . 530
8.495 nas_RfDedicatedBandInfoElements Struct Reference . . . 530
8.495.1 Detailed Description . . . 530
8.495.2 Field Documentation . . . 530
8.495.2.1 dedicatedBand . . . 530
8.495.2.2 radioInterface . . . 530
8.496 nas_RFInfoTlv Struct Reference . . . 530
8.496.1 Detailed Description . . . 531
8.496.2 Field Documentation . . . 531
8.496.2.1 activeBandClass . . . 531
8.496.2.2 activeChannel . . . 531

8.496.2.3 radioInterface . . . 531
8.496.2.4 radioInterfaceSize . . . 531
8.496.2.5 TlvPresent . . . 531
8.497 nas_roamIndList Struct Reference . . . 532
8.497.1 Detailed Description . . . 532
8.497.2 Field Documentation . . . 532
8.497.2.1 numInstances . . . 532
8.497.2.2 radioInterface . . . 532
8.497.2.3 roamIndicator . . . 532
8.498 nas_RSRPThresh Struct Reference . . . 532
8.498.1 Detailed Description . . . 533
8.498.2 Field Documentation . . . 533
8.498.2.1 pRSRPThresList . . . 533
8.498.2.2 RSRPThresListLen . . . 533
8.499 nas_rsrqInformation Struct Reference . . . 533
8.499.1 Detailed Description . . . 533
8.499.2 Field Documentation . . . 533
8.499.2.1 radioIf . . . 533
8.499.2.2 rsrq . . . 533
8.500 nas_RSRQThresh Struct Reference . . . 534
8.500.1 Detailed Description . . . 534
8.500.2 Field Documentation . . . 534
8.500.2.1 pRSRQThresList . . . 534
8.500.2.2 RSRQThresListLen . . . 534
8.501 nas_RSSIThresh Struct Reference . . . 534
8.501.1 Detailed Description . . . 534
8.501.2 Field Documentation . . . 535
8.501.2.1 pRSSIThresList . . . 535
8.501.2.2 RSSIThresListLen . . . 535
8.502 nas_rxInfo Struct Reference . . . 535
8.502.1 Detailed Description . . . 535
8.502.2 Field Documentation . . . 536
8.502.2.1 ecio . . . 536
8.502.2.2 isRadioTuned . . . 536
8.502.2.3 phase . . . 536
8.502.2.4 rscp . . . 536
8.502.2.5 rsrp . . . 536
8.502.2.6 rxPower . . . 536
8.503 nas_RxSigInfo Struct Reference . . . 536
8.503.1 Detailed Description . . . 536

8.503.2 Field Documentation . . . 537
8.503.2.1 isRadioTuned . . . 537
8.503.2.2 rsrp . . . 537
8.503.2.3 rxChainIndex . . . 537
8.503.2.4 rxPower . . . 537
8.504 nas_rxSignalStrengthListElement Struct Reference . . . 537
8.504.1 Detailed Description . . . 537
8.504.2 Field Documentation . . . 538
8.504.2.1 radioIf . . . 538
8.504.2.2 rxSignalStrength . . . 538
8.505 nas_SccRxInfo Struct Reference . . . 538
8.505.1 Detailed Description . . . 538
8.505.2 Field Documentation . . . 539
8.505.2.1 numInstances . . . 539
8.505.2.2 rsrq . . . 539
8.505.2.3 sigInfo . . . 539
8.505.2.4 snr . . . 539
8.505.2.5 TlvPresent . . . 539
8.506 nas_serviceProviderName Struct Reference . . . 539
8.506.1 Detailed Description . . . 539
8.506.2 Field Documentation . . . 539
8.506.2.1 displayCondition . . . 539
8.506.2.2 spn . . . 539
8.506.2.3 spnLength . . . 539
8.507 nas_servSystem Struct Reference . . . 540
8.507.1 Detailed Description . . . 540
8.507.2 Field Documentation . . . 541
8.507.2.1 csAttachState . . . 541
8.507.2.2 numRadioInterfaces . . . 541
8.507.2.3 psAttachState . . . 541
8.507.2.4 radioInterface . . . 541
8.507.2.5 regState . . . 541
8.507.2.6 selNetwork . . . 541
8.508 nas_sidNid Struct Reference . . . 541
8.508.1 Detailed Description . . . 541
8.508.2 Field Documentation . . . 541
8.508.2.1 nid . . . 541
8.508.2.2 sid . . . 541
8.509 nas_SignalStrengthTlv Struct Reference . . . 542
8.509.1 Detailed Description . . . 542

8.509.2 Field Documentation . . . 542
8.509.2.1 radioInterface . . . 542
8.509.2.2 signalStrength . . . 542
8.509.2.3 TlvPresent . . . 542
8.510 nas_SLQSSignalStrengthsIndReq Struct Reference . . . 542
8.510.1 Detailed Description . . . 542
8.510.2 Field Documentation . . . 543
8.510.2.1 ecioDelta . . . 543
8.510.2.2 ecioThresholdList . . . 543
8.510.2.3 ecioThresholdListLen . . . 544
8.510.2.4 ioDelta . . . 544
8.510.2.5 lteRsrpDelta . . . 544
8.510.2.6 lteSnrDelta . . . 544
8.510.2.7 rsrqDelta . . . 544
8.510.2.8 rxSignalStrengthDelta . . . 544
8.510.2.9 sinrDelta . . . 544
8.510.2.10 sinrThresholdList . . . 544
8.510.2.11 sinrThresholdListLen . . . 544
8.511 nas_SLQSSignalStrengthsInformation Struct Reference . . . 544
8.511.1 Detailed Description . . . 544
8.511.2 Field Documentation . . . 544
8.511.2.1 ecioInfo . . . 544
8.511.2.2 errorRateInfo . . . 544
8.511.2.3 io . . . 544
8.511.2.4 lteRsrpinfo . . . 545
8.511.2.5 lteSnrinfo . . . 545
8.511.2.6 rsrqInfo . . . 545
8.511.2.7 rxSignalStrengthInfo . . . 545
8.511.2.8 sinr . . . 545
8.512 nas_SLQSSignalStrengthsTlv Struct Reference . . . 545
8.512.1 Detailed Description . . . 545
8.512.2 Field Documentation . . . 545
8.512.2.1 sSLQSSignalStrengthsInfo . . . 545
8.512.2.2 TlvPresent . . . 545
8.513 nas_SrvStatusInfo Struct Reference . . . 545
8.513.1 Detailed Description . . . 545
8.513.2 Field Documentation . . . 546
8.513.2.1 isPrefDataPath . . . 546
8.513.2.2 srvStatus . . . 546
8.514 nas_sysInfoCommon Struct Reference . . . 546

8.514.1 Detailed Description . . . 546
8.514.2 Field Documentation . . . 548
8.514.2.1 isSysForbidden . . . 548
8.514.2.2 isSysForbiddenValid . . . 548
8.514.2.3 roamStatus . . . 548
8.514.2.4 roamStatusValid . . . 548
8.514.2.5 srvCapability . . . 548
8.514.2.6 srvCapabilityValid . . . 548
8.514.2.7 srvDomain . . . 548
8.514.2.8 srvDomainValid . . . 548
8.515 nas_TDSCDMAECIOThresh Struct Reference . . . 549
8.515.1 Detailed Description . . . 549
8.515.2 Field Documentation . . . 549
8.515.2.1 pTDSCDMAECIOThreshList . . . 549
8.515.2.2 TDSCDMAECIOThreshListLen . . . 549
8.516 nas_TDSCDMARSCPThresh Struct Reference . . . 549
8.516.1 Detailed Description . . . 549
8.516.2 Field Documentation . . . 549
8.516.2.1 pTDSCDMARSCPThreshList . . . 549
8.516.2.2 TDSCDMARSCPThreshListLen . . . 550
8.517 nas_TDSCDMARSSIThresh Struct Reference . . . 550
8.517.1 Detailed Description . . . 550
8.517.2 Field Documentation . . . 550
8.517.2.1 pTDSCDMARSSIThreshList . . . 550
8.517.2.2 TDSCDMARSSIThreshListLen . . . 550
8.518 nas_TDSCDMASINRCONFThresh Struct Reference . . . 550
8.518.1 Detailed Description . . . 550
8.518.2 Field Documentation . . . 550
8.518.2.1 pTDSCDMASINRCONFThreshList . . . 551
8.518.2.2 TDSCDMASINRCONFThreshListLen . . . 551
8.519 nas_TDSCDMASINRThresh Struct Reference . . . 551
8.519.1 Detailed Description . . . 551
8.519.2 Field Documentation . . . 551
8.519.2.1 pTDSCDMASINRThreshList . . . 551
8.519.2.2 TDSCDMASINRThreshListLen . . . 551
8.520 nas_timeInfo Struct Reference . . . 551
8.520.1 Detailed Description . . . 551
8.520.2 Field Documentation . . . 553
8.520.2.1 day . . . 553
8.520.2.2 dayLtSavingAdj . . . 553

8.520.2.3 dayOfWeek . . . 553
8.520.2.4 hour . . . 553
8.520.2.5 minute . . . 553
8.520.2.6 month . . . 553
8.520.2.7 radioInterface . . . 553
8.520.2.8 second . . . 553
8.520.2.9 timeZone . . . 553
8.520.2.10 TlvPresent . . . 553
8.520.2.11 year . . . 553
8.521 nas_trueIMSI Struct Reference . . . 553
8.521.1 Detailed Description . . . 553
8.521.2 Field Documentation . . . 554
8.521.2.1 imsiT1112 . . . 554
8.521.2.2 imsiTaddrNum . . . 554
8.521.2.3 imsiTS1 . . . 554
8.521.2.4 imsiTS2 . . . 554
8.521.2.5 mccT . . . 554
8.522 nas_txInfo Struct Reference . . . 554
8.522.1 Detailed Description . . . 554
8.522.2 Field Documentation . . . 554
8.522.2.1 isInTraffic . . . 554
8.522.2.2 txPower . . . 554
8.523 nas_UMTSInfo Struct Reference . . . 554
8.523.1 Detailed Description . . . 555
8.523.2 Field Documentation . . . 556
8.523.2.1 cellID . . . 556
8.523.2.2 ecio . . . 556
8.523.2.3 geranInst . . . 556
8.523.2.4 GeranInstInfo . . . 556
8.523.2.5 lac . . . 556
8.523.2.6 plmn . . . 556
8.523.2.7 psc . . . 556
8.523.2.8 rscp . . . 556
8.523.2.9 uarfcn . . . 556
8.523.2.10 umtsInst . . . 556
8.523.2.11 UMTSInstInfo . . . 556
8.524 nas_UMTSinstInfo Struct Reference . . . 556
8.524.1 Detailed Description . . . 556
8.524.2 Field Documentation . . . 557
8.524.2.1 umtsEcio . . . 557

8.524.2.2 umtsPsc . . . 557
8.524.2.3 umtsRscp . . . 557
8.524.2.4 umtsUarfcn . . . 557
8.525 nas_umtsLTENbrCell Struct Reference . . . 557
8.525.1 Detailed Description . . . 557
8.525.2 Field Documentation . . . 558
8.525.2.1 cellIsTDD . . . 558
8.525.2.2 earfcn . . . 558
8.525.2.3 pci . . . 558
8.525.2.4 rsrp . . . 558
8.525.2.5 rsrq . . . 558
8.525.2.6 srxlev . . . 558
8.526 nas_UniversalTime Struct Reference . . . 558
8.526.1 Detailed Description . . . 558
8.526.2 Field Documentation . . . 559
8.526.2.1 day . . . 559
8.526.2.2 dayOfWeek . . . 559
8.526.2.3 hour . . . 559
8.526.2.4 minute . . . 559
8.526.2.5 month . . . 559
8.526.2.6 second . . . 559
8.526.2.7 year . . . 559
8.527 nas_wcdmaCellInfo Struct Reference . . . 559
8.527.1 Detailed Description . . . 559
8.527.2 Field Documentation . . . 560
8.527.2.1 cpich_ecno . . . 560
8.527.2.2 cpich_rscp . . . 560
8.527.2.3 psc . . . 560
8.527.2.4 srxlev . . . 560
8.528 nas_WCDMAECIOThresh Struct Reference . . . 560
8.528.1 Detailed Description . . . 560
8.528.2 Field Documentation . . . 560
8.528.2.1 pWCDMAECIOThreshList . . . 560
8.528.2.2 WCDMAECIOThreshListLen . . . 560
8.529 nas_WCDMAInfoLTENeighborCell Struct Reference . . . 560
8.529.1 Detailed Description . . . 561
8.529.2 Field Documentation . . . 561
8.529.2.1 UMTSLTENbrCell . . . 561
8.529.2.2 umtsLTENbrCellLen . . . 561
8.529.2.3 wcdmaRRCState . . . 561

8.530 nas_WCDMARSSIThresh Struct Reference . . . 561
8.530.1 Detailed Description . . . 561
8.530.2 Field Documentation . . . 562
8.530.2.1 pWCDMARSSIThreshList . . . 562
8.530.2.2 WCDMARSSIThreshListLen . . . 562
8.531 nas_WCDMASysInfo Struct Reference . . . 562
8.531.1 Detailed Description . . . 562
8.531.2 Field Documentation . . . 564
8.531.2.1 cellId . . . 565
8.531.2.2 cellIdValid . . . 565
8.531.2.3 hsCallStatus . . . 565
8.531.2.4 hsCallStatusValid . . . 565
8.531.2.5 hsInd . . . 565
8.531.2.6 hsIndValid . . . 565
8.531.2.7 lac . . . 565
8.531.2.8 lacValid . . . 565
8.531.2.9 MCC . . . 565
8.531.2.10 MNC . . . 565
8.531.2.11 networkIdValid . . . 565
8.531.2.12 psc . . . 565
8.531.2.13 pscValid . . . 565
8.531.2.14 regRejectInfoValid . . . 565
8.531.2.15 rejCause . . . 565
8.531.2.16 rejectSrvDomain . . . 565
8.531.2.17 sysInfoWCDMA . . . 565
8.532 nas_wcdmaUARFCN Struct Reference . . . 565
8.532.1 Detailed Description . . . 565
8.532.2 Field Documentation . . . 566
8.532.2.1 status . . . 566
8.532.2.2 uarfcn . . . 566
8.533 NASBandPreferenceTlv Struct Reference . . . 566
8.533.1 Detailed Description . . . 566
8.533.2 Field Documentation . . . 566
8.533.2.1 band_pref . . . 566
8.533.2.2 TlvPresent . . . 566
8.534 nasCellLocationInfoResp Struct Reference . . . 566
8.534.1 Detailed Description . . . 566
8.534.2 Field Documentation . . . 567
8.534.2.1 pCDMAInfo . . . 567
8.534.2.2 pGERANInfo . . . 567

8.534.2.3 pLTEInfoInterfreq . . . 567
8.534.2.4 pLTEInfoIntrafreq . . . 567
8.534.2.5 pLTEInfoNeighboringGSM . . . 567
8.534.2.6 pLTEInfoNeighboringWCDMA . . . 567
8.534.2.7 pUMTSCellID . . . 567
8.534.2.8 pUMTSInfo . . . 567
8.534.2.9 pWCDMAInfoLTENeighborCell . . . 567
8.535 NASEmergencyModeTlv Struct Reference . . . 567
8.535.1 Detailed Description . . . 568
8.535.2 Field Documentation . . . 568
8.535.2.1 EmerMode . . . 568
8.535.2.2 TlvPresent . . . 568
8.536 nasGet3GPP2SubscriptionInfoReq Struct Reference . . . 568
8.536.1 Detailed Description . . . 568
8.536.2 Field Documentation . . . 568
8.536.2.1 namID . . . 568
8.537 nasGet3GPP2SubscriptionInfoResp Struct Reference . . . 568
8.537.1 Detailed Description . . . 568
8.537.2 Field Documentation . . . 569
8.537.2.1 pCDMAChannel . . . 569
8.537.2.2 pDirNum . . . 569
8.537.2.3 pHomeSIDNID . . . 569
8.537.2.4 pMinBasedIMSI . . . 569
8.537.2.5 pNAMNameInfo . . . 569
8.537.2.6 pTrueIMSI . . . 569
8.538 nasGeteDRXParamsResp Struct Reference . . . 569
8.538.1 Detailed Description . . . 569
8.538.2 Field Documentation . . . 570
8.538.2.1 pCycleLen . . . 570
8.538.2.2 pEdrxEnable . . . 570
8.538.2.3 pPagingTimeWindow . . . 570
8.539 nasGetHDRColorCodeResp Struct Reference . . . 570
8.539.1 Detailed Description . . . 570
8.539.2 Field Documentation . . . 570
8.539.2.1 pColorCode . . . 570
8.540 nasGetLTECphyCa Struct Reference . . . 570
8.540.1 Field Documentation . . . 571
8.540.1.1 sPhyCaAggPcellInfo . . . 571
8.540.1.2 sPhyCaAggScellDlBw . . . 571
8.540.1.3 sPhyCaAggScellIndex . . . 571

8.540.1.4 sPhyCaAggScellIndType . . . 571
8.540.1.5 sPhyCaAggScellInfo . . . 571
8.541 NasGetLTECphyCaInfo Struct Reference . . . 571
8.541.1 Detailed Description . . . 571
8.541.2 Field Documentation . . . 571
8.541.2.1 PhyCaAggPcellInfo . . . 571
8.541.2.2 PhyCaAggScellDlBw . . . 571
8.541.2.3 PhyCaAggScellIndex . . . 572
8.541.2.4 PhyCaAggScellIndType . . . 572
8.541.2.5 PhyCaAggScellInfo . . . 572
8.542 nasGetLTECphyCaResp Struct Reference . . . 572
8.542.1 Field Documentation . . . 572
8.542.1.1 pPhyCaAggPcellInfo . . . 572
8.542.1.2 pPhyCaAggScellDlBw . . . 572
8.542.1.3 pPhyCaAggScellIndex . . . 572
8.542.1.4 pPhyCaAggScellIndType . . . 572
8.542.1.5 pPhyCaAggScellInfo . . . 572
8.543 NasGetRFInfoResp Struct Reference . . . 572
8.543.1 Detailed Description . . . 572
8.543.2 Field Documentation . . . 573
8.543.2.1 pLTEOperationMode . . . 573
8.543.2.2 pRfBandInfoExtFormat . . . 573
8.543.2.3 pRfBandwidthInfo . . . 573
8.543.2.4 pRfDedicatedBandInfo . . . 573
8.543.2.5 rfbandInfoList . . . 573
8.544 nasGetSigInfoResp Struct Reference . . . 573
8.544.1 Detailed Description . . . 573
8.544.2 Field Documentation . . . 574
8.544.2.1 pCDMASSInfo . . . 574
8.544.2.2 pGSMSSInfo . . . 574
8.544.2.3 pHDRSSInfo . . . 574
8.544.2.4 pLTESSInfo . . . 574
8.544.2.5 pTDSCDMASigInfoExt . . . 574
8.544.2.6 pTDSCDMASigInfoRscp . . . 574
8.544.2.7 pWCDMASSInfo . . . 574
8.545 nasGetSysInfoResp Struct Reference . . . 574
8.545.1 Detailed Description . . . 575
8.545.2 Field Documentation . . . 576
8.545.2.1 pAddCDMASysInfo . . . 576
8.545.2.2 pAddGSMSysInfo . . . 576

8.545.2.3 pAddHDRSysInfo . . . 576
8.545.2.4 pAddLTESysInfo . . . 576
8.545.2.5 pAddWCDMASysInfo . . . 576
8.545.2.6 pCDMASrvStatusInfo . . . 576
8.545.2.7 pCDMASysInfo . . . 576
8.545.2.8 pGSMCallBarringSysInfo . . . 576
8.545.2.9 pGSMCipherDomainSysInfo . . . 576
8.545.2.10 pGSMSrvStatusInfo . . . 576
8.545.2.11 pGSMSysInfo . . . 576
8.545.2.12 pHDRSrvStatusInfo . . . 576
8.545.2.13 pHDRSysInfo . . . 577
8.545.2.14 pLTESrvStatusInfo . . . 577
8.545.2.15 pLTESysInfo . . . 577
8.545.2.16 pLTEVoiceSupportSysInfo . . . 577
8.545.2.17 pWCDMACallBarringSysInfo . . . 577
8.545.2.18 pWCDMACipherDomainSysInfo . . . 577
8.545.2.19 pWCDMASrvStatusInfo . . . 577
8.545.2.20 pWCDMASysInfo . . . 577
8.546 nasGetTxRxInfoReq Struct Reference . . . 577
8.546.1 Detailed Description . . . 577
8.546.2 Field Documentation . . . 577
8.546.2.1 radio_if . . . 577
8.547 nasGetTxRxInfoResp Struct Reference . . . 577
8.547.1 Detailed Description . . . 578
8.547.2 Field Documentation . . . 578
8.547.2.1 pRXChain0Info . . . 578
8.547.2.2 pRXChain1Info . . . 578
8.547.2.3 pTXInfo . . . 578
8.548 NASGWAcqOrderPrefTlv Struct Reference . . . 578
8.548.1 Detailed Description . . . 578
8.548.2 Field Documentation . . . 578
8.548.2.1 GWAcqOrderPref . . . 578
8.548.2.2 TlvPresent . . . 578
8.549 nasIndicationRegisterReq Struct Reference . . . 578
8.549.1 Detailed Description . . . 579
8.549.2 Field Documentation . . . 581
8.549.2.1 pDDTMInd . . . 581
8.549.2.2 pDualStandByPrefInd . . . 581
8.549.2.3 pErrorRateInd . . . 581
8.549.2.4 pHDRNewUATIAssInd . . . 581

8.549.2.5 pHDRSessionCloseInd . . . 581
8.549.2.6 pLTECphyCa . . . 581
8.549.2.7 pManagedRoamingInd . . . 581
8.549.2.8 pNetworkRejectInd . . . 581
8.549.2.9 pNetworkTimeInd . . . 581
8.549.2.10 pServingSystemInd . . . 581
8.549.2.11 pSignalStrengthInd . . . 581
8.549.2.12 pSubscriptionInfoInd . . . 581
8.549.2.13 pSuppressSysInfoInd . . . 581
8.549.2.14 pSysInfoInd . . . 581
8.549.2.15 pSystemSelectionInd . . . 582
8.550 nasInitNetworkReg Struct Reference . . . 582
8.550.1 Detailed Description . . . 582
8.550.2 Field Documentation . . . 582
8.550.2.1 pChangeDuration . . . 582
8.550.2.2 pMncPcsDigitStatus . . . 582
8.550.2.3 pMNRInfo . . . 582
8.550.2.4 regAction . . . 582
8.551 NASLTEBandPreferenceTlv Struct Reference . . . 582
8.551.1 Detailed Description . . . 583
8.551.2 Field Documentation . . . 583
8.551.2.1 LTEBandPref . . . 583
8.551.2.2 TlvPresent . . . 583
8.552 NASLteNasReleaseInfoTlv Struct Reference . . . 583
8.552.1 Detailed Description . . . 583
8.552.2 Field Documentation . . . 583
8.552.2.1 nas_major . . . 583
8.552.2.2 nas_minor . . . 583
8.552.2.3 nas_release . . . 584
8.552.2.4 TlvPresent . . . 584
8.553 NASModePreferenceTlv Struct Reference . . . 584
8.553.1 Detailed Description . . . 584
8.553.2 Field Documentation . . . 584
8.553.2.1 ModePref . . . 584
8.553.2.2 TlvPresent . . . 584
8.554 NASNetSelPreferenceTlv Struct Reference . . . 584
8.554.1 Detailed Description . . . 584
8.554.2 Field Documentation . . . 584
8.554.2.1 NetSelPref . . . 584
8.554.2.2 TlvPresent . . . 584

8.555 nasNetworkReject Struct Reference . . . 584
8.555.1 Detailed Description . . . 585
8.555.2 Field Documentation . . . 587
8.555.2.1 pCsgId . . . 587
8.555.2.2 pPlmnId . . . 587
8.555.2.3 radioIf . . . 587
8.555.2.4 rejectCause . . . 587
8.555.2.5 serviceDomain . . . 587
8.556 nasNetworkTime Struct Reference . . . 587
8.556.1 Detailed Description . . . 587
8.556.2 Field Documentation . . . 588
8.556.2.1 pDayltSavAdj . . . 588
8.556.2.2 pRadioInterface . . . 588
8.556.2.3 pTimeZone . . . 588
8.556.2.4 universalTime . . . 588
8.557 nasOperatorNameResp Struct Reference . . . 588
8.557.1 Detailed Description . . . 588
8.557.2 Field Documentation . . . 589
8.557.2.1 pNITZInformation . . . 589
8.557.2.2 pOperatorNameString . . . 589
8.557.2.3 pOperatorPLMNList . . . 589
8.557.2.4 pPLMNNetworkName . . . 589
8.557.2.5 pSrvcProviderName . . . 589
8.558 NASOTAMessageTlv Struct Reference . . . 589
8.558.1 Detailed Description . . . 589
8.558.2 Field Documentation . . . 590
8.558.2.1 data_buf . . . 590
8.558.2.2 data_len . . . 590
8.558.2.3 message_type . . . 590
8.558.2.4 TlvPresent . . . 590
8.559 NASPhyCaAggPcellInfo Struct Reference . . . 590
8.559.1 Detailed Description . . . 590
8.559.2 Field Documentation . . . 590
8.559.2.1 dl_bw_value . . . 590
8.559.2.2 freq . . . 590
8.559.2.3 iLTEbandValue . . . 590
8.559.2.4 pci . . . 591
8.559.2.5 TlvPresent . . . 591
8.560 NASPhyCaAggScellDlBw Struct Reference . . . 591
8.560.1 Detailed Description . . . 591

8.560.2 Field Documentation . . . 591
8.560.2.1 dl_bw_value . . . 591
8.560.2.2 TlvPresent . . . 591
8.561 NASPhyCaAggScellIndex Struct Reference . . . 591
8.561.1 Detailed Description . . . 591
8.561.2 Field Documentation . . . 591
8.561.2.1 scell_idx . . . 592
8.561.2.2 TlvPresent . . . 592
8.562 NASPhyCaAggScellIndType Struct Reference . . . 592
8.562.1 Detailed Description . . . 592
8.562.2 Field Documentation . . . 592
8.562.2.1 freq . . . 592
8.562.2.2 pci . . . 592
8.562.2.3 scell_state . . . 592
8.562.2.4 TlvPresent . . . 592
8.563 NASPhyCaAggScellInfo Struct Reference . . . 592
8.563.1 Detailed Description . . . 593
8.563.2 Field Documentation . . . 593
8.563.2.1 dl_bw_value . . . 593
8.563.2.2 freq . . . 593
8.563.2.3 iLTEbandValue . . . 593
8.563.2.4 pci . . . 593
8.563.2.5 scell_state . . . 593
8.563.2.6 TlvPresent . . . 593
8.564 nasPLMNNameReq Struct Reference . . . 593
8.564.1 Detailed Description . . . 594
8.564.2 Field Documentation . . . 594
8.564.2.1 mcc . . . 594
8.564.2.2 mnc . . . 594
8.564.2.3 pMncPcsStatus . . . 594
8.565 nasPLMNNameResp Struct Reference . . . 594
8.565.1 Detailed Description . . . 595
8.565.2 Field Documentation . . . 596
8.565.2.1 longName . . . 596
8.565.2.2 longNameCI . . . 596
8.565.2.3 longNameEn . . . 596
8.565.2.4 longNameLen . . . 596
8.565.2.5 longNameSB . . . 596
8.565.2.6 shortName . . . 596
8.565.2.7 shortNameCI . . . 596

8.565.2.8 shortNameEn . . . 596
8.565.2.9 shortNameLen . . . 596
8.565.2.10 shortNameSB . . . 596
8.565.2.11 spn . . . 596
8.565.2.12 spnEncoding . . . 597
8.565.2.13 spnLength . . . 597
8.566 NASPRLPreferenceTlv Struct Reference . . . 597
8.566.1 Detailed Description . . . 597
8.566.2 Field Documentation . . . 597
8.566.2.1 PRLPref . . . 597
8.566.2.2 TlvPresent . . . 597
8.567 NASQmiCbkNasSwiOTAMessageInd Struct Reference . . . 597
8.567.1 Detailed Description . . . 597
8.567.2 Field Documentation . . . 597
8.567.2.1 nasRelInfoTlv . . . 598
8.567.2.2 otaMsgTlv . . . 598
8.567.2.3 timeTlv . . . 598
8.568 NASQmiCbkNasSystemSelPrefInd Struct Reference . . . 598
8.568.1 Detailed Description . . . 598
8.568.2 Field Documentation . . . 599
8.568.2.1 BPTlv . . . 599
8.568.2.2 EMTlv . . . 599
8.568.2.3 GWAOPTlv . . . 599
8.568.2.4 LBPTlv . . . 599
8.568.2.5 MPTlv . . . 599
8.568.2.6 NSPTlv . . . 599
8.568.2.7 PRLPTlv . . . 599
8.568.2.8 RPTlv . . . 599
8.568.2.9 SDPTlv . . . 599
8.569 NASRoamPreferenceTlv Struct Reference . . . 599
8.569.1 Detailed Description . . . 599
8.569.2 Field Documentation . . . 599
8.569.2.1 RoamPref . . . 599
8.569.2.2 TlvPresent . . . 599
8.570 NASServDomainPrefTlv Struct Reference . . . 599
8.570.1 Detailed Description . . . 600
8.570.2 Field Documentation . . . 600
8.570.2.1 SrvDomainPref . . . 600
8.570.2.2 TlvPresent . . . 600
8.571 NASServingSystemInfo Struct Reference . . . 600

8.571.1 Detailed Description . . . 600
8.571.2 Field Documentation . . . 601
8.571.2.1 csAttachState . . . 601
8.571.2.2 hdrPersonality . . . 601
8.571.2.3 psAttachState . . . 601
8.571.2.4 radioInterfaceList . . . 601
8.571.2.5 radioInterfaceNo . . . 601
8.571.2.6 registrationState . . . 601
8.571.2.7 selectedNetwork . . . 601
8.572 nasSeteDRXParamsReq Struct Reference . . . 601
8.572.1 Detailed Description . . . 601
8.572.2 Field Documentation . . . 602
8.572.2.1 pCycleLen . . . 602
8.572.2.2 pEdrxEnable . . . 602
8.572.2.3 pPagingTimeWindow . . . 602
8.573 nasSigInfo Struct Reference . . . 602
8.573.1 Detailed Description . . . 602
8.573.2 Field Documentation . . . 603
8.573.2.1 pCDMASigInfo . . . 603
8.573.2.2 pGSMSigInfo . . . 603
8.573.2.3 pHDRSigInfo . . . 603
8.573.2.4 pLTESigInfo . . . 603
8.573.2.5 pRscp . . . 603
8.573.2.6 pTDSCDMASigInfoExt . . . 603
8.573.2.7 pWCDMASigInfo . . . 603
8.574 nasSwiGetChannelLockResp Struct Reference . . . 603
8.574.1 Detailed Description . . . 603
8.574.2 Field Documentation . . . 604
8.574.2.1 pLteEARFCN . . . 604
8.574.2.2 pLtePCI . . . 604
8.574.2.3 pWcdmaUARFCN . . . 604
8.575 NasSwiIndReg Struct Reference . . . 604
8.575.1 Detailed Description . . . 604
8.575.2 Field Documentation . . . 605
8.575.2.1 gsmUmtsDl . . . 605
8.575.2.2 gsmUmtsUl . . . 605
8.575.2.3 lteEmmDl . . . 605
8.575.2.4 lteEmmUl . . . 605
8.575.2.5 lteEsmDl . . . 605
8.575.2.6 lteEsmUl . . . 605

8.575.2.7 pRankIndicatorInd . . . 605
8.575.2.8 pTimer . . . 605
8.576 nasSwiSetChannelLockReq Struct Reference . . . 605
8.576.1 Detailed Description . . . 605
8.576.2 Field Documentation . . . 605
8.576.2.1 pLteEARFCN . . . 605
8.576.2.2 pLtePCI . . . 606
8.576.2.3 pWcdmaUARFCN . . . 606
8.577 nasSysInfo Struct Reference . . . 606
8.577.1 Detailed Description . . . 606
8.577.2 Field Documentation . . . 608
8.577.2.1 pAddCDMASysInfo . . . 608
8.577.2.2 pAddGSMSysInfo . . . 608
8.577.2.3 pAddHDRSysInfo . . . 608
8.577.2.4 pAddLTESysInfo . . . 608
8.577.2.5 pAddWCDMASysInfo . . . 608
8.577.2.6 pCDMASrvStatusInfo . . . 608
8.577.2.7 pCDMASysInfo . . . 608
8.577.2.8 pGSMCallBarringSysInfo . . . 608
8.577.2.9 pGSMCipherDomainSysInfo . . . 608
8.577.2.10 pGSMSrvStatusInfo . . . 608
8.577.2.11 pGSMSysInfo . . . 608
8.577.2.12 pHDRSrvStatusInfo . . . 608
8.577.2.13 pHDRSysInfo . . . 608
8.577.2.14 pLTESrvStatusInfo . . . 608
8.577.2.15 pLTESysInfo . . . 608
8.577.2.16 pLTEVoiceSupportSysInfo . . . 608
8.577.2.17 pSysInfoNoChange . . . 608
8.577.2.18 pWCDMACallBarringSysInfo . . . 608
8.577.2.19 pWCDMACipherDomainSysInfo . . . 608
8.577.2.20 pWCDMASrvStatusInfo . . . 609
8.577.2.21 pWCDMASysInfo . . . 609
8.578 NASTimeInfoTlv Struct Reference . . . 609
8.578.1 Detailed Description . . . 609
8.578.2 Field Documentation . . . 609
8.578.2.1 time . . . 609
8.578.2.2 TlvPresent . . . 609
8.579 nasTimers Struct Reference . . . 609
8.579.1 Detailed Description . . . 609
8.579.2 Field Documentation . . . 610

8.579.2.1 t3396_apn . . . 610
8.579.2.2 t3396_plmn_id . . . 610
8.579.2.3 t3396_val . . . 610
8.580 netSelectionPref Struct Reference . . . 610
8.580.1 Detailed Description . . . 610
8.580.2 Field Documentation . . . 610
8.580.2.1 mcc . . . 610
8.580.2.2 mnc . . . 610
8.580.2.3 netReg . . . 610
8.581 NetStats Struct Reference . . . 610
8.581.1 Detailed Description . . . 611
8.581.2 Field Documentation . . . 611
8.581.2.1 rx_bytes . . . 611
8.581.2.2 rx_errors . . . 611
8.581.2.3 rx_overflows . . . 611
8.581.2.4 rx_packets . . . 611
8.581.2.5 tx_bytes . . . 611
8.581.2.6 tx_errors . . . 611
8.581.2.7 tx_overflows . . . 611
8.581.2.8 tx_packets . . . 611
8.582 NetworkDebugResp Struct Reference . . . 612
8.582.1 Detailed Description . . . 612
8.582.2 Field Documentation . . . 612
8.582.2.1 pDataStatusDetail . . . 612
8.582.2.2 pDeviceConfigDetail . . . 612
8.582.2.3 pNetworkStat1x . . . 612
8.582.2.4 pNetworkStatEVDO . . . 612
8.582.2.5 pObjectVer . . . 612
8.583 NetworkStat1x Struct Reference . . . 612
8.583.1 Detailed Description . . . 613
8.583.2 Field Documentation . . . 614
8.583.2.1 ActSetCnt . . . 614
8.583.2.2 NeighborSetCnt . . . 614
8.583.2.3 pActPilotPNElements . . . 614
8.583.2.4 pNeighborSetPilotPN . . . 614
8.583.2.5 RX_EC_IO . . . 614
8.583.2.6 RX_PWR . . . 614
8.583.2.7 SO . . . 614
8.583.2.8 State . . . 614
8.583.2.9 TX_PWR . . . 615

8.584 NetworkStatEVDO Struct Reference . . . 615
8.584.1 Detailed Description . . . 615
8.584.2 Field Documentation . . . 616
8.584.2.1 MACIndex . . . 616
8.584.2.2 PER . . . 616
8.584.2.3 PilotEnergy . . . 616
8.584.2.4 pSectorID . . . 616
8.584.2.5 RX_PWR . . . 616
8.584.2.6 SectorIDLen . . . 616
8.584.2.7 SNR . . . 616
8.584.2.8 State . . . 616
8.585 newMTMessageTlv Struct Reference . . . 616
8.585.1 Detailed Description . . . 616
8.585.2 Field Documentation . . . 617
8.585.2.1 MTMessageInfo . . . 617
8.585.2.2 TlvPresent . . . 617
8.586 newPwdData Struct Reference . . . 617
8.586.1 Detailed Description . . . 617
8.586.2 Field Documentation . . . 617
8.586.2.1 newPwd . . . 617
8.586.2.2 newPwdAgain . . . 617
8.587 nmrCellInfo Struct Reference . . . 617
8.587.1 Detailed Description . . . 618
8.587.2 Field Documentation . . . 618
8.587.2.1 nmrArfcn . . . 618
8.587.2.2 nmrBsic . . . 618
8.587.2.3 nmrCellID . . . 618
8.587.2.4 nmrLac . . . 618
8.587.2.5 nmrPlmn . . . 618
8.587.2.6 nmrRxLev . . . 618
8.588 nr5gBandPref Struct Reference . . . 619
8.588.1 Detailed Description . . . 619
8.588.2 Field Documentation . . . 619
8.588.2.1 bits_129_192 . . . 619
8.588.2.2 bits_193_256 . . . 619
8.588.2.3 bits_1_64 . . . 619
8.588.2.4 bits_65_128 . . . 619
8.589 NSSAudioCtrl Struct Reference . . . 619
8.589.1 Detailed Description . . . 619
8.589.2 Field Documentation . . . 620

8.589.2.1 downLink . . . 620
8.589.2.2 upLink . . . 620
8.590 NWProfile Struct Reference . . . 620
8.590.1 Detailed Description . . . 620
8.590.2 Field Documentation . . . 620
8.590.2.1 pProfSz . . . 620
8.590.2.2 pProfValues . . . 620
8.590.2.3 tech . . . 620
8.591 omaDmConfigTlv Struct Reference . . . 620
8.591.1 Detailed Description . . . 620
8.591.2 Field Documentation . . . 621
8.591.2.1 alertmsg . . . 621
8.591.2.2 alertmsglength . . . 621
8.591.2.3 state . . . 621
8.591.2.4 userInputReq . . . 621
8.591.2.5 userInputTimeout . . . 621
8.592 omaDmConfigTlvExt Struct Reference . . . 621
8.592.1 Detailed Description . . . 621
8.592.2 Field Documentation . . . 623
8.592.2.1 alertmsg . . . 623
8.592.2.2 alertmsglength . . . 623
8.592.2.3 state . . . 623
8.592.2.4 userInputReq . . . 623
8.592.2.5 userInputTimeout . . . 623
8.593 omaDmFotaTlv Struct Reference . . . 623
8.593.1 Detailed Description . . . 623
8.593.2 Field Documentation . . . 624
8.593.2.1 description . . . 624
8.593.2.2 descriptionlength . . . 624
8.593.2.3 fwdloadsize . . . 625
8.593.2.4 fwloadComplete . . . 625
8.593.2.5 namelength . . . 625
8.593.2.6 package_name . . . 625
8.593.2.7 sessionType . . . 625
8.593.2.8 severity . . . 625
8.593.2.9 state . . . 625
8.593.2.10 updateCompleteStatus . . . 625
8.593.2.11 userInputReq . . . 625
8.593.2.12 userInputTimeout . . . 625
8.593.2.13 version . . . 625

8.593.2.14 versionlength . . . 625
8.594 omaDmFotaTlvExt Struct Reference . . . 625
8.594.1 Detailed Description . . . 625
8.594.2 Field Documentation . . . 627
8.594.2.1 description . . . 627
8.594.2.2 descriptionlength . . . 627
8.594.2.3 fumoResultCode . . . 627
8.594.2.4 namelength . . . 627
8.594.2.5 package_name . . . 627
8.594.2.6 packageSize . . . 627
8.594.2.7 receivedBytes . . . 627
8.594.2.8 reserved . . . 627
8.594.2.9 state . . . 627
8.594.2.10 userInputTimeout . . . 627
8.594.2.11 version . . . 627
8.594.2.12 versionlength . . . 627
8.595 omaDmNotificationsTlv Struct Reference . . . 627
8.595.1 Detailed Description . . . 627
8.595.2 Field Documentation . . . 628
8.595.2.1 notification . . . 628
8.595.2.2 sessionStatus . . . 628
8.596 operatorNameString Struct Reference . . . 628
8.596.1 Detailed Description . . . 628
8.596.2 Field Documentation . . . 628
8.596.2.1 PLMNName . . . 629
8.597 OperatorPLMNData Struct Reference . . . 629
8.597.1 Detailed Description . . . 629
8.597.2 Field Documentation . . . 629
8.597.2.1 lac1 . . . 629
8.597.2.2 lac2 . . . 629
8.597.2.3 mcc . . . 629
8.597.2.4 mnc . . . 629
8.597.2.5 PLMNRecID . . . 629
8.598 operatorPLMNList Struct Reference . . . 629
8.598.1 Detailed Description . . . 630
8.598.2 Field Documentation . . . 630
8.598.2.1 numInstance . . . 630
8.598.2.2 PLMNData . . . 630
8.599 pack_dms_ActivateAutomatic_t Struct Reference . . . 630
8.599.1 Detailed Description . . . 630

8.599.2 Field Documentation . . . 630
8.599.2.1 actCode . . . 630
8.600 pack_dms_GetCustFeaturesV2_t Struct Reference . . . 630
8.600.1 Detailed Description . . . 631
8.600.2 Field Documentation . . . 631
8.600.2.1 cust_id . . . 631
8.600.2.2 list_type . . . 631
8.600.2.3 Tlvresult . . . 631
8.601 pack_dms_ResetToFactoryDefaults_t Struct Reference . . . 631
8.601.1 Detailed Description . . . 631
8.601.2 Field Documentation . . . 631
8.601.2.1 spc . . . 631
8.602 pack_dms_SetActivationStatusCallback_t Struct Reference . . . 631
8.602.1 Detailed Description . . . 632
8.602.2 Field Documentation . . . 632
8.602.2.1 activationState . . . 632
8.603 pack_dms_SetCrashAction_t Struct Reference . . . 632
8.603.1 Detailed Description . . . 632
8.603.2 Field Documentation . . . 632
8.603.2.1 crashAction . . . 632
8.604 pack_dms_SetCustFeature_t Struct Reference . . . 632
8.604.1 Detailed Description . . . 633
8.604.2 Field Documentation . . . 634
8.604.2.1 DHCPRelayEnabled . . . 634
8.604.2.2 DisableIMSI . . . 634
8.604.2.3 GpsEnable . . . 634
8.604.2.4 GPSLPM . . . 634
8.604.2.5 GPSSel . . . 634
8.604.2.6 IPFamSupport . . . 634
8.604.2.7 IsVoiceEnabled . . . 634
8.604.2.8 RMAutoConnect . . . 634
8.604.2.9 SMSSupport . . . 634
8.605 pack_dms_SetCustFeaturesV2_t Struct Reference . . . 634
8.605.1 Detailed Description . . . 635
8.605.2 Field Documentation . . . 635
8.605.2.1 cust_id . . . 635
8.605.2.2 cust_value . . . 635
8.605.2.3 Tlvresult . . . 635
8.605.2.4 value_length . . . 635
8.606 pack_dms_SetEventReport_t Struct Reference . . . 635

8.606.1 Detailed Description . . . 635
8.606.2 Field Documentation . . . 635
8.606.2.1 mode . . . 635
8.607pack_dms_SetIndicationRegister_t Struct Reference . . . 636
8.607.1 Detailed Description . . . 636
8.607.2 Field Documentation . . . 636
8.607.2.1 PSMCfgChangeInfo . . . 636
8.607.2.2 PSMStatus . . . 636
8.607.2.3 RptIMSCapability . . . 636
8.608pack_dms_SetPower_t Struct Reference . . . 636
8.608.1 Detailed Description . . . 637
8.608.2 Field Documentation . . . 637
8.608.2.1 mode . . . 637
8.608.2.2 Tlvresult . . . 637
8.609pack_dms_SetUSBComp_t Struct Reference . . . 637
8.609.1 Detailed Description . . . 637
8.609.2 Field Documentation . . . 638
8.609.2.1 Tlvresult . . . 638
8.609.2.2 USBComp . . . 638
8.610pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 638
8.610.1 Detailed Description . . . 638
8.610.2 Field Documentation . . . 639
8.610.2.1 resetInfoInd . . . 639
8.611pack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference . . . 639
8.611.1 Detailed Description . . . 639
8.611.2 Field Documentation . . . 639
8.611.2.1 pActiveTimer . . . 639
8.611.2.2 pPeriodicUpdateTimer . . . 639
8.611.2.3 pPsmEnableState . . . 639
8.612pack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 639
8.612.1 Detailed Description . . . 640
8.612.2 Field Documentation . . . 640
8.612.2.1 clear . . . 640
8.613pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 640
8.613.1 Detailed Description . . . 640
8.613.2 Field Documentation . . . 640
8.613.2.1 pDestSMSContent . . . 640
8.613.2.2 pDestSMSNum . . . 640
8.614pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 640
8.614.1 Detailed Description . . . 641

8.614.2 Field Documentation . . . 641
8.614.2.1 manString . . . 641
8.614.2.2 modelString . . . 641
8.614.2.3 plasmaIDString . . . 641
8.614.2.4 swVerString . . . 641
8.615 pack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 641
8.615.1 Detailed Description . . . 641
8.615.2 Field Documentation . . . 642
8.615.2.1 nameString . . . 642
8.615.2.2 versionString . . . 642
8.616 pack_dms_SwiSetEventReport_t Struct Reference . . . 642
8.616.1 Detailed Description . . . 642
8.616.2 Field Documentation . . . 642
8.616.2.1 pTempReport . . . 642
8.616.2.2 pVoltReport . . . 642
8.617 pack_dms_UIMChangePIN_t Struct Reference . . . 642
8.617.1 Detailed Description . . . 642
8.617.2 Field Documentation . . . 643
8.617.2.1 id . . . 643
8.617.2.2 newValue . . . 643
8.617.2.3 oldValue . . . 643
8.618 pack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 643
8.618.1 Detailed Description . . . 643
8.618.2 Field Documentation . . . 643
8.618.2.1 facility . . . 643
8.619 pack_dms_UIMGetICCID_t Struct Reference . . . 643
8.619.1 Detailed Description . . . 644
8.619.2 Field Documentation . . . 644
8.619.2.1 Tlvresult . . . 644
8.620 pack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 644
8.620.1 Detailed Description . . . 644
8.620.2 Field Documentation . . . 644
8.620.2.1 facility . . . 644
8.620.2.2 facilityCk . . . 644
8.620.2.3 facilityState . . . 645
8.621 pack_dms_UIMSetPINProtection_t Struct Reference . . . 645
8.621.1 Detailed Description . . . 645
8.621.2 Field Documentation . . . 645
8.621.2.1 bEnable . . . 645
8.621.2.2 id . . . 645

8.621.2.3 value . . . 645
8.622 pack_dms_UIMUnblockControlKey_t Struct Reference . . . 645
8.622.1 Detailed Description . . . 645
8.622.2 Field Documentation . . . 646
8.622.2.1 facility . . . 646
8.622.2.2 facilityCk . . . 646
8.623 pack_dms_UIMUnblockPIN_t Struct Reference . . . 646
8.623.1 Detailed Description . . . 646
8.623.2 Field Documentation . . . 646
8.623.2.1 id . . . 646
8.623.2.2 newPin . . . 646
8.623.2.3 pukValue . . . 646
8.624 pack_dms_UIMVerifyPIN_t Struct Reference . . . 647
8.624.1 Detailed Description . . . 647
8.624.2 Field Documentation . . . 647
8.624.2.1 id . . . 647
8.624.2.2 value . . . 647
8.625 pack_fms_GetImagesPreference_t Struct Reference . . . 647
8.625.1 Detailed Description . . . 647
8.625.2 Field Documentation . . . 647
8.625.2.1 TlvResult . . . 647
8.626 pack_fms_GetStoredImages_t Struct Reference . . . 647
8.626.1 Detailed Description . . . 648
8.626.2 Field Documentation . . . 648
8.626.2.1 TlvResult . . . 648
8.627 pack_fms_SetImagesPreference_t Struct Reference . . . 648
8.627.1 Detailed Description . . . 648
8.627.2 Field Documentation . . . 649
8.627.2.1 bForceDownload . . . 649
8.627.2.2 imageListSize . . . 649
8.627.2.3 modemIndex . . . 649
8.627.2.4 pImageList . . . 649
8.627.2.5 TlvResult . . . 649
8.628 pack_loc_Delete_Assist_Data_t Struct Reference . . . 649
8.628.1 Detailed Description . . . 649
8.628.2 Field Documentation . . . 649
8.628.2.1 pBdsSVInfo . . . 649
8.628.2.2 pCellDb . . . 649
8.628.2.3 pClkInfo . . . 650
8.628.2.4 pGnssData . . . 650

8.628.2.5 pSVInfo . . . 650
8.628.2.6 Tlvresult . . . 650
8.629 pack_loc_EventRegister_t Struct Reference . . . 650
8.629.1 Detailed Description . . . 650
8.629.2 Field Documentation . . . 652
8.629.2.1 eventRegister . . . 652
8.629.2.2 Tlvresult . . . 652
8.630 pack_loc_SetExtPowerState_t Struct Reference . . . 652
8.630.1 Detailed Description . . . 652
8.630.2 Field Documentation . . . 652
8.630.2.1 extPowerState . . . 652
8.630.2.2 Tlvresult . . . 652
8.631 pack_loc_SetOperationMode_t Struct Reference . . . 652
8.631.1 Detailed Description . . . 653
8.631.2 Field Documentation . . . 653
8.631.2.1 mode . . . 653
8.631.2.2 Tlvresult . . . 653
8.632 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 653
8.632.1 Detailed Description . . . 653
8.632.2 Field Documentation . . . 653
8.632.2.1 Tlvresult . . . 653
8.632.2.2 xid . . . 654
8.633 pack_loc_SLQSLOCGetServer_t Struct Reference . . . 654
8.633.1 Detailed Description . . . 654
8.633.2 Field Documentation . . . 654
8.633.2.1 has_serverAddrTypeMask . . . 654
8.633.2.2 serverAddrTypeMask . . . 654
8.633.2.3 serverType . . . 654
8.634 pack_loc_SLQSLOCInjectPosition_t Struct Reference . . . 654
8.634.1 Detailed Description . . . 655
8.634.2 Field Documentation . . . 658
8.634.2.1 altitudeSrcInfo . . . 658
8.634.2.2 altitudeWrtEllipsoid . . . 658
8.634.2.3 altitudeWrtMeanSeaLevel . . . 658
8.634.2.4 has_altitudeSrcInfo . . . 658
8.634.2.5 has_altitudeWrtEllipsoid . . . 658
8.634.2.6 has_altitudeWrtMeanSeaLevel . . . 658
8.634.2.7 has_horConfidence . . . 658
8.634.2.8 has_horReliability . . . 658
8.634.2.9 has_horUncCircular . . . 658

8.634.2.10 has_latitude . . . 658
8.634.2.11 has_longitude . . . 658
8.634.2.12 has_positionSrc . . . 658
8.634.2.13 has_rawHorConfidence . . . 658
8.634.2.14 has_rawHorUncCircular . . . 658
8.634.2.15 has_timestampAge . . . 658
8.634.2.16 has_timestampUtc . . . 658
8.634.2.17 has_vertConfidence . . . 658
8.634.2.18 has_vertRelicability . . . 659
8.634.2.19 has_vertUnc . . . 659
8.634.2.20 horConfidence . . . 659
8.634.2.21 horReliability . . . 659
8.634.2.22 horUncCircular . . . 659
8.634.2.23 latitude . . . 659
8.634.2.24 longitude . . . 659
8.634.2.25 positionSrc . . . 659
8.634.2.26 rawHorConfidence . . . 659
8.634.2.27 rawHorUncCircular . . . 659
8.634.2.28 timestampAge . . . 659
8.634.2.29 timestampUtc . . . 659
8.634.2.30 vertConfidence . . . 659
8.634.2.31 vertReliability . . . 659
8.634.2.32 vertUnc . . . 659
8.635 pack_loc_SLQSLOCInjectSensorData_t Struct Reference . . . 659
8.635.1 Detailed Description . . . 659
8.635.2 Field Documentation . . . 660
8.635.2.1 acceleroData . . . 660
8.635.2.2 acceleroTimeSrc . . . 660
8.635.2.3 accelTemp . . . 660
8.635.2.4 gyroData . . . 660
8.635.2.5 gyroTemp . . . 661
8.635.2.6 gyroTimeSrc . . . 661
8.635.2.7 has_acceleroTimeSrc . . . 661
8.635.2.8 has_accelTemp . . . 661
8.635.2.9 has_accleroData . . . 661
8.635.2.10 has_gyroData . . . 661
8.635.2.11 has_gyroTemp . . . 661
8.635.2.12 has_gyroTimeSrc . . . 661
8.635.2.13 has_opaqueId . . . 661
8.635.2.14 opaqueId . . . 661

8.636 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference . . . 661
8.636.1 Detailed Description . . . 661
8.636.2 Field Documentation . . . 661
8.636.2.1 timeMsec . . . 661
8.636.2.2 timeUncMsec . . . 661
8.637 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference . . . 661
8.637.1 Detailed Description . . . 662
8.637.2 Field Documentation . . . 662
8.637.2.1 confidence . . . 662
8.637.2.2 has_confidence . . . 662
8.637.2.3 state . . . 662
8.638 pack_loc_SLQSLOCSetServer_t Struct Reference . . . 662
8.638.1 Detailed Description . . . 662
8.638.2 Field Documentation . . . 663
8.638.2.1 pIPv4Config . . . 663
8.638.2.2 pIPv6Config . . . 663
8.638.2.3 pURLAddr . . . 663
8.638.2.4 serverType . . . 663
8.639 pack_loc_Start_t Struct Reference . . . 663
8.639.1 Detailed Description . . . 663
8.639.2 Field Documentation . . . 664
8.639.2.1 pApplicationInfo . . . 664
8.639.2.2 pConfigAltitudeAssumed . . . 664
8.639.2.3 pHorizontalAccuracyLvl . . . 664
8.639.2.4 pIntermediateReportState . . . 664
8.639.2.5 pMinIntervalTime . . . 664
8.639.2.6 pRecurrenceType . . . 665
8.639.2.7 SessionId . . . 665
8.639.2.8 Tlvresult . . . 665
8.640 pack_loc_Stop_t Struct Reference . . . 665
8.640.1 Detailed Description . . . 665
8.640.2 Field Documentation . . . 665
8.640.2.1 SessionId . . . 665
8.640.2.2 Tlvresult . . . 665
8.641 pack_nas_InitiateDomainAttach_t Struct Reference . . . 665
8.641.1 Detailed Description . . . 665
8.641.2 Field Documentation . . . 665
8.641.2.1 action . . . 666
8.642 pack_nas_SetACCOLC_t Struct Reference . . . 666
8.642.1 Detailed Description . . . 666

8.642.2 Field Documentation . . . 666
8.642.2.1 accolc . . . 666
8.642.2.2 spc . . . 666
8.643pack_nas_SetCDMANetworkParameters_t Struct Reference . . . 666
8.643.1 Detailed Description . . . 666
8.643.2 Field Documentation . . . 667
8.643.2.1 pApplication . . . 667
8.643.2.2 pBroadcast . . . 667
8.643.2.3 pCustomSCP . . . 667
8.643.2.4 pForceRev0 . . . 667
8.643.2.5 pProtocol . . . 667
8.643.2.6 pRoaming . . . 667
8.643.2.7 pSPC . . . 668
8.644pack_nas_SetNetworkPreference_t Struct Reference . . . 668
8.644.1 Detailed Description . . . 668
8.644.2 Field Documentation . . . 668
8.644.2.1 Duration . . . 668
8.644.2.2 TechnologyPref . . . 668
8.644.2.3 Tlvresult . . . 668
8.645pack_nas_SLQSConfigSigInfo_t Struct Reference . . . 668
8.645.1 Detailed Description . . . 669
8.645.2 Field Documentation . . . 670
8.645.2.1 pECIOThresh . . . 670
8.645.2.2 pHDRSINRThresh . . . 670
8.645.2.3 pIOThresh . . . 670
8.645.2.4 pLTESigRptCfg . . . 670
8.645.2.5 pLTESNRThresh . . . 670
8.645.2.6 pRSRPThresh . . . 670
8.645.2.7 pRSRQThresh . . . 670
8.645.2.8 pRSSIThresh . . . 670
8.645.2.9 pTDSCDMASINRCONFThresh . . . 670
8.646pack_nas_SLQSGetPLMNName_t Struct Reference . . . 670
8.646.1 Detailed Description . . . 670
8.646.2 Field Documentation . . . 671
8.646.2.1 mcc . . . 671
8.646.2.2 mnc . . . 671
8.646.2.3 pMncPcsStatus . . . 671
8.647pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference . . . 671
8.647.1 Detailed Description . . . 671
8.647.2 Field Documentation . . . 671

8.647.2.1 pChangeDuration . . . 671
8.647.2.2 pMncPcsDigitStatus . . . 671
8.647.2.3 pMNRInfo . . . 672
8.647.2.4 regAction . . . 672
8.648 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . 672
8.648.1 Detailed Description . . . 672
8.648.2 Field Documentation . . . 675
8.648.2.1 pCDMAECIODelta . . . 675
8.648.2.2 pCDMAECIOThresh . . . 675
8.648.2.3 pCDMARSSIDelta . . . 675
8.648.2.4 pCDMARSSIThresh . . . 675
8.648.2.5 pGSMRSSIDelta . . . 675
8.648.2.6 pGSMRSSIThresh . . . 675
8.648.2.7 pHDRECIODelta . . . 675
8.648.2.8 pHDRECIOThresh . . . 675
8.648.2.9 pHDRIODelta . . . 675
8.648.2.10 pHDRIOThresh . . . 675
8.648.2.11 pHDRRSSIDelta . . . 675
8.648.2.12 pHDRRSSIThresh . . . 675
8.648.2.13 pHDRSINRDelta . . . 675
8.648.2.14 pHDRSINRThresh . . . 675
8.648.2.15 pLTERSRPDelta . . . 676
8.648.2.16 pLTERSRPThresh . . . 676
8.648.2.17 pLTERSRQDelta . . . 676
8.648.2.18 pLTERSRQThresh . . . 676
8.648.2.19 pLTERSSIDelta . . . 676
8.648.2.20 pLTERSSIThresh . . . 676
8.648.2.21 pLTESigRptConfig . . . 676
8.648.2.22 pLTESNRDelta . . . 676
8.648.2.23 pLTESNRThresh . . . 676
8.648.2.24 pTDSCDMAECIODelta . . . 676
8.648.2.25 pTDSCDMAECIOThresh . . . 676
8.648.2.26 pTDSCDMARSCPDelta . . . 676
8.648.2.27 pTDSCDMARSCPThresh . . . 676
8.648.2.28 pTDSCDMARSSIDelta . . . 676
8.648.2.29 pTDSCDMARSSIThresh . . . 676
8.648.2.30 pTDSCDMASINRDelta . . . 676
8.648.2.31 pTDSCDMASINRThresh . . . 676
8.648.2.32 pWCDMAECIODelta . . . 676
8.648.2.33 pWCDMAECIOThresh . . . 676

8.648.2.34 pWCDMARSSIDelta . . . 676
8.648.2.35 pWCDMARSSIThresh . . . 676
8.649 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 676
8.649.1 Detailed Description . . . 676
8.649.2 Field Documentation . . . 677
8.649.2.1 namID . . . 677
8.650 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 677
8.650.1 Detailed Description . . . 677
8.650.2 Field Documentation . . . 677
8.650.2.1 radio_if . . . 677
8.651 pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference . . . 677
8.651.1 Detailed Description . . . 678
8.651.2 Field Documentation . . . 679
8.651.2.1 pDDTMInd . . . 679
8.651.2.2 pDualStandByPrefInd . . . 680
8.651.2.3 pErrorRateInd . . . 680
8.651.2.4 pHDRNewUATIAssInd . . . 680
8.651.2.5 pHDRSessionCloseInd . . . 680
8.651.2.6 pLTECphyCa . . . 680
8.651.2.7 pManagedRoamingInd . . . 680
8.651.2.8 pNetworkRejectInd . . . 680
8.651.2.9 pNetworkTimeInd . . . 680
8.651.2.10 pServingSystemInd . . . 680
8.651.2.11 pSignalStrengthInd . . . 680
8.651.2.12 pSubscriptionInfoInd . . . 680
8.651.2.13 pSuppressSysInfoInd . . . 680
8.651.2.14 pSysInfoInd . . . 680
8.651.2.15 pSystemSelectionInd . . . 680
8.652 pack_nas_SLQSNASSeteDRXParams_t Struct Reference . . . 680
8.652.1 Detailed Description . . . 680
8.652.2 Field Documentation . . . 681
8.652.2.1 pCycleLen . . . 681
8.652.2.2 pEdrxEnable . . . 681
8.652.2.3 pPagingTimeWindow . . . 681
8.653 pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference . . . 681
8.653.1 Detailed Description . . . 681
8.653.2 Field Documentation . . . 682
8.653.2.1 gsmUmtsDl . . . 682
8.653.2.2 gsmUmtsUl . . . 682
8.653.2.3 lteEmmDl . . . 682

8.653.2.4 lteEmmUl . . . 682
8.653.2.5 lteEsmDl . . . 682
8.653.2.6 lteEsmUl . . . 682
8.653.2.7 pRankIndicatorInd . . . 682
8.653.2.8 pTimer . . . 682
8.654 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference . . . 682
8.654.1 Detailed Description . . . 682
8.654.2 Field Documentation . . . 683
8.654.2.1 pLteEARFCN . . . 683
8.654.2.2 pLtePCI . . . 683
8.654.2.3 pWcdmaUARFCN . . . 683
8.655 pack_nas_SLQSSetBandPreference_t Struct Reference . . . 683
8.655.1 Detailed Description . . . 683
8.655.2 Field Documentation . . . 684
8.655.2.1 bandPref . . . 685
8.656 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference . . . 685
8.656.1 Detailed Description . . . 685
8.656.2 Field Documentation . . . 685
8.656.2.1 bEnable . . . 685
8.656.2.2 pSigIndReq . . . 685
8.657 pack_nas_SLQSSetSysSelectionPref_t Struct Reference . . . 685
8.657.1 Detailed Description . . . 685
8.657.2 Field Documentation . . . 690
8.657.2.1 pAcqOrderPref . . . 690
8.657.2.2 pBandPref . . . 690
8.657.2.3 pChgDuration . . . 690
8.657.2.4 pCSGID . . . 690
8.657.2.5 pEmerMode . . . 690
8.657.2.6 pGWAcqOrderPref . . . 690
8.657.2.7 pLTEBandPref . . . 690
8.657.2.8 pMNCIncPCSDigStat . . . 690
8.657.2.9 pModePref . . . 690
8.657.2.10 pNetSelPref . . . 690
8.657.2.11 pPRLPref . . . 690
8.657.2.12 pRAT . . . 690
8.657.2.13 pRoamPref . . . 690
8.657.2.14 pSrvDomainPref . . . 690
8.657.2.15 pSrvRegRestriction . . . 691
8.657.2.16 pTdscdmaBandPref . . . 691
8.658 pack_nas_SLQSSetSysSelectionPrefExt_t Struct Reference . . . 691

8.658.1 Detailed Description . . . 691
8.658.2 Field Documentation . . . 699
8.658.2.1 pAcqOrderPref . . . 699
8.658.2.2 pBandPref . . . 699
8.658.2.3 pChgDuration . . . 699
8.658.2.4 pCiotAcqOrderPref . . . 699
8.658.2.5 pCiotLteOpMode . . . 699
8.658.2.6 pCiotLteOpModePref . . . 699
8.658.2.7 pCSGID . . . 699
8.658.2.8 pEmerMode . . . 699
8.658.2.9 pGWAcqOrderPref . . . 699
8.658.2.10 pLTEBandPref . . . 699
8.658.2.11 pLteM1BandPref . . . 699
8.658.2.12 pLteNb1BandPref . . . 699
8.658.2.13 pMNCIncPCSDigStat . . . 699
8.658.2.14 pModePref . . . 699
8.658.2.15 pNetSelPref . . . 699
8.658.2.16 pNr5gBandPref . . . 699
8.658.2.17 pPRLPref . . . 699
8.658.2.18 pRAT . . . 700
8.658.2.19 pRoamPref . . . 700
8.658.2.20 pSrvDomainPref . . . 700
8.658.2.21 pSrvRegRestriction . . . 700
8.658.2.22 pTdscdmaBandPref . . . 700
8.659 pack_nas_SLQSSwiPSDetach_t Struct Reference . . . 700
8.659.1 Detailed Description . . . 700
8.659.2 Field Documentation . . . 700
8.659.2.1 pDetachAction . . . 700
8.660 pack_pds_PDSInjectTimeReference_t Struct Reference . . . 700
8.660.1 Detailed Description . . . 700
8.660.2 Field Documentation . . . 700
8.660.2.1 systemDiscontinuities . . . 701
8.660.2.2 systemTime . . . 701
8.661 pack_pds_ResetPDSData_t Struct Reference . . . 701
8.661.1 Detailed Description . . . 701
8.661.2 Field Documentation . . . 701
8.661.2.1 pCellDataMask . . . 701
8.661.2.2 pGPSDataMask . . . 702
8.662 pack_pds_SetEventReportCallback_t Struct Reference . . . 702
8.662.1 Detailed Description . . . 702

8.662.2 Field Documentation . . . 702
8.662.2.1 posDataNmea . . . 702
8.662.2.2 rptPosData . . . 702
8.663pack_pds_SetPDSDefaults_t Struct Reference . . . 702
8.663.1 Detailed Description . . . 702
8.663.2 Field Documentation . . . 703
8.663.2.1 accuracy . . . 703
8.663.2.2 interval . . . 703
8.663.2.3 operation . . . 703
8.663.2.4 timeout . . . 703
8.664pack_pds_SetPDSState_t Struct Reference . . . 703
8.664.1 Detailed Description . . . 703
8.664.2 Field Documentation . . . 703
8.664.2.1 enable . . . 703
8.665pack_pds_SetPortAutomaticTracking_t Struct Reference . . . 703
8.665.1 Detailed Description . . . 704
8.665.2 Field Documentation . . . 704
8.665.2.1 bAuto . . . 704
8.666pack_pds_SetServiceAutomaticTracking_t Struct Reference . . . 704
8.666.1 Detailed Description . . . 704
8.666.2 Field Documentation . . . 704
8.666.2.1 bAuto . . . 704
8.667pack_pds_SetXTRAAutomaticDownload_t Struct Reference . . . 704
8.667.1 Detailed Description . . . 705
8.667.2 Field Documentation . . . 705
8.667.2.1 bEnabled . . . 705
8.667.2.2 interval . . . 705
8.668pack_pds_SetXTRANetwork_t Struct Reference . . . 705
8.668.1 Detailed Description . . . 705
8.668.2 Field Documentation . . . 705
8.668.2.1 preference . . . 705
8.669pack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 705
8.669.1 Detailed Description . . . 706
8.669.2 Field Documentation . . . 706
8.669.2.1 pNetworkMode . . . 706
8.670pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference . . . 706
8.670.1 Detailed Description . . . 706
8.670.2 Field Documentation . . . 706
8.670.2.1 forceFlag . . . 706
8.670.2.2 timeBase . . . 706

8.670.2.3 timeMsec . . . 707
8.670.2.4 timeUncMsec . . . 707
8.671 pack_pds_SLQSPDSInjectPositionData_t Struct Reference . . . 707
8.671.1 Detailed Description . . . 707
8.671.2 Field Documentation . . . 708
8.671.2.1 pAltitudeWrtEllipsoid . . . 708
8.671.2.2 pAltitudeWrtSealevel . . . 708
8.671.2.3 pHorizontalConfidence . . . 708
8.671.2.4 pHorizontalUncCircular . . . 708
8.671.2.5 pLatitude . . . 708
8.671.2.6 pLongitude . . . 708
8.671.2.7 pPositionSource . . . 708
8.671.2.8 pTimeStamp . . . 708
8.671.2.9 pTimeType . . . 708
8.671.2.10 pVerticalConfidence . . . 708
8.671.2.11 pVerticalUnc . . . 709
8.672 pack_pds_SLQSSetAGPSConfig_t Struct Reference . . . 709
8.672.1 Detailed Description . . . 709
8.672.2 Field Documentation . . . 709
8.672.2.1 pNetworkMode . . . 709
8.672.2.2 pServerAddress . . . 709
8.672.2.3 pServerPort . . . 709
8.672.2.4 pServerURL . . . 709
8.672.2.5 pServerURLLength . . . 709
8.673 pack_pds_SLQSSetPositionMethodState_t Struct Reference . . . 709
8.673.1 Detailed Description . . . 710
8.673.2 Field Documentation . . . 710
8.673.2.1 pWifiState . . . 710
8.673.2.2 pXtraDataState . . . 710
8.673.2.3 pXtraTimeState . . . 710
8.674 pack_pds_StartPDSTrackingSessionExt_t Struct Reference . . . 710
8.674.1 Detailed Description . . . 710
8.674.2 Field Documentation . . . 711
8.674.2.1 fixAccuracy . . . 711
8.674.2.2 fixCount . . . 711
8.674.2.3 fixInterval . . . 711
8.674.2.4 fixTimeout . . . 711
8.674.2.5 sessionControl . . . 711
8.674.2.6 sessionOperation . . . 711
8.674.2.7 sessionServerOption . . . 711

8.674.2.8 sessionType . . . 711
8.675 pack_qmi_t Struct Reference . . . 711
8.675.1 Detailed Description . . . 712
8.675.2 Field Documentation . . . 712
8.675.2.1 msgid . . . 712
8.675.2.2 svc . . . 712
8.675.2.3 timeout . . . 712
8.675.2.4 xid . . . 712
8.676 pack_qos_BindDataPort_t Struct Reference . . . 712
8.676.1 Detailed Description . . . 712
8.676.2 Field Documentation . . . 713
8.676.2.1 pMuxID . . . 713
8.676.2.2 pPeripheralEndPointID . . . 713
8.676.2.3 pSIODataPort . . . 713
8.677 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 713
8.677.1 Detailed Description . . . 713
8.677.2 Field Documentation . . . 713
8.677.2.1 apnId . . . 713
8.678 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 713
8.678.1 Detailed Description . . . 713
8.678.2 Field Documentation . . . 713
8.678.2.1 apnId . . . 713
8.679 pack_qos_SLQSSetQosEventCallback_t Struct Reference . . . 713
8.679.1 Detailed Description . . . 714
8.679.2 Field Documentation . . . 714
8.679.2.1 enable . . . 714
8.680 pack_rms_SetSMSWake_t Struct Reference . . . 714
8.680.1 Detailed Description . . . 714
8.680.2 Field Documentation . . . 714
8.680.2.1 enabled . . . 714
8.680.2.2 wake_mask . . . 714
8.681 pack_sar_SLQSSetRfSarState_t Struct Reference . . . 714
8.681.1 Detailed Description . . . 715
8.681.2 Field Documentation . . . 715
8.681.2.1 RfSarState . . . 715
8.682 pack_sms_SaveSMS_t Struct Reference . . . 715
8.682.1 Detailed Description . . . 715
8.682.2 Field Documentation . . . 716
8.682.2.1 messageFormat . . . 716
8.682.2.2 messageSize . . . 716

8.682.2.3 pMessage . . . 716
8.682.2.4 storageType . . . 716
8.683 pack_sms_SendSMS_t Struct Reference . . . 716
8.683.1 Detailed Description . . . 716
8.683.2 Field Documentation . . . 716
8.683.2.1 messageFormat . . . 716
8.683.2.2 messageSize . . . 716
8.683.2.3 pLinktimer . . . 717
8.683.2.4 pMessage . . . 717
8.684 pack_sms_SetNewSMSCallback_t Struct Reference . . . 717
8.684.1 Detailed Description . . . 717
8.684.2 Field Documentation . . . 717
8.684.2.1 status . . . 717
8.685 pack_sms_SetSMSCAddress_t Struct Reference . . . 717
8.685.1 Detailed Description . . . 717
8.685.2 Field Documentation . . . 717
8.685.2.1 pSMSCAddress . . . 717
8.685.2.2 pSMSCType . . . 717
8.686 pack_sms_SLQSDeleteSMS_t Struct Reference . . . 717
8.686.1 Detailed Description . . . 718
8.686.2 Field Documentation . . . 718
8.686.2.1 pMessageIndex . . . 718
8.686.2.2 pMessageMode . . . 718
8.686.2.3 pMessageTag . . . 718
8.686.2.4 storageType . . . 718
8.687 pack_sms_SLQSGetSMS_t Struct Reference . . . 718
8.687.1 Detailed Description . . . 719
8.687.2 Field Documentation . . . 719
8.687.2.1 messageIndex . . . 719
8.687.2.2 pMessageMode . . . 719
8.687.2.3 storageType . . . 719
8.688 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 719
8.688.1 Detailed Description . . . 719
8.688.2 Field Documentation . . . 719
8.688.2.1 mode . . . 719
8.689 pack_sms_SLQSGetSMSList_t Struct Reference . . . 719
8.689.1 Detailed Description . . . 720
8.689.2 Field Documentation . . . 720
8.689.2.1 pMessageMode . . . 720
8.689.2.2 pRequestedTag . . . 720

8.689.2.3 storageType . . . 720
8.690 pack_sms_SLQSModifySMSStatus_t Struct Reference . . . 720
8.690.1 Detailed Description . . . 720
8.690.2 Field Documentation . . . 721
8.690.2.1 messageIndex . . . 721
8.690.2.2 messageTag . . . 721
8.690.2.3 pMessageMode . . . 721
8.690.2.4 storageType . . . 721
8.691 pack_sms_SLQSSendAsyncSMS_t Struct Reference . . . 721
8.691.1 Detailed Description . . . 721
8.691.2 Field Documentation . . . 721
8.691.2.1 pSendSmsParams . . . 721
8.692 pack_sms_SLQSSetIndicationRegister_t Struct Reference . . . 721
8.692.1 Detailed Description . . . 722
8.692.2 Field Documentation . . . 722
8.692.2.1 pSetIndicationRegReq . . . 722
8.693 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . 722
8.693.1 Detailed Description . . . 722
8.693.2 Field Documentation . . . 722
8.693.2.1 broadcastActivate . . . 722
8.693.2.2 mode . . . 722
8.694 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference . . . 722
8.694.1 Detailed Description . . . 723
8.694.2 Field Documentation . . . 723
8.694.2.1 mode . . . 723
8.694.2.2 pBroadcastConfig . . . 723
8.694.2.3 pCDMABroadcastConfig . . . 723
8.695 pack_sms_SLQSSetSmsStorage_t Struct Reference . . . 723
8.695.1 Detailed Description . . . 723
8.695.2 Field Documentation . . . 723
8.695.2.1 smsStorage . . . 724
8.696 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 724
8.696.1 Detailed Description . . . 724
8.696.2 Field Documentation . . . 724
8.696.2.1 pMaxStorageSizeReq . . . 724
8.697 pack_sms_SLQSSmsSetRoutes_t Struct Reference . . . 724
8.697.1 Detailed Description . . . 724
8.697.2 Field Documentation . . . 724
8.697.2.1 pSetRoutesReq . . . 724
8.698 pack_swidms_SLQSSwiDmsSetMTU_t Struct Reference . . . 724

8.698.1 Detailed Description . . . 725
8.698.2 Field Documentation . . . 725
8.698.2.1 MTUSize . . . 725
8.699 pack_swidms_SLQSSwiDmsSetUsbNetNum_t Struct Reference . . . 725
8.699.1 Detailed Description . . . 725
8.699.2 Field Documentation . . . 725
8.699.2.1 nUsbNetNum . . . 725
8.700 pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . 725
8.700.1 Detailed Description . . . 726
8.700.2 Field Documentation . . . 727
8.700.2.1 fix_rate . . . 727
8.700.2.2 fix_type . . . 727
8.700.2.3 function . . . 727
8.700.2.4 max_dist . . . 727
8.700.2.5 max_time . . . 727
8.700.2.6 set_fix_rate . . . 727
8.700.2.7 set_fix_type . . . 727
8.700.2.8 set_function . . . 727
8.700.2.9 set_max_dist . . . 727
8.700.2.10 set_max_time . . . 727
8.701 pack_swioma_SLQSOMADMCancelSession_t Struct Reference . . . 727
8.701.1 Detailed Description . . . 727
8.701.2 Field Documentation . . . 727
8.701.2.1 sessionType . . . 727
8.702 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 727
8.702.1 Detailed Description . . . 727
8.702.2 Field Documentation . . . 728
8.702.2.1 SessionType . . . 728
8.703 pack_swioma_SLQSOMADMSendSelection_t Struct Reference . . . 728
8.703.1 Detailed Description . . . 728
8.703.2 Field Documentation . . . 728
8.703.2.1 pDeferTime . . . 728
8.703.2.2 pRejectReason . . . 728
8.703.2.3 selection . . . 728
8.704 pack_swioma_SLQSOMADMSetSettings_t Struct Reference . . . 728
8.704.1 Detailed Description . . . 729
8.704.2 Field Documentation . . . 729
8.704.2.1 FOTAdownload . . . 729
8.704.2.2 FOTAUpdate . . . 729
8.704.2.3 pAutosdm . . . 729

8.704.2.4 pFwAutoCheck . . . 729
8.705 pack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 729
8.705.1 Detailed Description . . . 730
8.705.2 Field Documentation . . . 730
8.705.2.1 sessionType . . . 730
8.706 pack_uim_ChangePin_t Struct Reference . . . 730
8.706.1 Detailed Description . . . 730
8.706.2 Field Documentation . . . 731
8.706.2.1 changePIN . . . 731
8.706.2.2 EncryptedPIN1 . . . 731
8.706.2.3 pIndicationToken . . . 731
8.706.2.4 pKeyReferenceID . . . 731
8.706.2.5 sessionInfo . . . 731
8.706.2.6 Tlvresult . . . 731
8.707 pack_uim_ReadTransparent_t Struct Reference . . . 731
8.707.1 Detailed Description . . . 731
8.707.2 Field Documentation . . . 732
8.707.2.1 fileIndex . . . 732
8.707.2.2 pEncryptData . . . 732
8.707.2.3 pIndicationToken . . . 732
8.707.2.4 readTransparent . . . 732
8.707.2.5 sessionInfo . . . 732
8.707.2.6 Tlvresult . . . 732
8.708 pack_uim_SetPinProtection_t Struct Reference . . . 732
8.708.1 Detailed Description . . . 732
8.708.2 Field Documentation . . . 733
8.708.2.1 EncryptedPIN1 . . . 733
8.708.2.2 pIndicationToken . . . 733
8.708.2.3 pinProtection . . . 733
8.708.2.4 pKeyReferenceID . . . 733
8.708.2.5 sessionInfo . . . 733
8.708.2.6 Tlvresult . . . 733
8.709 pack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 733
8.709.1 Detailed Description . . . 733
8.709.2 Field Documentation . . . 734
8.709.2.1 authData . . . 734
8.709.2.2 pIndicationToken . . . 734
8.709.2.3 sessionInfo . . . 734
8.710 pack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 734
8.710.1 Detailed Description . . . 734

8.710.2 Field Documentation . . . 734
8.710.2.1 depersonilisationInfo . . . 734
8.711 pack_uim_SLQSUIMEventRegister_t Struct Reference . . . 734
8.711.1 Detailed Description . . . 734
8.711.2 Field Documentation . . . 734
8.711.2.1 eventMask . . . 734
8.712 pack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 735
8.712.1 Detailed Description . . . 735
8.712.2 Field Documentation . . . 735
8.712.2.1 pConfigurationMask . . . 735
8.713 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 735
8.713.1 Detailed Description . . . 735
8.713.2 Field Documentation . . . 736
8.713.2.1 fileIndex . . . 736
8.713.2.2 pIndicationToken . . . 736
8.713.2.3 sessionInfo . . . 736
8.714 pack_uim_SLQSUIMPowerDown_t Struct Reference . . . 736
8.714.1 Detailed Description . . . 736
8.714.2 Field Documentation . . . 736
8.714.2.1 slot . . . 736
8.715 pack_uim_SLQSUIMPowerUp_t Struct Reference . . . 736
8.715.1 Detailed Description . . . 736
8.715.2 Field Documentation . . . 737
8.715.2.1 pIgnoreHotSwapSwitch . . . 737
8.715.2.2 slot . . . 737
8.716 pack_uim_SLQSUIMRefreshComplete_t Struct Reference . . . 737
8.716.1 Detailed Description . . . 737
8.716.2 Field Documentation . . . 737
8.716.2.1 refreshComplete . . . 737
8.716.2.2 sessionInfo . . . 737
8.717 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 737
8.717.1 Detailed Description . . . 737
8.717.2 Field Documentation . . . 738
8.717.2.1 sessionInfo . . . 738
8.718 pack_uim_SLQSUIMRefreshOK_t Struct Reference . . . 738
8.718.1 Detailed Description . . . 738
8.718.2 Field Documentation . . . 738
8.718.2.1 OKtoRefresh . . . 738
8.718.2.2 sessionInfo . . . 738
8.719 pack_uim_SLQSUIMRefreshRegister_t Struct Reference . . . 738

8.719.1 Detailed Description . . . 738
8.719.2 Field Documentation . . . 739
8.719.2.1 regRefresh . . . 739
8.719.2.2 sessionInfo . . . 739
8.720 pack_uim_SLQSUIMSwitchSlot_t Struct Reference . . . 739
8.720.1 Detailed Description . . . 739
8.720.2 Field Documentation . . . 739
8.720.2.1 bLogicalSlot . . . 739
8.720.2.2 ulPhysicalSlot . . . 739
8.721 pack_uim_UnblockPin_t Struct Reference . . . 740
8.721.1 Detailed Description . . . 740
8.721.2 Field Documentation . . . 740
8.721.2.1 EncryptedPIN1 . . . 740
8.721.2.2 pIndicationToken . . . 740
8.721.2.3 pinProtection . . . 740
8.721.2.4 pKeyReferenceID . . . 740
8.721.2.5 sessionInfo . . . 740
8.721.2.6 Tlvresult . . . 741
8.722 pack_uim_VerifyPin_t Struct Reference . . . 741
8.722.1 Detailed Description . . . 741
8.722.2 Field Documentation . . . 741
8.722.2.1 pEncryptedPIN1 . . . 741
8.722.2.2 pIndicationToken . . . 741
8.722.2.3 pKeyReferenceID . . . 742
8.722.2.4 sessionInfo . . . 742
8.722.2.5 Tlvresult . . . 742
8.722.2.6 verifyPIN . . . 742
8.723 pack_voice_AnswerUSSD_t Struct Reference . . . 742
8.723.1 Detailed Description . . . 742
8.723.2 Field Documentation . . . 742
8.723.2.1 pInfo . . . 742
8.724 pack_voice_OriginateUSSD_t Struct Reference . . . 742
8.724.1 Detailed Description . . . 742
8.724.2 Field Documentation . . . 742
8.724.2.1 pInfo . . . 742
8.725 pack_voice_SLQSOriginateUSSD_t Struct Reference . . . 742
8.725.1 Detailed Description . . . 743
8.725.2 Field Documentation . . . 743
8.725.2.1 ussData . . . 743
8.725.2.2 ussDCS . . . 743

8.725.2.3 ussLen . . . 743
8.726 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference . . . 743
8.726.1 Detailed Description . . . 743
8.726.2 Field Documentation . . . 743
8.726.2.1 lineValue . . . 744
8.727 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference . . . 744
8.727.1 Detailed Description . . . 744
8.727.2 Field Documentation . . . 744
8.727.2.1 switchOption . . . 744
8.728 pack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 744
8.728.1 Detailed Description . . . 744
8.728.2 Field Documentation . . . 744
8.728.2.1 pCallId . . . 744
8.729 pack_voice_SLQSVoiceBindSubscription_t Struct Reference . . . 744
8.729.1 Detailed Description . . . 745
8.729.2 Field Documentation . . . 745
8.729.2.1 subsType . . . 745
8.730 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 745
8.730.1 Detailed Description . . . 745
8.730.2 Field Documentation . . . 745
8.730.2.1 BurstDTMFInfo . . . 745
8.730.2.2 pBurstDTMFLengths . . . 745
8.731 pack_voice_SLQSVoiceDialCall_t Struct Reference . . . 745
8.731.1 Detailed Description . . . 746
8.731.2 Field Documentation . . . 747
8.731.2.1 callNumber . . . 747
8.731.2.2 pCallPartySubAdd . . . 747
8.731.2.3 pCallType . . . 747
8.731.2.4 pCLIRType . . . 747
8.731.2.5 pCUGInfo . . . 747
8.731.2.6 pEmergencyCategory . . . 747
8.731.2.7 pSvcType . . . 747
8.731.2.8 pUUSInFo . . . 747
8.732 pack_voice_SLQSVoiceEndCall_t Struct Reference . . . 747
8.732.1 Detailed Description . . . 747
8.732.2 Field Documentation . . . 747
8.732.2.1 pCallId . . . 747
8.733 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 747
8.733.1 Detailed Description . . . 748
8.733.2 Field Documentation . . . 748

8.733.2.1 pSvcClass . . . 748
8.733.2.2 reason . . . 748
8.734 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 748
8.734.1 Detailed Description . . . 748
8.734.2 Field Documentation . . . 749
8.734.2.1 pSvcClass . . . 749
8.734.2.2 Reason . . . 749
8.735 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 749
8.735.1 Detailed Description . . . 749
8.735.2 Field Documentation . . . 749
8.735.2.1 callID . . . 749
8.736 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 749
8.736.1 Detailed Description . . . 750
8.736.2 Field Documentation . . . 750
8.736.2.1 pSvcClass . . . 750
8.737 pack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 750
8.737.1 Detailed Description . . . 750
8.737.2 Field Documentation . . . 751
8.737.2.1 pAirTimer . . . 751
8.737.2.2 pAMRStatus . . . 751
8.737.2.3 pAutoAnswer . . . 751
8.737.2.4 pNamID . . . 751
8.737.2.5 pPrefVoicePrivacy . . . 751
8.737.2.6 pPrefVoiceSO . . . 751
8.737.2.7 pRoamTimer . . . 751
8.737.2.8 pTTYMode . . . 751
8.737.2.9 pVoiceDomainPref . . . 751
8.738 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference . . . 752
8.738.1 Detailed Description . . . 752
8.738.2 Field Documentation . . . 752
8.738.2.1 pRegDTMFEvents . . . 752
8.738.2.2 pRegVoicePrivacyEvents . . . 752
8.738.2.3 pSuppsNotifEvents . . . 752
8.739 pack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 752
8.739.1 Detailed Description . . . 753
8.739.2 Field Documentation . . . 753
8.739.2.1 pCallID . . . 753
8.739.2.2 SUPSType . . . 753
8.740 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference . . . 753
8.740.1 Detailed Description . . . 753

8.740.2 Field Documentation . . . 754
8.740.2.1 USSInformation . . . 754
8.741 pack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 754
8.741.1 Detailed Description . . . 754
8.741.2 Field Documentation . . . 754
8.741.2.1 pCallID . . . 754
8.741.2.2 pFlashPayLd . . . 754
8.741.2.3 pFlashType . . . 754
8.742 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 754
8.742.1 Detailed Description . . . 755
8.742.2 Field Documentation . . . 755
8.742.2.1 newPasswd . . . 755
8.742.2.2 newPasswdAgain . . . 755
8.742.2.3 oldPasswd . . . 755
8.742.2.4 Reason . . . 755
8.743 pack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 755
8.743.1 Detailed Description . . . 756
8.743.2 Field Documentation . . . 757
8.743.2.1 pAirTimerConfig . . . 757
8.743.2.2 pAutoAnswer . . . 757
8.743.2.3 pPrefVoiceDomain . . . 757
8.743.2.4 pPrefVoiceSO . . . 757
8.743.2.5 pRoamTimerConfig . . . 757
8.743.2.6 pTTYMode . . . 757
8.744 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference . . . 757
8.744.1 Detailed Description . . . 757
8.744.2 Field Documentation . . . 757
8.744.2.1 privacyPref . . . 757
8.745 pack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 757
8.745.1 Detailed Description . . . 757
8.745.2 Field Documentation . . . 759
8.745.2.1 pCallBarringPasswd . . . 759
8.745.2.2 pCallForwardingNumber . . . 759
8.745.2.3 pCallFwdTypeAndPlan . . . 759
8.745.2.4 pServiceClass . . . 759
8.745.2.5 pTimerVal . . . 759
8.745.2.6 reason . . . 759
8.745.2.7 voiceSvc . . . 759
8.746 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 759
8.746.1 Detailed Description . . . 759

8.746.2 Field Documentation . . . 760
8.746.2.1 DTMFdigit . . . 760
8.746.2.2 pCallID . . . 760
8.747pack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 760
8.747.1 Detailed Description . . . 760
8.747.2 Field Documentation . . . 760
8.747.2.1 callID . . . 760
8.748pack_wds_DHCPv4ClientLeaseChange_t Struct Reference . . . 760
8.748.1 Detailed Description . . . 760
8.748.2 Field Documentation . . . 761
8.748.2.1 pEnableNotification . . . 761
8.749pack_wds_GetDefaultProfile_t Struct Reference . . . 761
8.749.1 Detailed Description . . . 761
8.749.2 Field Documentation . . . 761
8.749.2.1 profiletype . . . 761
8.750pack_wds_GetDefaultProfileNum_t Struct Reference . . . 761
8.750.1 Detailed Description . . . 761
8.750.2 Field Documentation . . . 761
8.750.2.1 family . . . 761
8.750.2.2 type . . . 762
8.751pack_wds_GetDormancyState_t Struct Reference . . . 762
8.751.1 Detailed Description . . . 762
8.752pack_wds_GetLastMobileIPError_t Struct Reference . . . 762
8.752.1 Detailed Description . . . 762
8.753pack_wds_GetMobileIP_t Struct Reference . . . 762
8.753.1 Detailed Description . . . 762
8.754pack_wds_GetMobileIPProfile_t Struct Reference . . . 762
8.754.1 Detailed Description . . . 762
8.754.2 Field Documentation . . . 763
8.754.2.1 index . . . 763
8.755pack_wds_GetPacketStatistics_t Struct Reference . . . 763
8.755.1 Detailed Description . . . 763
8.755.2 Field Documentation . . . 763
8.755.2.1 pStatMask . . . 763
8.756pack_wds_GetPacketStatus_t Struct Reference . . . 763
8.756.1 Detailed Description . . . 763
8.756.2 Field Documentation . . . 764
8.756.2.1 statmask . . . 764
8.757pack_wds_GetSessionDuration_t Struct Reference . . . 764
8.757.1 Detailed Description . . . 764

8.758 pack_wds_RMSetTransferStatistics_t Struct Reference . . . 764
8.758.1 Detailed Description . . . 764
8.758.2 Field Documentation . . . 764
8.758.2.1 RmTrasnferStaticsReq . . . 764
8.759 pack_wds_SetAutoconnect_t Struct Reference . . . 765
8.759.1 Detailed Description . . . 765
8.759.2 Field Documentation . . . 765
8.759.2.1 acroamsetting . . . 765
8.759.2.2 acsetting . . . 765
8.760 pack_wds_SetDefaultProfile_t Struct Reference . . . 765
8.760.1 Detailed Description . . . 765
8.760.2 Field Documentation . . . 766
8.760.2.1 authentication . . . 767
8.760.2.2 ipAddress . . . 767
8.760.2.3 pApnname . . . 767
8.760.2.4 pdpType . . . 767
8.760.2.5 pName . . . 767
8.760.2.6 pPassword . . . 767
8.760.2.7 primaryDNS . . . 767
8.760.2.8 profileType . . . 767
8.760.2.9 pUsername . . . 767
8.760.2.10 secondaryDNS . . . 767
8.761 pack_wds_SetDefaultProfileNum_t Struct Reference . . . 767
8.761.1 Detailed Description . . . 767
8.761.2 Field Documentation . . . 767
8.761.2.1 family . . . 767
8.761.2.2 index . . . 767
8.761.2.3 type . . . 767
8.762 pack_wds_SetMobileIP_t Struct Reference . . . 768
8.762.1 Detailed Description . . . 768
8.762.2 Field Documentation . . . 768
8.762.2.1 mode . . . 768
8.763 pack_wds_SetMobileIPParameters_t Struct Reference . . . 768
8.763.1 Detailed Description . . . 768
8.763.2 Field Documentation . . . 769
8.763.2.1 pHA2002bis . . . 769
8.763.2.2 pHAAuthenticator . . . 769
8.763.2.3 pMode . . . 769
8.763.2.4 pReRegPeriod . . . 769
8.763.2.5 pReRegTraffic . . . 769

8.763.2.6 pRetryInterval . . . 769
8.763.2.7 pRetryLimit . . . 769
8.763.2.8 pSPC . . . 769
8.764 pack_wds_SetMobileIPProfile_t Struct Reference . . . 769
8.764.1 Detailed Description . . . 770
8.764.2 Field Documentation . . . 770
8.764.2.1 index . . . 771
8.764.2.2 pAAASPI . . . 771
8.764.2.3 pAddress . . . 771
8.764.2.4 pEnabled . . . 771
8.764.2.5 pHASPI . . . 771
8.764.2.6 pMNAAA . . . 771
8.764.2.7 pMNHA . . . 771
8.764.2.8 pNAI . . . 771
8.764.2.9 pPrimaryHA . . . 771
8.764.2.10 pRevTunneling . . . 771
8.764.2.11 pSecondaryHA . . . 771
8.764.2.12 spc . . . 771
8.765 pack_wds_SLQSCreateProfile_t Struct Reference . . . 771
8.765.1 Detailed Description . . . 771
8.765.2 Field Documentation . . . 772
8.765.2.1 pCurProfile . . . 772
8.765.2.2 pProfileId . . . 772
8.765.2.3 pProfileType . . . 772
8.766 pack_wds_SLQSDeleteProfile_t Struct Reference . . . 772
8.766.1 Detailed Description . . . 772
8.766.2 Field Documentation . . . 772
8.766.2.1 profileIndex . . . 772
8.766.2.2 profileType . . . 772
8.767 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 772
8.767.1 Detailed Description . . . 772
8.768 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 773
8.768.1 Detailed Description . . . 773
8.769 pack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 773
8.769.1 Detailed Description . . . 773
8.769.2 Field Documentation . . . 774
8.769.2.1 Mask . . . 774
8.769.2.2 pReportChannelRate . . . 774
8.769.2.3 pReportConnStatus . . . 774
8.769.2.4 pReportDataBearerTech . . . 774

8.769.2.5 pReportDormStatus . . . 774
8.769.2.6 pTransferStatInd . . . 774
8.770 pack_wds_SLQSGetProfileSettings_t Struct Reference . . . 774
8.770.1 Detailed Description . . . 774
8.770.2 Field Documentation . . . 775
8.770.2.1 ProfileId . . . 775
8.770.2.2 ProfileType . . . 775
8.771 pack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 775
8.771.1 Detailed Description . . . 775
8.771.2 Field Documentation . . . 776
8.771.2.1 pReqSettings . . . 776
8.772 pack_wds_SLQSModifyProfile_t Struct Reference . . . 776
8.772.1 Detailed Description . . . 776
8.772.2 Field Documentation . . . 777
8.772.2.1 curProfile . . . 777
8.772.2.2 pProfileId . . . 777
8.772.2.3 pProfileType . . . 777
8.773 pack_wds_SLQSSet3GPPConfigItem_t Struct Reference . . . 777
8.773.1 Detailed Description . . . 777
8.773.2 Field Documentation . . . 778
8.773.2.1 LTEAttachProfileListLen . . . 778
8.773.2.2 p3gppRelease . . . 778
8.773.2.3 pDefaultPDNEnabled . . . 778
8.773.2.4 pLTEAttachProfile . . . 778
8.773.2.5 pLTEAttachProfileList . . . 778
8.773.2.6 pProfileList . . . 779
8.774 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 779
8.774.1 Detailed Description . . . 779
8.774.2 Field Documentation . . . 779
8.774.2.1 IPFamilyPreference . . . 779
8.775 pack_wds_SLQSSetWdsEventCallback_t Struct Reference . . . 779
8.775.1 Detailed Description . . . 779
8.775.2 Field Documentation . . . 780
8.775.2.1 currentDataBearer . . . 780
8.775.2.2 dataBearer . . . 780
8.775.2.3 dataBearerTechExt . . . 780
8.775.2.4 dataSystemStatus . . . 780
8.775.2.5 dormancyStatus . . . 780
8.775.2.6 interval . . . 780
8.775.2.7 mobileIP . . . 780

8.775.2.8 transferStats . . . 780
8.776 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 780
8.776.1 Detailed Description . . . 780
8.776.2 Field Documentation . . . 780
8.776.2.1 pProfileId . . . 780
8.777 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference . . . 780
8.777.1 Detailed Description . . . 780
8.777.2 Field Documentation . . . 781
8.777.2.1 pHwConfig . . . 781
8.777.2.2 pProfileId . . . 781
8.777.2.3 pRequestOptionList . . . 781
8.778 pack_wds_SLQSSSetLoopback_t Struct Reference . . . 781
8.778.1 Detailed Description . . . 781
8.778.2 Field Documentation . . . 781
8.778.2.1 loopbackMode . . . 781
8.778.2.2 loopbackMultiplier . . . 781
8.779 pack_wds_SLQSStartDataSession_t Struct Reference . . . 781
8.779.1 Detailed Description . . . 782
8.779.2 Field Documentation . . . 782
8.779.2.1 pAuth . . . 782
8.779.2.2 pPass . . . 782
8.779.2.3 pprofileid3gpp . . . 782
8.779.2.4 pprofileid3gpp2 . . . 782
8.779.2.5 pTech . . . 782
8.779.2.6 pUser . . . 783
8.780 pack_wds_SLQSStopDataSession_t Struct Reference . . . 783
8.780.1 Detailed Description . . . 783
8.780.2 Field Documentation . . . 783
8.780.2.1 psid . . . 783
8.781 pack_wds_SLQSWdsSetEventReport_t Struct Reference . . . 783
8.781.1 Detailed Description . . . 783
8.781.2 Field Documentation . . . 784
8.781.2.1 pCurrChannelRateInd . . . 784
8.781.2.2 pCurrDataBearerTechInd . . . 784
8.781.2.3 pCurrPrefDataSysInd . . . 784
8.781.2.4 pDataBearerTechInd . . . 784
8.781.2.5 pDataCallStatusChangeInd . . . 784
8.781.2.6 pDataSystemStatusChangeInd . . . 785
8.781.2.7 pDormancyStatusInd . . . 785
8.781.2.8 pEVDOPageMonPerChangeInd . . . 785

8.781.2.9 pMIPStatusInd . . . 785
8.781.2.10 pTransferStatInd . . . 785
8.782 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 785
8.782.1 Detailed Description . . . 785
8.782.2 Field Documentation . . . 785
8.782.2.1 contextId . . . 785
8.782.2.2 contextType . . . 785
8.783 PackCreateProfileOut Struct Reference . . . 785
8.783.1 Detailed Description . . . 785
8.783.2 Field Documentation . . . 786
8.783.2.1 ExtErrorCode . . . 786
8.783.2.2 ProfileIndex . . . 786
8.783.2.3 ProfileType . . . 786
8.784 packgetDyingGaspCfg Struct Reference . . . 786
8.784.1 Detailed Description . . . 786
8.784.2 Field Documentation . . . 786
8.784.2.1 pDestSMSContent . . . 786
8.784.2.2 pDestSMSNum . . . 786
8.785 packgetDyingGaspStatistics Struct Reference . . . 786
8.785.1 Detailed Description . . . 787
8.785.2 Field Documentation . . . 787
8.785.2.1 pSMSAttemptedFlag . . . 787
8.785.2.2 pTimeStamp . . . 787
8.786 PCMparams Struct Reference . . . 787
8.786.1 Detailed Description . . . 787
8.786.2 Field Documentation . . . 787
8.786.2.1 iFaceTab . . . 787
8.786.2.2 iFaceTabLen . . . 787
8.787 PCSCFFQDNAddress Struct Reference . . . 788
8.787.1 Detailed Description . . . 788
8.787.2 Field Documentation . . . 788
8.787.2.1 fqdnAddr . . . 788
8.787.2.2 fqdnLen . . . 788
8.788 PCSCFFQDNAddressList Struct Reference . . . 788
8.788.1 Detailed Description . . . 788
8.788.2 Field Documentation . . . 788
8.788.2.1 numInstances . . . 788
8.788.2.2 pcsfFQDNAddress . . . 788
8.789 PCSCFIPv4ServerAddressList Struct Reference . . . 789
8.789.1 Detailed Description . . . 789

8.789.2 Field Documentation . . . 789
8.789.2.1 numInstances . . . 789
8.789.2.2 pscsfIPv4Addr . . . 789
8.790 PDSPositionData Struct Reference . . . 789
8.790.1 Detailed Description . . . 789
8.790.2 Field Documentation . . . 791
8.790.2.1 pAltitudeWrtEllipsoid . . . 791
8.790.2.2 pAltitudeWrtSealevel . . . 791
8.790.2.3 pHorizontalConfidence . . . 791
8.790.2.4 pHorizontalUncCircular . . . 791
8.790.2.5 pLatitude . . . 791
8.790.2.6 pLongitude . . . 791
8.790.2.7 pPositionSource . . . 791
8.790.2.8 pTimeStamp . . . 791
8.790.2.9 pTimeType . . . 791
8.790.2.10 pVerticalConfidence . . . 791
8.790.2.11 pVerticalUnc . . . 791
8.791 PDSPosMethodStateReq Struct Reference . . . 791
8.791.1 Detailed Description . . . 791
8.791.2 Field Documentation . . . 792
8.791.2.1 pWifiState . . . 792
8.791.2.2 pXtraDataState . . . 792
8.791.2.3 pXtraTimeState . . . 792
8.792 peerNumberInfo Struct Reference . . . 792
8.792.1 Detailed Description . . . 792
8.792.2 Field Documentation . . . 793
8.792.2.1 callID . . . 793
8.792.2.2 number . . . 793
8.792.2.3 numLen . . . 793
8.792.2.4 numPI . . . 793
8.792.2.5 numPlan . . . 793
8.792.2.6 numSI . . . 794
8.792.2.7 numType . . . 794
8.793 personalizationStatus Struct Reference . . . 794
8.793.1 Detailed Description . . . 794
8.793.2 Field Documentation . . . 794
8.793.2.1 feature . . . 794
8.793.2.2 numFeatures . . . 794
8.793.2.3 unblockLeft . . . 794
8.793.2.4 verifyLeft . . . 794

8.794 PhyCaAggPcellInfo Struct Reference . . . 795
8.794.1 Detailed Description . . . 795
8.794.2 Field Documentation . . . 795
8.794.2.1 dl_bw_value . . . 795
8.794.2.2 freq . . . 795
8.794.2.3 iLTEbandValue . . . 795
8.794.2.4 pci . . . 795
8.794.2.5 TlvPresent . . . 795
8.795 PhyCaAggScellDlBw Struct Reference . . . 795
8.795.1 Detailed Description . . . 796
8.795.2 Field Documentation . . . 796
8.795.2.1 dl_bw_value . . . 796
8.795.2.2 TlvPresent . . . 796
8.796 PhyCaAggScellIndex Struct Reference . . . 796
8.796.1 Detailed Description . . . 796
8.796.2 Field Documentation . . . 796
8.796.2.1 scell_idx . . . 796
8.796.2.2 TlvPresent . . . 796
8.797 PhyCaAggScellIndType Struct Reference . . . 796
8.797.1 Detailed Description . . . 797
8.797.2 Field Documentation . . . 797
8.797.2.1 freq . . . 797
8.797.2.2 pci . . . 797
8.797.2.3 scell_state . . . 797
8.797.2.4 TlvPresent . . . 797
8.798 PhyCaAggScellInfo Struct Reference . . . 797
8.798.1 Detailed Description . . . 797
8.798.2 Field Documentation . . . 800
8.798.2.1 dl_bw_value . . . 800
8.798.2.2 freq . . . 800
8.798.2.3 iLTEbandValue . . . 800
8.798.2.4 pci . . . 800
8.798.2.5 scell_state . . . 800
8.798.2.6 TlvPresent . . . 800
8.799 PilotSetData Struct Reference . . . 800
8.799.1 Detailed Description . . . 800
8.799.2 Field Documentation . . . 800
8.799.2.1 NumPilots . . . 800
8.799.2.2 pPilotSetInfo . . . 800
8.800 PilotSetParams Struct Reference . . . 801

8.800.1 Detailed Description . . . 801
8.800.2 Field Documentation . . . 801
8.800.2.1 PilotPN . . . 801
8.800.2.2 PilotStrength . . . 801
8.800.2.3 PilotType . . . 801
8.801 pktErrRate Struct Reference . . . 801
8.801.1 Detailed Description . . . 801
8.801.2 Field Documentation . . . 801
8.801.2.1 exponent . . . 801
8.801.2.2 multiplier . . . 802
8.802 PlmnID Struct Reference . . . 802
8.802.1 Detailed Description . . . 802
8.802.2 Field Documentation . . . 802
8.802.2.1 mcc . . . 802
8.802.2.2 mnc . . . 802
8.802.2.3 pcsDigit . . . 802
8.802.2.4 TlvPresent . . . 802
8.803 PLMNNetworkName Struct Reference . . . 802
8.803.1 Detailed Description . . . 803
8.803.2 Field Documentation . . . 803
8.803.2.1 numInstance . . . 803
8.803.2.2 PLMNNetName . . . 803
8.804 PLMNNetworkNameData Struct Reference . . . 803
8.804.1 Detailed Description . . . 803
8.804.2 Field Documentation . . . 804
8.804.2.1 codingScheme . . . 804
8.804.2.2 countryInitials . . . 804
8.804.2.3 longName . . . 804
8.804.2.4 longNameLen . . . 804
8.804.2.5 longNameSpareBits . . . 804
8.804.2.6 shortName . . . 804
8.804.2.7 shortNameLen . . . 805
8.804.2.8 shortNameSpareBits . . . 805
8.805 Port Struct Reference . . . 805
8.805.1 Detailed Description . . . 805
8.805.2 Field Documentation . . . 805
8.805.2.1 port . . . 805
8.805.2.2 range . . . 805
8.806 PowerSaveModeConfigReq Struct Reference . . . 805
8.806.1 Detailed Description . . . 805

8.806.2 Field Documentation . . . 806
8.806.2.1 pActiveTimer . . . 806
8.806.2.2 pPeriodicUpdateTimer . . . 806
8.806.2.3 pPsmEnableState . . . 806
8.807 PowerSaveModeConfigResp Struct Reference . . . 806
8.807.1 Detailed Description . . . 806
8.807.2 Field Documentation . . . 807
8.807.2.1 pActiveTimer . . . 807
8.807.2.2 pDurationDueToOOS . . . 807
8.807.2.3 pDurationThreshold . . . 807
8.807.2.4 pEarlyWakeupTime . . . 807
8.807.2.5 pPeriodicUpdateTimer . . . 807
8.807.2.6 pPsmEnableState . . . 807
8.807.2.7 pRandomizationWindow . . . 807
8.808 precisionDilution_s Struct Reference . . . 807
8.808.1 Detailed Description . . . 807
8.808.2 Field Documentation . . . 807
8.808.2.1 HDOP . . . 807
8.808.2.2 PDOP . . . 807
8.808.2.3 VDOP . . . 807
8.809 PrefImageList Struct Reference . . . 808
8.809.1 Detailed Description . . . 808
8.809.2 Field Documentation . . . 808
8.809.2.1 listEntries . . . 808
8.809.2.2 listSize . . . 808
8.810 prefVoiceSO Struct Reference . . . 808
8.810.1 Detailed Description . . . 808
8.810.2 Field Documentation . . . 810
8.810.2.1 evrcCapability . . . 810
8.810.2.2 homeOrigVoiceSO . . . 810
8.810.2.3 homePageVoiceSO . . . 810
8.810.2.4 namID . . . 810
8.810.2.5 roamOrigVoiceSO . . . 810
8.811 Profile3GPP Struct Reference . . . 810
8.811.1 Detailed Description . . . 811
8.811.2 Field Documentation . . . 815
8.811.2.1 pAddrAllocPref . . . 815
8.811.2.2 pAPNClass . . . 815
8.811.2.3 pAPNDisabledFlag . . . 815
8.811.2.4 pAPNName . . . 815

8.811.2.5 pAPNnameSize . . . 815
8.811.2.6 pAuthenticationPref . . . 815
8.811.2.7 pGPRSMinimumQoS . . . 815
8.811.2.8 pGPRSRequestedQos . . . 815
8.811.2.9 pImCnFlag . . . 815
8.811.2.10 pIPv4AddrPref . . . 815
8.811.2.11 pIPv6AddPref . . . 815
8.811.2.12 pPassword . . . 815
8.811.2.13 pPasswordSize . . . 815
8.811.2.14 pPcscfAddrUsingDhcp . . . 815
8.811.2.15 pPcscfAddrUsingPCO . . . 815
8.811.2.16 pPDNInactivTimeout . . . 815
8.811.2.17 pPdpAccessConFlag . . . 815
8.811.2.18 pPdpContext . . . 815
8.811.2.19 pPdpDataCompType . . . 815
8.811.2.20 pPdpHdrCompType . . . 816
8.811.2.21 pPDPtype . . . 816
8.811.2.22 pPriDNSIPv4AddPref . . . 816
8.811.2.23 pPriDNSIPv6addpref . . . 816
8.811.2.24 pPrimaryID . . . 816
8.811.2.25 pProfilename . . . 816
8.811.2.26 pProfilenameSize . . . 816
8.811.2.27 pQosClassID . . . 816
8.811.2.28 pSecDNSIPv4AddPref . . . 816
8.811.2.29 pSecDNSIPv6addpref . . . 816
8.811.2.30 pSecondaryFlag . . . 816
8.811.2.31 pSupportEmergencyCalls . . . 816
8.811.2.32 pTFTID1Params . . . 816
8.811.2.33 pTFTID2Params . . . 816
8.811.2.34 pUMTSMinQoS . . . 816
8.811.2.35 pUMTSMinQosSigInd . . . 816
8.811.2.36 pUMTSReqQoS . . . 816
8.811.2.37 pUMTSReqQoSSigInd . . . 816
8.811.2.38 pUsername . . . 816
8.811.2.39 pUsernameSize . . . 816
8.812 Profile3GPP2 Struct Reference . . . 816
8.812.1 Detailed Description . . . 817
8.812.2 Field Documentation . . . 820
8.812.2.1 pAllowLinger . . . 820
8.812.2.2 pAPNClass3GPP2 . . . 820

8.812.2.3 pAPNEnabled3GPP2 . . . 820
8.812.2.4 pApnString . . . 820
8.812.2.5 pApnStringSize . . . 820
8.812.2.6 pAppPriority . . . 820
8.812.2.7 pAppType . . . 820
8.812.2.8 pAuthPassword . . . 821
8.812.2.9 pAuthPasswordSize . . . 821
8.812.2.10 pAuthProtocol . . . 821
8.812.2.11 pAuthRetryCount . . . 821
8.812.2.12 pAuthTimeout . . . 821
8.812.2.13 pDataMode . . . 821
8.812.2.14 pDataRate . . . 821
8.812.2.15 pIpcpAckTimeout . . . 821
8.812.2.16 pIpcpCreqRetryCount . . . 821
8.812.2.17 pIsPcscfAddressNedded . . . 821
8.812.2.18 pLcpAckTimeout . . . 821
8.812.2.19 pLcpCreqRetryCount . . . 821
8.812.2.20 pNegoDnsSrvrPref . . . 821
8.812.2.21 pPDNInactivTimeout3GPP2 . . . 821
8.812.2.22 pPdnType . . . 821
8.812.2.23 pPppSessCloseTimer1x . . . 821
8.812.2.24 pPppSessCloseTimerDO . . . 821
8.812.2.25 pPrimaryV4DnsAddress . . . 821
8.812.2.26 pPriV6DnsAddress . . . 821
8.812.2.27 pRATType . . . 821
8.812.2.28 pSecondaryV4DnsAddress . . . 821
8.812.2.29 pSecV6DnsAddress . . . 821
8.812.2.30 pUserId . . . 821
8.812.2.31 pUserIdSize . . . 821
8.813 ProfileIdentifier Struct Reference . . . 821
8.813.1 Detailed Description . . . 822
8.813.2 Field Documentation . . . 822
8.813.2.1 profileIndex . . . 822
8.813.2.2 profileType . . . 822
8.814 protocolSubtypeElement Struct Reference . . . 822
8.814.1 Detailed Description . . . 822
8.814.2 Field Documentation . . . 823
8.814.2.1 AccessMac . . . 823
8.814.2.2 AuthProt . . . 823
8.814.2.3 ControlMac . . . 823

8.814.2.4 EncryptProt . . . 823
8.814.2.5 ForwardMac . . . 823
8.814.2.6 IdleState . . . 823
8.814.2.7 KeyExchange . . . 823
8.814.2.8 MultDisc . . . 823
8.814.2.9 PhysicalLayer . . . 823
8.814.2.10 ReverseMac . . . 823
8.814.2.11 SecProt . . . 824
8.814.2.12 VirtStream . . . 824
8.815 PSDetachReq Struct Reference . . . 824
8.815.1 Detailed Description . . . 824
8.815.2 Field Documentation . . . 824
8.815.2.1 pDetachAction . . . 824
8.816 PSMActiveTimerTlv Struct Reference . . . 824
8.816.1 Detailed Description . . . 824
8.816.2 Field Documentation . . . 824
8.816.2.1 activeTimer . . . 824
8.816.2.2 TlvPresent . . . 824
8.817 PSMDurationDueToOOSTlv Struct Reference . . . 825
8.817.1 Detailed Description . . . 825
8.817.2 Field Documentation . . . 825
8.817.2.1 durationDueToOOS . . . 825
8.817.2.2 TlvPresent . . . 825
8.818 PSMDurationThresholdTlv Struct Reference . . . 825
8.818.1 Detailed Description . . . 825
8.818.2 Field Documentation . . . 825
8.818.2.1 durationThreshold . . . 825
8.818.2.2 TlvPresent . . . 825
8.819 PSMEarlyWakeupTimeTlv Struct Reference . . . 826
8.819.1 Detailed Description . . . 826
8.819.2 Field Documentation . . . 826
8.819.2.1 earlyWakeupTime . . . 826
8.819.2.2 TlvPresent . . . 826
8.820 PSMEnableStateTlv Struct Reference . . . 826
8.820.1 Detailed Description . . . 826
8.820.2 Field Documentation . . . 826
8.820.2.1 enableState . . . 826
8.820.2.2 TlvPresent . . . 827
8.821 PSMPeriodicUpdateTimerTlv Struct Reference . . . 827
8.821.1 Detailed Description . . . 827

8.821.2 Field Documentation . . . 827
8.821.2.1 periodicUpdateTimer . . . 827
8.821.2.2 TlvPresent . . . 827
8.822 PSMRandomizationWindowTlv Struct Reference . . . 827
8.822.1 Detailed Description . . . 827
8.822.2 Field Documentation . . . 827
8.822.2.1 randomizationWindow . . . 827
8.822.2.2 TlvPresent . . . 828
8.823 qaQmi3Gpp2TimeZone Struct Reference . . . 828
8.823.1 Detailed Description . . . 828
8.823.2 Field Documentation . . . 828
8.823.2.1 daylightSavings . . . 828
8.823.2.2 leapSeconds . . . 828
8.823.2.3 localTimeOffset . . . 828
8.824 qaQmiInterfaceInfo Struct Reference . . . 828
8.824.1 Detailed Description . . . 828
8.824.2 Field Documentation . . . 829
8.824.2.1 qaQmiinstanceid . . . 829
8.824.2.2 qaQmisvctype . . . 829
8.824.2.3 v4sessionId . . . 829
8.824.2.4 v6sessionId . . . 829
8.825 qaQmiServingSystemParam Struct Reference . . . 829
8.825.1 Detailed Description . . . 830
8.825.2 Field Documentation . . . 832
8.825.2.1 BasestationID . . . 832
8.825.2.2 BasestationLatitude . . . 832
8.825.2.3 BasestationLongitude . . . 832
8.825.2.4 CallBarStatus . . . 832
8.825.2.5 CDMA_P_Rev . . . 832
8.825.2.6 CDMASystemInfoExt . . . 832
8.825.2.7 CellID . . . 832
8.825.2.8 concSvcInfo . . . 832
8.825.2.9 CurrentPLMN . . . 832
8.825.2.10 DataSrvCapabilities . . . 832
8.825.2.11 defaultRoamInd . . . 832
8.825.2.12 DetailedSvcInfo . . . 832
8.825.2.13 DTMInd . . . 832
8.825.2.14 Gpp2TimeZone . . . 832
8.825.2.15 GppNetworkDSTAdjustment . . . 832
8.825.2.16 GppTimeZone . . . 832

8.825.2.17 hdrPersonality . . . 833
8.825.2.18 Lac . . . 833
8.825.2.19 NetworkID . . . 833
8.825.2.20 PRLInd . . . 833
8.825.2.21 roamIndicatorVal . . . 833
8.825.2.22 RoamingIndicatorList . . . 833
8.825.2.23 ServingSystem . . . 833
8.825.2.24 SystemID . . . 833
8.825.2.25 trackAreaCode . . . 833
8.826 QmiCbkCatEventStatusReportInd Struct Reference . . . 833
8.826.1 Field Documentation . . . 833
8.826.1.1 CCETlv . . . 833
8.826.1.2 event_Index . . . 833
8.827 QmiCbkLocBestAvailPosInd Struct Reference . . . 833
8.827.1 Detailed Description . . . 834
8.827.2 Field Documentation . . . 838
8.827.2.1 pAltitudeWrtEllipsoid . . . 838
8.827.2.2 pAltitudeWrtMeanSeaLevel . . . 838
8.827.2.3 pGpsTime . . . 838
8.827.2.4 pHeading . . . 838
8.827.2.5 pHeadingUnc . . . 838
8.827.2.6 pHorCirConf . . . 838
8.827.2.7 pHorEllpConf . . . 838
8.827.2.8 pHorReliability . . . 838
8.827.2.9 pHorUncCircular . . . 838
8.827.2.10 pHorUncEllipseOrientAzimuth . . . 838
8.827.2.11 pHorUncEllipseSemiMajor . . . 838
8.827.2.12 pHorUncEllipseSemiMinor . . . 838
8.827.2.13 pLatitude . . . 838
8.827.2.14 pLongitude . . . 838
8.827.2.15 pMagneticDeviation . . . 838
8.827.2.16 pPrecisionDilution . . . 838
8.827.2.17 pSensorDataUsage . . . 838
8.827.2.18 pSpeedHorizontal . . . 838
8.827.2.19 pSpeedUnc . . . 838
8.827.2.20 pSpeedVertical . . . 838
8.827.2.21 pSpeedVerticalUnc . . . 838
8.827.2.22 pSvUsedforFix . . . 838
8.827.2.23 pTechnologyMask . . . 838
8.827.2.24 pTimeSrc . . . 838

8.827.2.25 pTimestampUtc . . . 838
8.827.2.26 pTimeUnc . . . 838
8.827.2.27 pVertConfidence . . . 838
8.827.2.28 pVertReliability . . . 839
8.827.2.29 pVertUnc . . . 839
8.827.2.30 pXid . . . 839
8.827.2.31 status . . . 839
8.828 QmiCbkLocCradleMountInd Struct Reference . . . 839
8.828.1 Detailed Description . . . 839
8.828.2 Field Documentation . . . 839
8.828.2.1 cradleMountConfigStatus . . . 839
8.829 QmiCbkLocEngineStateInd Struct Reference . . . 839
8.829.1 Detailed Description . . . 839
8.829.2 Field Documentation . . . 840
8.829.2.1 engineState . . . 840
8.830 QmiCbkLocEventTimeSyncInd Struct Reference . . . 840
8.830.1 Detailed Description . . . 840
8.830.2 Field Documentation . . . 840
8.830.2.1 timeSyncRefCounter . . . 840
8.831 QmiCbkLocGetServerInd Struct Reference . . . 840
8.831.1 Detailed Description . . . 840
8.831.2 Field Documentation . . . 841
8.831.2.1 pIPv4AddrInfo . . . 841
8.831.2.2 pIPv6AddrInfo . . . 841
8.831.2.3 pURL . . . 841
8.831.2.4 serverStatus . . . 841
8.831.2.5 serverType . . . 841
8.832 QmiCbkLocInjectPositionInd Struct Reference . . . 842
8.832.1 Detailed Description . . . 842
8.832.2 Field Documentation . . . 842
8.832.2.1 status . . . 842
8.833 QmiCbkLocInjectSensorDataInd Struct Reference . . . 842
8.833.1 Detailed Description . . . 843
8.833.2 Field Documentation . . . 843
8.833.2.1 injectSensorDataStatus . . . 843
8.833.2.2 pAccelSamplesAccepted . . . 843
8.833.2.3 pAccelTempSamplesAccepted . . . 843
8.833.2.4 pGyroSamplesAccepted . . . 843
8.833.2.5 pGyroTempSamplesAccepted . . . 843
8.833.2.6 pOpaqueIdentifier . . . 843

8.834 QmiCbkLocInjectTimeInd Struct Reference . . . 844
8.834.1 Detailed Description . . . 844
8.834.2 Field Documentation . . . 844
8.834.2.1 injectTimeSyncStatus . . . 844
8.835 QmiCbkLocInjectUTCTimeInd Struct Reference . . . 844
8.835.1 Detailed Description . . . 844
8.835.2 Field Documentation . . . 845
8.835.2.1 status . . . 845
8.836 QmiCbkLocPositionReportInd Struct Reference . . . 845
8.836.1 Detailed Description . . . 846
8.836.2 Field Documentation . . . 849
8.836.2.1 pAltitudeAssumed . . . 849
8.836.2.2 pAltitudeWrtEllipsoid . . . 849
8.836.2.3 pAltitudeWrtMeanSeaLevel . . . 849
8.836.2.4 pFixId . . . 850
8.836.2.5 pGpsTime . . . 850
8.836.2.6 pHeading . . . 850
8.836.2.7 pHeadingUnc . . . 850
8.836.2.8 pHorConfidence . . . 850
8.836.2.9 pHorReliability . . . 850
8.836.2.10 pHorUncCircular . . . 850
8.836.2.11 pHorUncEllipseOrientAzimuth . . . 850
8.836.2.12 pHorUncEllipseSemiMajor . . . 850
8.836.2.13 pHorUncEllipseSemiMinor . . . 850
8.836.2.14 pLatitude . . . 850
8.836.2.15 pLeapSeconds . . . 850
8.836.2.16 pLongitude . . . 850
8.836.2.17 pMagneticDeviation . . . 850
8.836.2.18 pPrecisionDilution . . . 850
8.836.2.19 pSensorDataUsage . . . 850
8.836.2.20 pSpeedHorizontal . . . 850
8.836.2.21 pSpeedUnc . . . 850
8.836.2.22 pSpeedVertical . . . 850
8.836.2.23 pSvUsedforFix . . . 850
8.836.2.24 pTechnologyMask . . . 850
8.836.2.25 pTimeSrc . . . 850
8.836.2.26 pTimestampUtc . . . 850
8.836.2.27 pTimeUnc . . . 850
8.836.2.28 pVertConfidence . . . 850
8.836.2.29 pVertReliability . . . 850

8.836.2.30 pVertUnc . . . 850
8.836.2.31 sessionId . . . 850
8.836.2.32 sessionStatus . . . 851
8.837 QmiCbkLocSensorStreamingInd Struct Reference . . . 851
8.837.1 Detailed Description . . . 851
8.837.2 Field Documentation . . . 851
8.837.2.1 pAccelAcceptReady . . . 851
8.837.2.2 pAccelTempAcceptReady . . . 851
8.837.2.3 pGyroAcceptReady . . . 851
8.837.2.4 pGyroTempAcceptReady . . . 851
8.838 QmiCbkLocSetExtPowerConfigInd Struct Reference . . . 851
8.838.1 Detailed Description . . . 851
8.838.2 Field Documentation . . . 852
8.838.2.1 status . . . 852
8.839 QmiCbkLocSetServerInd Struct Reference . . . 852
8.839.1 Detailed Description . . . 852
8.839.2 Field Documentation . . . 853
8.839.2.1 serverStatus . . . 853
8.840 QmiCbkNasLTECphyCaInfo Struct Reference . . . 853
8.840.1 Detailed Description . . . 853
8.840.2 Field Documentation . . . 853
8.840.2.1 sPhyCaAggPcellInfo . . . 853
8.840.2.2 sPhyCaAggScellDlBw . . . 853
8.840.2.3 sPhyCaAggScellIndex . . . 853
8.840.2.4 sPhyCaAggScellIndType . . . 853
8.840.2.5 sPhyCaAggScellInfo . . . 853
8.841 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference . . . 853
8.841.1 Field Documentation . . . 854
8.841.1.1 SITlv . . . 854
8.842 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference . . . 854
8.842.1 Field Documentation . . . 854
8.842.1.1 SITlv . . . 854
8.843 QmiCbkTmdMitiLvlRptInd Struct Reference . . . 854
8.843.1 Detailed Description . . . 854
8.843.2 Field Documentation . . . 854
8.843.2.1 currentMitigationLvl . . . 854
8.843.2.2 MitigationDevInfo . . . 854
8.844 QmiCbkWdsStatisticsIndState Struct Reference . . . 854
8.844.1 Detailed Description . . . 855
8.844.2 Field Documentation . . . 855

8.844.2.1 RxDropConutTlv . . . 855
8.844.2.2 RxOkByteCountTlv . . . 855
8.844.2.3 RxOkConutTlv . . . 855
8.844.2.4 TxDropConutTlv . . . 855
8.844.2.5 TxOkByteCountTlv . . . 855
8.844.2.6 TxOkConutTlv . . . 855
8.845 qmifwinfo_s Struct Reference . . . 855
8.845.1 Detailed Description . . . 855
8.845.2 Field Documentation . . . 856
8.845.2.1 dev . . . 856
8.845.2.2 g . . . 856
8.845.2.3 s . . . 856
8.846 QmiNas3GppNetworkInfo Struct Reference . . . 856
8.846.1 Detailed Description . . . 856
8.846.2 Field Documentation . . . 857
8.846.2.1 pDesription . . . 857
8.846.2.2 pForbidden . . . 857
8.846.2.3 pInUse . . . 857
8.846.2.4 pMCC . . . 857
8.846.2.5 pMNC . . . 857
8.846.2.6 pPreferred . . . 857
8.846.2.7 pRoaming . . . 857
8.847 QmiNasGetRFBandInfoResp Struct Reference . . . 857
8.847.1 Field Documentation . . . 858
8.847.1.1 pInstancesSize . . . 858
8.847.1.2 pLTEOperationMode . . . 858
8.847.1.3 pRFBandInfoElements . . . 858
8.847.1.4 pRfBandInfoExtFormat . . . 858
8.847.1.5 pRfBandwidthInfo . . . 858
8.847.1.6 pRfDedicatedBandInfo . . . 858
8.847.1.7 results . . . 858
8.848 QmiNasPerformNetworkScanResp Struct Reference . . . 858
8.848.1 Field Documentation . . . 858
8.848.1.1 pInstances . . . 858
8.848.1.2 pInstanceSize . . . 858
8.848.1.3 results . . . 858
8.849 qmiSmsMessageList Struct Reference . . . 858
8.849.1 Detailed Description . . . 858
8.849.2 Field Documentation . . . 859
8.849.2.1 messageIndex . . . 859

8.849.2.2 messageTag . . . 859
8.850 qmiWDSDataBearerTechnology Struct Reference . . . 859
8.850.1 Detailed Description . . . 859
8.850.2 Field Documentation . . . 860
8.850.2.1 currentNetwork . . . 860
8.850.2.2 ratMask . . . 860
8.850.2.3 soMask . . . 860
8.851 QmiWdsIpAddressInfo Struct Reference . . . 860
8.851.1 Detailed Description . . . 860
8.851.2 Field Documentation . . . 861
8.851.2.1 pIPAddressV4 . . . 861
8.851.2.2 pIPAddressV6 . . . 861
8.851.2.3 pIPv6prefixlen . . . 861
8.852 qmiWdsRunTimeSettings Struct Reference . . . 861
8.852.1 Detailed Description . . . 862
8.852.2 Field Documentation . . . 864
8.852.2.1 pAPNName . . . 864
8.852.2.2 pAuthentication . . . 864
8.852.2.3 pDomainList . . . 864
8.852.2.4 pGPRSGrantedQoS . . . 864
8.852.2.5 pGWAddressV4 . . . 864
8.852.2.6 pIMCNflag . . . 864
8.852.2.7 pIPAddressV4 . . . 864
8.852.2.8 pIPFamilyPreference . . . 864
8.852.2.9 pIPV6AddrInfo . . . 864
8.852.2.10 pIPV6GWAddrInfo . . . 864
8.852.2.11 pMtu . . . 864
8.852.2.12 pPCSCFAddrPCO . . . 864
8.852.2.13 pPCSCFFQDNAddrList . . . 864
8.852.2.14 pPDPType . . . 864
8.852.2.15 pPrimaryDNSV4 . . . 864
8.852.2.16 pPrimaryDNSV6 . . . 864
8.852.2.17 pProfileID . . . 864
8.852.2.18 pProfileName . . . 865
8.852.2.19 pSecondaryDNSV4 . . . 865
8.852.2.20 pSecondaryDNSV6 . . . 865
8.852.2.21 pServerAddrList . . . 865
8.852.2.22 pSubnetMaskV4 . . . 865
8.852.2.23 pTechnology . . . 865
8.852.2.24 pUMTSGrantedQoS . . . 865

8.852.2.25 pUsername . . . 865
8.853 qos_BindDataPortMuxID_t Struct Reference . . . 865
8.853.1 Detailed Description . . . 865
8.853.2 Field Documentation . . . 865
8.853.2.1 MuxID . . . 865
8.854 qos_BindDataPortPeripheralEndPointID_t Struct Reference . . . 865
8.854.1 Detailed Description . . . 865
8.854.2 Field Documentation . . . 866
8.854.2.1 EndPointType . . . 866
8.854.2.2 IfaceID . . . 866
8.855 qos_BindDataPortSIODataPort_t Struct Reference . . . 866
8.855.1 Detailed Description . . . 866
8.855.2 Field Documentation . . . 866
8.855.2.1 SIODataPort . . . 866
8.856 QosClassID Struct Reference . . . 866
8.856.1 Detailed Description . . . 866
8.856.2 Field Documentation . . . 867
8.856.2.1 gDlBitRate . . . 867
8.856.2.2 gUlBitRate . . . 867
8.856.2.3 maxDlBitRate . . . 867
8.856.2.4 maxUlBitRate . . . 867
8.856.2.5 QCI . . . 867
8.857 QosEventInfo Struct Reference . . . 867
8.857.1 Detailed Description . . . 867
8.857.2 Field Documentation . . . 868
8.857.2.1 pDataBearer . . . 868
8.857.2.2 pPacketsCountRX . . . 868
8.857.2.3 pPacketsCountTX . . . 868
8.857.2.4 pTotalBytesRX . . . 868
8.857.2.5 pTotalBytesTX . . . 869
8.858 QosFlowInfo Struct Reference . . . 869
8.858.1 Detailed Description . . . 869
8.858.2 Field Documentation . . . 869
8.858.2.1 pBearerID . . . 869
8.858.2.2 pQFlowState . . . 869
8.858.2.3 pRxQFilter . . . 869
8.858.2.4 pRxQFlowGranted . . . 869
8.858.2.5 pTxQFilter . . . 869
8.858.2.6 pTxQFlowGranted . . . 870
8.859 QosFlowInfoState Struct Reference . . . 870

8.859.1 Detailed Description . . . 870
8.859.2 Field Documentation . . . 870
8.859.2.1 id . . . 870
8.859.2.2 isNewFlow . . . 870
8.859.2.3 state . . . 870
8.860 QosMap Struct Reference . . . 870
8.860.1 Detailed Description . . . 870
8.860.2 Field Documentation . . . 871
8.860.2.1 dscp . . . 871
8.860.2.2 qos_id . . . 871
8.860.2.3 state . . . 871
8.861 RankIndicatorInd Struct Reference . . . 871
8.861.1 Field Documentation . . . 871
8.861.1.1 Count1 . . . 871
8.861.1.2 Count2 . . . 871
8.862 readResult Struct Reference . . . 871
8.862.1 Detailed Description . . . 871
8.862.2 Field Documentation . . . 872
8.862.2.1 content . . . 872
8.862.2.2 contentLen . . . 872
8.863 readTransparentInfo Struct Reference . . . 872
8.863.1 Detailed Description . . . 872
8.863.2 Field Documentation . . . 872
8.863.2.1 length . . . 872
8.863.2.2 offset . . . 872
8.864 redirNumInfo Struct Reference . . . 872
8.864.1 Detailed Description . . . 872
8.864.2 Field Documentation . . . 874
8.864.2.1 number . . . 874
8.864.2.2 numLen . . . 874
8.864.2.3 numPlan . . . 874
8.864.2.4 numType . . . 874
8.864.2.5 PI . . . 874
8.864.2.6 reason . . . 874
8.864.2.7 SI . . . 874
8.865 registerRefresh Struct Reference . . . 874
8.865.1 Detailed Description . . . 874
8.865.2 Field Documentation . . . 875
8.865.2.1 arrfileInfo . . . 875
8.865.2.2 numFiles . . . 875

8.865.2.3 registerFlag . . . 875
8.865.2.4 voteForInit . . . 875
8.866 remainingRetries Struct Reference . . . 875
8.866.1 Detailed Description . . . 875
8.866.2 Field Documentation . . . 875
8.866.2.1 unblockLeft . . . 875
8.866.2.2 verifyLeft . . . 875
8.867 remotePartyName Struct Reference . . . 875
8.867.1 Detailed Description . . . 876
8.867.2 Field Documentation . . . 876
8.867.2.1 callerName . . . 876
8.867.2.2 codingScheme . . . 876
8.867.2.3 nameLen . . . 876
8.867.2.4 namePI . . . 876
8.868 remotePartyNum Struct Reference . . . 876
8.868.1 Detailed Description . . . 876
8.868.2 Field Documentation . . . 877
8.868.2.1 numLen . . . 877
8.868.2.2 presentationInd . . . 877
8.868.2.3 remPartyNumber . . . 877
8.869 ReqFieldsList Struct Reference . . . 877
8.869.1 Detailed Description . . . 877
8.869.2 Field Documentation . . . 877
8.869.2.1 requestFields . . . 877
8.869.2.2 requestFieldsLen . . . 878
8.870 RespFieldsList Struct Reference . . . 878
8.870.1 Detailed Description . . . 878
8.870.2 Field Documentation . . . 878
8.870.2.1 responseFields . . . 878
8.870.2.2 responseFieldsLen . . . 878
8.871 RFBandInfoElements Struct Reference . . . 878
8.871.1 Detailed Description . . . 878
8.871.2 Field Documentation . . . 879
8.871.2.1 activeBandClass . . . 879
8.871.2.2 activeBandClass . . . 879
8.871.2.3 activeChannel . . . 879
8.871.2.4 activeChannel . . . 879
8.871.2.5 radioInterface . . . 879
8.871.2.6 radioInterface . . . 879
8.872 RfBandInfoExtFormat Struct Reference . . . 879

8.872.1 Detailed Description . . . 879
8.872.2 Field Documentation . . . 880
8.872.2.1 pInstancesSize . . . 880
8.872.2.2 pRfBandInfoExtFormatParam . . . 880
8.872.2.3 TlvPresent . . . 880
8.873 RfBandInfoExtFormatElements Struct Reference . . . 880
8.873.1 Detailed Description . . . 880
8.873.2 Field Documentation . . . 880
8.873.2.1 activeBand . . . 880
8.873.2.2 activeChannel . . . 880
8.873.2.3 radioInterface . . . 880
8.874 RfBandInfoList Struct Reference . . . 881
8.874.1 Detailed Description . . . 881
8.874.2 Field Documentation . . . 881
8.874.2.1 pInstanceSize . . . 881
8.874.2.2 pRFBandInfoParam . . . 881
8.875 RfBandwidthInfo Struct Reference . . . 881
8.875.1 Detailed Description . . . 881
8.875.2 Field Documentation . . . 882
8.875.2.1 pInstancesSize . . . 882
8.875.2.2 pRfBandwidthInfoParam . . . 882
8.875.2.3 TlvPresent . . . 882
8.876 RfBandwidthInfoElements Struct Reference . . . 882
8.876.1 Detailed Description . . . 882
8.876.2 Field Documentation . . . 882
8.876.2.1 bandwidth . . . 882
8.876.2.2 radioInterface . . . 882
8.877 RfDedicatedBandInfo Struct Reference . . . 882
8.877.1 Detailed Description . . . 883
8.877.2 Field Documentation . . . 883
8.877.2.1 pInstancesSize . . . 883
8.877.2.2 pRfDedicatedBandInfoParam . . . 883
8.877.2.3 TlvPresent . . . 883
8.878 RfDedicatedBandInfoElements Struct Reference . . . 883
8.878.1 Detailed Description . . . 883
8.878.2 Field Documentation . . . 884
8.878.2.1 dedicatedBand . . . 884
8.878.2.2 radioInterface . . . 884
8.879 rmTrasnferStaticsReq Struct Reference . . . 884
8.879.1 Detailed Description . . . 884

8.879.2 Field Documentation . . . 884
8.879.2.1 bResetStatistics . . . 884
8.879.2.2 ulMask . . . 884
8.880 roamIndList Struct Reference . . . 884
8.880.1 Detailed Description . . . 885
8.880.2 Field Documentation . . . 885
8.880.2.1 numInstances . . . 885
8.880.2.2 radioInterface . . . 885
8.880.2.3 roamIndicator . . . 885
8.881 RoamingInfo Struct Reference . . . 885
8.881.1 Field Documentation . . . 885
8.881.1.1 roaming_ind . . . 885
8.881.1.2 TlvPresent . . . 885
8.882 roamTimer Struct Reference . . . 885
8.882.1 Detailed Description . . . 886
8.882.2 Field Documentation . . . 886
8.882.2.1 namID . . . 886
8.882.2.2 roamTimerValue . . . 886
8.883 RSRPThresh Struct Reference . . . 886
8.883.1 Detailed Description . . . 886
8.883.2 Field Documentation . . . 887
8.883.2.1 pRSRPThresList . . . 887
8.883.2.2 RSRPThresListLen . . . 887
8.884 rsrqInformation Struct Reference . . . 887
8.884.1 Detailed Description . . . 887
8.884.2 Field Documentation . . . 887
8.884.2.1 radioIf . . . 887
8.884.2.2 rsrq . . . 887
8.885 RSRQThresh Struct Reference . . . 887
8.885.1 Detailed Description . . . 888
8.885.2 Field Documentation . . . 888
8.885.2.1 pRSRQThresList . . . 888
8.885.2.2 RSRQThresListLen . . . 888
8.886 RSSIThresh Struct Reference . . . 888
8.886.1 Detailed Description . . . 888
8.886.2 Field Documentation . . . 889
8.886.2.1 pRSSIThresList . . . 889
8.886.2.2 RSSIThresListLen . . . 889
8.887 RXAGCList Struct Reference . . . 889
8.887.1 Detailed Description . . . 889

8.887.2 Field Documentation . . . 890
8.887.2.1 pRXAIG . . . 890
8.887.2.2 pRXComprSlope . . . 890
8.887.2.3 pRXComprThres . . . 890
8.887.2.4 pRXExpSlope . . . 890
8.887.2.5 pRXExpThres . . . 890
8.887.2.6 pRXStaticGain . . . 890
8.888 RXAVCList Struct Reference . . . 890
8.888.1 Detailed Description . . . 890
8.888.2 Field Documentation . . . 890
8.888.2.1 pAVRXAVCHeadroom . . . 890
8.888.2.2 pAVRXAVCSens . . . 890
8.889 rxInfo Struct Reference . . . 890
8.889.1 Detailed Description . . . 891
8.889.2 Field Documentation . . . 891
8.889.2.1 ecio . . . 891
8.889.2.2 isRadioTuned . . . 891
8.889.2.3 phase . . . 891
8.889.2.4 rscp . . . 891
8.889.2.5 rsrp . . . 891
8.889.2.6 rxPower . . . 891
8.890 RXPCMIIRFltr Struct Reference . . . 892
8.890.1 Detailed Description . . . 892
8.890.2 Field Documentation . . . 893
8.890.2.1 pFlag . . . 893
8.890.2.2 pStage0Val . . . 893
8.890.2.3 pStage1Val . . . 893
8.890.2.4 pStage2Val . . . 893
8.890.2.5 pStage3Val . . . 893
8.890.2.6 pStage4Val . . . 893
8.890.2.7 pStageCnt . . . 893
8.891 RxSigInfo Struct Reference . . . 893
8.891.1 Detailed Description . . . 893
8.891.2 Field Documentation . . . 894
8.891.2.1 isRadioTuned . . . 894
8.891.2.2 rsrp . . . 894
8.891.2.3 rxChainIndex . . . 894
8.891.2.4 rxPower . . . 894
8.892 rxSignalStrengthListElement Struct Reference . . . 894
8.892.1 Detailed Description . . . 894

8.892.2 Field Documentation . . . 895
8.892.2.1 radioIf . . . 895
8.892.2.2 rxSignalStrength . . . 895
8.893 sApnExtraParams Struct Reference . . . 895
8.893.1 Detailed Description . . . 895
8.893.2 Field Documentation . . . 896
8.893.2.1 ambr_dl . . . 896
8.893.2.2 ambr_dl_ext . . . 896
8.893.2.3 ambr_dl_ext2 . . . 896
8.893.2.4 ambr_ul . . . 896
8.893.2.5 ambr_ul_ext . . . 896
8.893.2.6 ambr_ul_ext2 . . . 896
8.893.2.7 apnId . . . 896
8.894 satelliteInfo Struct Reference . . . 896
8.894.1 Detailed Description . . . 897
8.894.2 Field Documentation . . . 898
8.894.2.1 azimuth . . . 898
8.894.2.2 elevation . . . 898
8.894.2.3 gnssSvId . . . 898
8.894.2.4 healthStatus . . . 898
8.894.2.5 snr . . . 898
8.894.2.6 svInfoMask . . . 898
8.894.2.7 svListLen . . . 898
8.894.2.8 svStatus . . . 898
8.894.2.9 system . . . 898
8.894.2.10 validMask . . . 898
8.895 SccRxInfo Struct Reference . . . 899
8.895.1 Detailed Description . . . 899
8.895.2 Field Documentation . . . 899
8.895.2.1 numInstances . . . 899
8.895.2.2 rsrq . . . 899
8.895.2.3 sigInfo . . . 899
8.895.2.4 snr . . . 899
8.895.2.5 TlvPresent . . . 899
8.896 sensorData Struct Reference . . . 899
8.896.1 Detailed Description . . . 900
8.896.2 Field Documentation . . . 901
8.896.2.1 flags . . . 901
8.896.2.2 sensorDataLen . . . 901
8.896.2.3 timeOfFirstSample . . . 901

8.896.2.4 timeOffset . . . 901
8.896.2.5 xAxis . . . 901
8.896.2.6 yAxis . . . 901
8.896.2.7 zAxis . . . 901
8.897 sensorData_t Struct Reference . . . 901
8.897.1 Detailed Description . . . 901
8.897.2 Field Documentation . . . 902
8.897.2.1 flags . . . 902
8.897.2.2 sensorDataLen . . . 902
8.897.2.3 timeOfFirstSample . . . 902
8.897.2.4 timeOffset . . . 902
8.897.2.5 xAxis . . . 902
8.897.2.6 yAxis . . . 902
8.897.2.7 zAxis . . . 902
8.898 sensorDataUsage_s Struct Reference . . . 903
8.898.1 Detailed Description . . . 903
8.898.2 Field Documentation . . . 903
8.898.2.1 aidingIndicatorMask . . . 903
8.898.2.2 usageMask . . . 903
8.899 serialNumbersInfo Struct Reference . . . 903
8.899.1 Detailed Description . . . 904
8.899.2 Field Documentation . . . 904
8.899.2.1 esnSize . . . 904
8.899.2.2 imeiSize . . . 904
8.899.2.3 imeiSvnSize . . . 904
8.899.2.4 meidSize . . . 904
8.899.2.5 pESNString . . . 904
8.899.2.6 pIMEIString . . . 904
8.899.2.7 pImeiSvnString . . . 904
8.899.2.8 pMEIDString . . . 904
8.900 serviceProviderName Struct Reference . . . 905
8.900.1 Detailed Description . . . 905
8.900.2 Field Documentation . . . 905
8.900.2.1 displayCondition . . . 905
8.900.2.2 spn . . . 905
8.900.2.3 spnLength . . . 905
8.901 ServingSystemInfo Struct Reference . . . 905
8.901.1 Detailed Description . . . 905
8.901.2 Field Documentation . . . 906
8.901.2.1 csAttachState . . . 906

8.901.2.2 hdrPersonality . . . 906
8.901.2.3 psAttachState . . . 906
8.901.2.4 radioInterfaceList . . . 907
8.901.2.5 radioInterfaceNo . . . 907
8.901.2.6 registrationState . . . 907
8.901.2.7 selectedNetwork . . . 907
8.902 servSystem Struct Reference . . . 907
8.902.1 Detailed Description . . . 907
8.902.2 Field Documentation . . . 908
8.902.2.1 csAttachState . . . 908
8.902.2.2 numRadioInterfaces . . . 908
8.902.2.3 psAttachState . . . 908
8.902.2.4 radioInterface . . . 908
8.902.2.5 regState . . . 908
8.902.2.6 selNetwork . . . 908
8.903 sessionInfo Union Reference . . . 908
8.903.1 Detailed Description . . . 908
8.903.2 Field Documentation . . . 909
8.903.2.1 omaDmConfig . . . 909
8.903.2.2 omaDmFota . . . 909
8.903.2.3 omaDmNotifications . . . 909
8.904 sessionInfoExt Union Reference . . . 909
8.904.1 Detailed Description . . . 909
8.904.2 Field Documentation . . . 909
8.904.2.1 omaDmConfig . . . 909
8.904.2.2 omaDmFota . . . 909
8.905 sessionInfoTlv Struct Reference . . . 909
8.905.1 Detailed Description . . . 909
8.905.2 Field Documentation . . . 909
8.905.2.1 sessionInfo . . . 909
8.905.2.2 sessionType . . . 909
8.905.2.3 TlvPresent . . . 909
8.906 sessionInfoTlvExt Struct Reference . . . 910
8.906.1 Detailed Description . . . 910
8.906.2 Field Documentation . . . 910
8.906.2.1 sessionInfo . . . 910
8.906.2.2 sessionType . . . 910
8.906.2.3 TlvPresent . . . 910
8.907 SetAudioPathConfigReq Struct Reference . . . 910
8.907.1 Detailed Description . . . 910

8.907.2 Field Documentation . . . 911
8.907.2.1 pCodecSTGain . . . 911
8.907.2.2 pDTMFTXGain . . . 912
8.907.2.3 pECMode . . . 912
8.907.2.4 pNSEnable . . . 912
8.907.2.5 Profile . . . 912
8.907.2.6 pRXAGCList . . . 912
8.907.2.7 pRXAVCAGCSwitch . . . 912
8.907.2.8 pRXAVCList . . . 912
8.907.2.9 pRXPCMIIRFltr . . . 912
8.907.2.10 pTXAGCList . . . 912
8.907.2.11 pTXAVCSwitch . . . 912
8.907.2.12 pTXGain . . . 912
8.907.2.13 pTXPCMIIRFltr . . . 912
8.908 SetAudioProfileReq Struct Reference . . . 912
8.908.1 Detailed Description . . . 912
8.908.2 Field Documentation . . . 913
8.908.2.1 EarMute . . . 913
8.908.2.2 Generator . . . 913
8.908.2.3 MicMute . . . 913
8.908.2.4 Profile . . . 913
8.908.2.5 Volume . . . 913
8.909 SetAudioVolTLBConfigReq Struct Reference . . . 913
8.909.1 Detailed Description . . . 913
8.909.2 Field Documentation . . . 914
8.909.2.1 Generator . . . 914
8.909.2.2 Item . . . 914
8.909.2.3 Profile . . . 914
8.909.2.4 Volume . . . 914
8.909.2.5 VolValue . . . 914
8.910 SetAudioVolTLBConfigResp Struct Reference . . . 914
8.910.1 Detailed Description . . . 914
8.910.2 Field Documentation . . . 914
8.910.2.1 ResCode . . . 914
8.911 setCustomSettingV2 Struct Reference . . . 915
8.911.1 Detailed Description . . . 915
8.911.2 Field Documentation . . . 915
8.911.2.1 cust_id . . . 915
8.911.2.2 cust_value . . . 915
8.911.2.3 value_length . . . 915

8.912 setDyingGaspCfg Struct Reference . . . 915
8.912.1 Detailed Description . . . 915
8.912.2 Field Documentation . . . 916
8.912.2.1 pDestSMSContent . . . 916
8.912.2.2 pDestSMSNum . . . 916
8.913 SetIMSSMSConfigReq Struct Reference . . . 916
8.913.1 Detailed Description . . . 916
8.913.2 Field Documentation . . . 916
8.913.2.1 pPhoneCtxtURI . . . 916
8.913.2.2 pPhoneCtxtURILen . . . 916
8.913.2.3 pSMSFormat . . . 916
8.913.2.4 pSMSOverIPNwInd . . . 916
8.914 SetIMSSMSConfigResp Struct Reference . . . 916
8.914.1 Detailed Description . . . 917
8.914.2 Field Documentation . . . 917
8.914.2.1 pSettingResp . . . 917
8.915 SetIMSUserConfigReq Struct Reference . . . 917
8.915.1 Detailed Description . . . 917
8.915.2 Field Documentation . . . 917
8.915.2.1 pIMSDomain . . . 917
8.915.2.2 pIMSDomainLen . . . 917
8.916 SetIMSUserConfigResp Struct Reference . . . 917
8.916.1 Detailed Description . . . 918
8.916.2 Field Documentation . . . 918
8.916.2.1 pSettingResp . . . 918
8.917 SetIMSVoIPConfigReq Struct Reference . . . 918
8.917.1 Detailed Description . . . 918
8.917.2 Field Documentation . . . 920
8.917.2.1 pAmrMode . . . 920
8.917.2.2 pAmrOctetAligned . . . 920
8.917.2.3 pAmrWbEnable . . . 920
8.917.2.4 pAmrWBMode . . . 920
8.917.2.5 pAmrWBOctetAligned . . . 920
8.917.2.6 pMinSessionExpiryTimer . . . 920
8.917.2.7 pRingBackTimer . . . 920
8.917.2.8 pRingingTimer . . . 920
8.917.2.9 pRTPRTCPInactTimer . . . 920
8.917.2.10 pScrAmrEnable . . . 920
8.917.2.11 pScrAmrWbEnable . . . 920
8.917.2.12 pSessionExpiryTimer . . . 920

8.918 SetIMSVoIPConfigResp Struct Reference . . . 920
8.918.1 Detailed Description . . . 920
8.918.2 Field Documentation . . . 921
8.918.2.1 pSettingResp . . . 921
8.919 SetM2MAudioAVCFGReq Struct Reference . . . 921
8.919.1 Detailed Description . . . 921
8.919.2 Field Documentation . . . 921
8.919.2.1 Device . . . 921
8.919.2.2 PIFACEId . . . 921
8.919.2.3 pPCMParams . . . 921
8.919.2.4 Profile . . . 921
8.920 SetM2MAudioLPBKReq Struct Reference . . . 922
8.920.1 Detailed Description . . . 922
8.920.2 Field Documentation . . . 922
8.920.2.1 Enable . . . 922
8.921 SetM2MAudioProfileReq Struct Reference . . . 922
8.921.1 Detailed Description . . . 922
8.921.2 Field Documentation . . . 923
8.921.2.1 pCwtMute . . . 923
8.921.2.2 pEarMute . . . 923
8.921.2.3 pGenerator . . . 923
8.921.2.4 pMicMute . . . 923
8.921.2.5 Profile . . . 923
8.921.2.6 pVolume . . . 923
8.922 SetM2MAudioVolumeReq Struct Reference . . . 923
8.922.1 Detailed Description . . . 923
8.922.2 Field Documentation . . . 924
8.922.2.1 Generator . . . 924
8.922.2.2 Level . . . 924
8.922.2.3 Profile . . . 924
8.923 SetM2MAVMuteReq Struct Reference . . . 924
8.923.1 Detailed Description . . . 924
8.923.2 Field Documentation . . . 924
8.923.2.1 EarMute . . . 924
8.923.2.2 MicMute . . . 924
8.923.2.3 pCwtMute . . . 925
8.923.2.4 Profile . . . 925
8.924 SetM2MSpkrGainReq Struct Reference . . . 925
8.924.1 Detailed Description . . . 925
8.924.2 Field Documentation . . . 925

8.924.2.1 Profile . . . 925
8.924.2.2 Value . . . 925
8.925 setPINProtection Struct Reference . . . 925
8.925.1 Detailed Description . . . 925
8.925.2 Field Documentation . . . 926
8.925.2.1 pinID . . . 926
8.925.2.2 pinLength . . . 926
8.925.2.3 pinOperation . . . 926
8.925.2.4 pinValue . . . 926
8.926 SetRegMgrConfigReq Struct Reference . . . 926
8.926.1 Detailed Description . . . 926
8.926.2 Field Documentation . . . 927
8.926.2.1 pCSCFPortName . . . 927
8.926.2.2 pCSCFPortNameLen . . . 927
8.926.2.3 pIMSTestMode . . . 927
8.926.2.4 pPriCSCFPort . . . 927
8.927 SetRegMgrConfigResp Struct Reference . . . 927
8.927.1 Detailed Description . . . 927
8.927.2 Field Documentation . . . 927
8.927.2.1 pSettingResp . . . 927
8.928 setSignalStrengthInfo Struct Reference . . . 927
8.928.1 Detailed Description . . . 928
8.928.2 Field Documentation . . . 931
8.928.2.1 pCDMAECIODelta . . . 931
8.928.2.2 pCDMAECIOThresh . . . 931
8.928.2.3 pCDMARSSIDelta . . . 931
8.928.2.4 pCDMARSSIThresh . . . 931
8.928.2.5 pGSMRSSIDelta . . . 931
8.928.2.6 pGSMRSSIThresh . . . 931
8.928.2.7 pHDRECIODelta . . . 931
8.928.2.8 pHDRECIOThresh . . . 931
8.928.2.9 pHDRIODelta . . . 931
8.928.2.10 pHDRIOThresh . . . 931
8.928.2.11 pHDRRSSIDelta . . . 931
8.928.2.12 pHDRRSSIThresh . . . 931
8.928.2.13 pHDRSINRDelta . . . 931
8.928.2.14 pHDRSINRThresh . . . 931
8.928.2.15 pLTERSRPDelta . . . 931
8.928.2.16 pLTERSRPThresh . . . 931
8.928.2.17 pLTERSRQDelta . . . 931

8.928.2.18 pLTERSRQThresh . . . 932
8.928.2.19 pLTERSSIDelta . . . 932
8.928.2.20 pLTERSSIThresh . . . 932
8.928.2.21 pLTESigRptConfig . . . 932
8.928.2.22 pLTESNRDelta . . . 932
8.928.2.23 pLTESNRThresh . . . 932
8.928.2.24 pTDSCDMAECIODelta . . . 932
8.928.2.25 pTDSCDMAECIOThresh . . . 932
8.928.2.26 pTDSCDMARSCPDelta . . . 932
8.928.2.27 pTDSCDMARSCPThresh . . . 932
8.928.2.28 pTDSCDMARSSIDelta . . . 932
8.928.2.29 pTDSCDMARSSIThresh . . . 932
8.928.2.30 pTDSCDMASINRDelta . . . 932
8.928.2.31 pTDSCDMASINRThresh . . . 932
8.928.2.32 pWCDMAECIODelta . . . 932
8.928.2.33 pWCDMAECIOThresh . . . 932
8.928.2.34 pWCDMARSSIDelta . . . 932
8.928.2.35 pWCDMARSSIThresh . . . 932
8.929 SetSIPConfigReq Struct Reference . . . 932
8.929.1 Detailed Description . . . 932
8.929.2 Field Documentation . . . 933
8.929.2.1 pSigCompEnabled . . . 933
8.929.2.2 pSIPLocalPort . . . 933
8.929.2.3 pSubscribeTimer . . . 933
8.929.2.4 pTimerSIPReg . . . 933
8.929.2.5 pTimerT1 . . . 933
8.929.2.6 pTimerT2 . . . 933
8.929.2.7 pTimerTf . . . 933
8.930 SetSIPConfigResp Struct Reference . . . 933
8.930.1 Detailed Description . . . 933
8.930.2 Field Documentation . . . 934
8.930.2.1 pSettingResp . . . 934
8.931 sGetDeviceSeriesResult Struct Reference . . . 934
8.931.1 Detailed Description . . . 934
8.931.2 Field Documentation . . . 934
8.931.2.1 eDevice . . . 934
8.931.2.2 uResult . . . 934
8.932 sidNid Struct Reference . . . 934
8.932.1 Detailed Description . . . 934
8.932.2 Field Documentation . . . 935

8.932.2.1 nid . . . 935
8.932.2.2 sid . . . 935
8.933 sigInfo Struct Reference . . . 935
8.933.1 Detailed Description . . . 935
8.933.2 Field Documentation . . . 936
8.933.2.1 pECIOThresh . . . 936
8.933.2.2 pHDRSINRThresh . . . 936
8.933.2.3 pIOThresh . . . 936
8.933.2.4 pLTESigRptCfg . . . 936
8.933.2.5 pLTESNRThresh . . . 936
8.933.2.6 pRSRPThresh . . . 936
8.933.2.7 pRSRQThresh . . . 936
8.933.2.8 pRSSIThresh . . . 936
8.933.2.9 pTDSCDMASINRCONFThresh . . . 936
8.934 signalInfo Struct Reference . . . 936
8.934.1 Detailed Description . . . 936
8.934.2 Field Documentation . . . 937
8.934.2.1 alertPitch . . . 937
8.934.2.2 signal . . . 937
8.934.2.3 signalType . . . 937
8.935 SignalStrengthDataType Struct Reference . . . 937
8.935.1 Field Documentation . . . 937
8.935.1.1 thresholds . . . 937
8.935.1.2 thresholdsSize . . . 937
8.936 slot_t Struct Reference . . . 937
8.936.1 Detailed Description . . . 937
8.936.2 Field Documentation . . . 938
8.936.2.1 bICCID . . . 938
8.936.2.2 bICCIDLength . . . 938
8.936.2.3 bLogicalSlot . . . 938
8.936.2.4 uPhyCardStatus . . . 938
8.936.2.5 uPhySlotStatus . . . 938
8.937 slotInf Struct Reference . . . 938
8.937.1 Detailed Description . . . 938
8.937.2 Field Documentation . . . 939
8.937.2.1 AppStatus . . . 939
8.937.2.2 cardState . . . 939
8.937.2.3 errorState . . . 939
8.937.2.4 numApp . . . 939
8.937.2.5 upinRetries . . . 940

8.937.2.6 upinState . . . 940
8.937.2.7 upukRetries . . . 940
8.938 slotInfo Struct Reference . . . 940
8.938.1 Detailed Description . . . 940
8.938.2 Field Documentation . . . 941
8.938.2.1 AppStatus . . . 941
8.938.2.2 cardState . . . 941
8.938.2.3 errorState . . . 941
8.938.2.4 numApp . . . 941
8.938.2.5 upinRetries . . . 941
8.938.2.6 upinState . . . 941
8.938.2.7 upukRetries . . . 941
8.939 slots_t Struct Reference . . . 941
8.939.1 Detailed Description . . . 941
8.939.2 Field Documentation . . . 942
8.939.2.1 uimSlotStatus . . . 942
8.940 slqsautoconnect Struct Reference . . . 942
8.940.1 Detailed Description . . . 942
8.940.2 Field Documentation . . . 942
8.940.2.1 acroamsetting . . . 942
8.940.2.2 acsetting . . . 942
8.940.2.3 action . . . 942
8.941 SLQSDeleteProfileParams Struct Reference . . . 942
8.941.1 Detailed Description . . . 943
8.941.2 Field Documentation . . . 943
8.941.2.1 profileIndex . . . 943
8.941.2.2 profileType . . . 943
8.942 slqsfwinfo_s Struct Reference . . . 943
8.942.1 Detailed Description . . . 943
8.942.2 Field Documentation . . . 944
8.942.2.1 appversion_str . . . 944
8.942.2.2 bootversion_str . . . 944
8.942.2.3 carrier_str . . . 944
8.942.2.4 cur_carr_name . . . 944
8.942.2.5 cur_carr_rev . . . 944
8.942.2.6 modelid_str . . . 944
8.942.2.7 packageid_str . . . 944
8.942.2.8 priversion_str . . . 944
8.942.2.9 sku_str . . . 944
8.943 SlqsNas3GppNetworkInfo Struct Reference . . . 944

8.943.1 Detailed Description . . . 945
8.943.2 Field Documentation . . . 945
8.943.2.1 Description . . . 945
8.943.2.2 Forbidden . . . 945
8.943.2.3 InUse . . . 945
8.943.2.4 MCC . . . 945
8.943.2.5 MNC . . . 945
8.943.2.6 Preferred . . . 946
8.943.2.7 Roaming . . . 946
8.944 SlqsNasPcsDigit Struct Reference . . . 946
8.944.1 Detailed Description . . . 946
8.944.2 Field Documentation . . . 946
8.944.2.1 includes_pcs_digit . . . 946
8.944.2.2 MCC . . . 946
8.944.2.3 MNC . . . 946
8.945 slqssendasyncsmsparams_s Struct Reference . . . 946
8.945.1 Detailed Description . . . 947
8.945.2 Field Documentation . . . 948
8.945.2.1 messageFormat . . . 948
8.945.2.2 messageSize . . . 948
8.945.2.3 pFollowOnDC . . . 948
8.945.2.4 pForceOnDC . . . 948
8.945.2.5 pLinktimer . . . 948
8.945.2.6 pMessage . . . 948
8.945.2.7 pRetryMessage . . . 948
8.945.2.8 pRetryMessageId . . . 948
8.945.2.9 pServiceOption . . . 948
8.945.2.10 pSmsOnIms . . . 948
8.945.2.11 pUserData . . . 948
8.946 slqssendsmsparams_s Struct Reference . . . 948
8.946.1 Detailed Description . . . 949
8.946.2 Field Documentation . . . 949
8.946.2.1 messageFailureCode . . . 949
8.946.2.2 messageFormat . . . 949
8.946.2.3 messageID . . . 949
8.946.2.4 messageSize . . . 949
8.946.2.5 pLinktimer . . . 949
8.946.2.6 pMessage . . . 949
8.946.2.7 pSmsOnIms . . . 949
8.947 slqsSessionStateInfo Struct Reference . . . 950

8.947.1 Detailed Description . . . 950
8.947.2 Field Documentation . . . 950
8.947.2.1 pQmiInterfaceInfo . . . 950
8.947.2.2 reconfiguration_required . . . 950
8.947.2.3 sessionEndReason . . . 950
8.947.2.4 state . . . 950
8.948 slqsSignalStrengthInfo Struct Reference . . . 950
8.948.1 Detailed Description . . . 951
8.948.2 Field Documentation . . . 952
8.948.2.1 ecioList . . . 952
8.948.2.2 ecioListLen . . . 952
8.948.2.3 errorRateList . . . 952
8.948.2.4 errorRateListLen . . . 952
8.948.2.5 Io . . . 952
8.948.2.6 ltersrp . . . 952
8.948.2.7 ltesnr . . . 952
8.948.2.8 rsrqInfo . . . 952
8.948.2.9 rxSignalStrengthList . . . 952
8.948.2.10 rxSignalStrengthListLen . . . 952
8.948.2.11 signalStrengthReqMask . . . 953
8.948.2.12 sinr . . . 953
8.949 SLQSSignalStrengthsIndReq Struct Reference . . . 953
8.949.1 Detailed Description . . . 953
8.949.2 Field Documentation . . . 954
8.949.2.1 ecioDelta . . . 954
8.949.2.2 ecioThresholdList . . . 954
8.949.2.3 ecioThresholdListLen . . . 954
8.949.2.4 ioDelta . . . 954
8.949.2.5 lteRsrpDelta . . . 954
8.949.2.6 lteSnrDelta . . . 954
8.949.2.7 rsrqDelta . . . 954
8.949.2.8 rxSignalStrengthDelta . . . 954
8.949.2.9 sinrDelta . . . 954
8.949.2.10 sinrThresholdList . . . 954
8.949.2.11 sinrThresholdListLen . . . 954
8.950 SLQSSignalStrengthsInformation Struct Reference . . . 955
8.950.1 Detailed Description . . . 955
8.950.2 Field Documentation . . . 955
8.950.2.1 ecioInfo . . . 956
8.950.2.2 errorRateInfo . . . 956

8.950.2.3 io . . . 956
8.950.2.4 lteRsrpinfo . . . 956
8.950.2.5 lteSnrinfo . . . 956
8.950.2.6 rsrqInfo . . . 956
8.950.2.7 rxSignalStrengthInfo . . . 956
8.950.2.8 sinr . . . 956
8.951 slqsWdsEventInfo Struct Reference . . . 956
8.951.1 Detailed Description . . . 956
8.951.2 Field Documentation . . . 957
8.951.2.1 pDataBearer . . . 957
8.951.2.2 pDormancyStatus . . . 957
8.951.2.3 pPacketsCountRX . . . 957
8.951.2.4 pPacketsCountTX . . . 957
8.951.2.5 pQmiInterfaceInfo . . . 958
8.951.2.6 pTotalBytesRX . . . 958
8.951.2.7 pTotalBytesTX . . . 958
8.952 sms_BroadcastConfig Struct Reference . . . 958
8.952.1 Detailed Description . . . 958
8.952.2 Field Documentation . . . 958
8.952.2.1 fromServiceId . . . 958
8.952.2.2 selected . . . 958
8.952.2.3 toServiceId . . . 958
8.953 sms_CDMABroadcastConfig Struct Reference . . . 958
8.953.1 Detailed Description . . . 958
8.953.2 Field Documentation . . . 959
8.953.2.1 language . . . 959
8.953.2.2 selected . . . 959
8.953.2.3 serviceCategory . . . 959
8.954 sms_getIndicationReg Struct Reference . . . 959
8.954.1 Detailed Description . . . 959
8.954.2 Field Documentation . . . 960
8.954.2.1 pRegCallStatInfoEvt . . . 960
8.954.2.2 pRegTransLayerInfoEvt . . . 960
8.954.2.3 pRegTransNWRegInfoEvt . . . 960
8.955 sms_getMsgWaitingInfo Struct Reference . . . 960
8.955.1 Detailed Description . . . 960
8.955.2 Field Documentation . . . 960
8.955.2.1 msgWaitInfo . . . 960
8.955.2.2 numInstances . . . 960
8.956 sms_getTransLayerInfo Struct Reference . . . 960

8.956.1 Detailed Description . . . 961
8.956.2 Field Documentation . . . 961
8.956.2.1 pRegInd . . . 961
8.956.2.2 pTransLayerInfo . . . 961
8.957 sms_getTransNWRegInfo Struct Reference . . . 961
8.957.1 Detailed Description . . . 961
8.957.2 Field Documentation . . . 962
8.957.2.1 pRegStatus . . . 962
8.958 sms_maxStorageSizeReq Struct Reference . . . 962
8.958.1 Detailed Description . . . 962
8.958.2 Field Documentation . . . 962
8.958.2.1 pMessageMode . . . 962
8.958.2.2 storageType . . . 962
8.959 sms_maxStorageSizeResp Struct Reference . . . 962
8.959.1 Detailed Description . . . 962
8.959.2 Field Documentation . . . 963
8.959.2.1 freeSlots . . . 963
8.959.2.2 maxStorageSize . . . 963
8.960 sms_messageWaitingInfoContent Struct Reference . . . 963
8.960.1 Detailed Description . . . 963
8.960.2 Field Documentation . . . 963
8.960.2.1 activeInd . . . 963
8.960.2.2 msgCount . . . 963
8.960.2.3 msgType . . . 963
8.961 sms_msgProtocolResp Struct Reference . . . 964
8.961.1 Detailed Description . . . 964
8.961.2 Field Documentation . . . 964
8.961.2.1 msgProtocol . . . 964
8.962 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . 964
8.962.1 Detailed Description . . . 964
8.962.2 Field Documentation . . . 965
8.962.2.1 activated_ind . . . 965
8.962.2.2 CDMABroadcastConfig . . . 965
8.962.2.3 num_instances . . . 965
8.963 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference . . . 965
8.963.1 Detailed Description . . . 965
8.963.2 Field Documentation . . . 965
8.963.2.1 activated_ind . . . 965
8.963.2.2 broadcastConfig . . . 965
8.963.2.3 num_instances . . . 965

8.964 sms_routeEntry Struct Reference . . . 965
8.964.1 Detailed Description . . . 966
8.964.2 Field Documentation . . . 966
8.964.2.1 messageClass . . . 966
8.964.2.2 messageType . . . 966
8.964.2.3 receiptAction . . . 966
8.964.2.4 routeStorage . . . 966
8.965 sms_sendAsyncsmsParams Struct Reference . . . 966
8.965.1 Detailed Description . . . 967
8.965.2 Field Documentation . . . 968
8.965.2.1 messageFormat . . . 968
8.965.2.2 messageSize . . . 968
8.965.2.3 pFollowOnDC . . . 968
8.965.2.4 pForceOnDC . . . 968
8.965.2.5 pLinktimer . . . 968
8.965.2.6 pMessage . . . 968
8.965.2.7 pRetryMessage . . . 968
8.965.2.8 pRetryMessageId . . . 968
8.965.2.9 pServiceOption . . . 968
8.965.2.10 pSmsOnIms . . . 968
8.965.2.11 pUserData . . . 968
8.966 sms_setIndicationReg Struct Reference . . . 969
8.966.1 Detailed Description . . . 969
8.966.2 Field Documentation . . . 969
8.966.2.1 pRegCallStatInfoEvt . . . 969
8.966.2.2 pRegTransLayerInfoEvt . . . 969
8.966.2.3 pRegTransNWRegInfoEvt . . . 969
8.967 sms_setRoutesReq Struct Reference . . . 969
8.967.1 Detailed Description . . . 970
8.967.2 Field Documentation . . . 970
8.967.2.1 numOfRoutes . . . 970
8.967.2.2 pTransferStatusReport . . . 970
8.967.2.3 routeList . . . 970
8.968 sms_transLayerInfo Struct Reference . . . 970
8.968.1 Detailed Description . . . 970
8.968.2 Field Documentation . . . 970
8.968.2.1 TransCap . . . 971
8.968.2.2 TransType . . . 971
8.969 SMSAsyncRawSend_s Struct Reference . . . 971
8.969.1 Detailed Description . . . 971

8.969.2 Field Documentation . . . 972
8.969.2.1 alphaIDLen . . . 972
8.969.2.2 causeCode . . . 972
8.969.2.3 errorClass . . . 972
8.969.2.4 messageID . . . 972
8.969.2.5 msgDelFailureCause . . . 972
8.969.2.6 msgDelFailureType . . . 972
8.969.2.7 pAlphaID . . . 972
8.969.2.8 RPCause . . . 972
8.969.2.9 sendStatus . . . 972
8.969.2.10 TPCause . . . 972
8.969.2.11 userData . . . 972
8.970 SMSCAddress Struct Reference . . . 972
8.970.1 Detailed Description . . . 973
8.970.2 Field Documentation . . . 973
8.970.2.1 data . . . 973
8.970.2.2 length . . . 973
8.971 sMSCAddressInfo Struct Reference . . . 973
8.971.1 Detailed Description . . . 973
8.971.2 Field Documentation . . . 973
8.971.2.1 data . . . 973
8.971.2.2 length . . . 973
8.972 sMSCAddressTlv Struct Reference . . . 973
8.972.1 Detailed Description . . . 974
8.972.2 Field Documentation . . . 974
8.972.2.1 SMSCInfo . . . 974
8.972.2.2 TlvPresent . . . 974
8.973 SMSEtwsMessage Struct Reference . . . 974
8.973.1 Detailed Description . . . 974
8.973.2 Field Documentation . . . 974
8.973.2.1 data . . . 974
8.973.2.2 length . . . 974
8.973.2.3 notificationType . . . 974
8.974 sMSEtwsMessageInfo Struct Reference . . . 975
8.974.1 Detailed Description . . . 975
8.974.2 Field Documentation . . . 975
8.974.2.1 data . . . 975
8.974.2.2 length . . . 975
8.974.2.3 notificationType . . . 975
8.975 sMSEtwsMessageTlv Struct Reference . . . 975

8.975.1 Detailed Description . . . 975
8.975.2 Field Documentation . . . 976
8.975.2.1 EtwsMessageInfo . . . 976
8.975.2.2 TlvPresent . . . 976
8.976 SMSEtwsPlmn Struct Reference . . . 976
8.976.1 Detailed Description . . . 976
8.976.2 Field Documentation . . . 976
8.976.2.1 mobileCountryCode . . . 976
8.976.2.2 mobileNetworkCode . . . 976
8.977 sMSEtwsPlmnInfo Struct Reference . . . 976
8.977.1 Detailed Description . . . 976
8.977.2 Field Documentation . . . 977
8.977.2.1 mobileCountryCode . . . 977
8.977.2.2 mobileNetworkCode . . . 977
8.978 SMSEventInfo_s Struct Reference . . . 977
8.978.1 Detailed Description . . . 977
8.978.2 Field Documentation . . . 978
8.978.2.1 pEtwsMessageInfo . . . 978
8.978.2.2 pEtwsPlmnInfo . . . 978
8.978.2.3 pMessageModeInfo . . . 978
8.978.2.4 pMTMessageInfo . . . 978
8.978.2.5 pSMSCAddressInfo . . . 978
8.978.2.6 pSMSOnIMSInfo . . . 978
8.978.2.7 pTransferRouteMTMessageInfo . . . 978
8.978.2.8 smsEventType . . . 978
8.979 smsMaxStorageSizeReq Struct Reference . . . 978
8.979.1 Detailed Description . . . 978
8.979.2 Field Documentation . . . 978
8.979.2.1 pMessageMode . . . 979
8.979.2.2 storageType . . . 979
8.980 smsMaxStorageSizeResp Struct Reference . . . 979
8.980.1 Detailed Description . . . 979
8.980.2 Field Documentation . . . 979
8.980.2.1 freeSlots . . . 979
8.980.2.2 maxStorageSize . . . 979
8.981 SMSMemoryInfo Struct Reference . . . 979
8.981.1 Detailed Description . . . 979
8.981.2 Field Documentation . . . 980
8.981.2.1 messageMode . . . 980
8.981.2.2 storageType . . . 980

8.982 SMSMessageMode Struct Reference . . . 980
8.982.1 Detailed Description . . . 980
8.982.2 Field Documentation . . . 980
8.982.2.1 messageMode . . . 980
8.983 sMSMessageModeInfo Struct Reference . . . 980
8.983.1 Detailed Description . . . 980
8.983.2 Field Documentation . . . 980
8.983.2.1 messageMode . . . 980
8.984 smsMsgprotocolResp Struct Reference . . . 980
8.984.1 Detailed Description . . . 981
8.984.2 Field Documentation . . . 981
8.984.2.1 msgProtocol . . . 981
8.985 SMSMTMessage Struct Reference . . . 981
8.985.1 Detailed Description . . . 981
8.985.2 Field Documentation . . . 981
8.985.2.1 messageIndex . . . 981
8.985.2.2 storageType . . . 981
8.986 sMSMTMessageInfo Struct Reference . . . 981
8.986.1 Detailed Description . . . 982
8.986.2 Field Documentation . . . 982
8.986.2.1 messageIndex . . . 982
8.986.2.2 storageType . . . 982
8.987 SMSOnIMS Struct Reference . . . 982
8.987.1 Detailed Description . . . 982
8.987.2 Field Documentation . . . 982
8.987.2.1 smsOnIMS . . . 982
8.988 sMSOnIMSInfo Struct Reference . . . 982
8.988.1 Detailed Description . . . 982
8.988.2 Field Documentation . . . 983
8.988.2.1 smsOnIMS . . . 983
8.989 sMSOnIMSTlv Struct Reference . . . 983
8.989.1 Detailed Description . . . 983
8.989.2 Field Documentation . . . 983
8.989.2.1 IMSInfo . . . 983
8.989.2.2 TlvPresent . . . 983
8.990 smsRouteEntry Struct Reference . . . 983
8.990.1 Detailed Description . . . 983
8.990.2 Field Documentation . . . 984
8.990.2.1 messageClass . . . 984
8.990.2.2 messageType . . . 984

8.990.2.3 receiptAction . . . 984
8.990.2.4 routeStorage . . . 984
8.991 smsSetRoutesReq Struct Reference . . . 984
8.991.1 Detailed Description . . . 984
8.991.2 Field Documentation . . . 985
8.991.2.1 numOfRoutes . . . 985
8.991.2.2 pTransferStatusReport . . . 985
8.991.2.3 routeList . . . 985
8.992 SMSTransferRouteMTMessage Struct Reference . . . 985
8.992.1 Detailed Description . . . 985
8.992.2 Field Documentation . . . 986
8.992.2.1 ackIndicator . . . 986
8.992.2.2 data . . . 986
8.992.2.3 format . . . 986
8.992.2.4 length . . . 986
8.992.2.5 transactionID . . . 986
8.993 sMSTransferRouteMTMessageInfo Struct Reference . . . 986
8.993.1 Detailed Description . . . 986
8.993.2 Field Documentation . . . 986
8.993.2.1 ackIndicator . . . 986
8.993.2.2 data . . . 986
8.993.2.3 format . . . 986
8.993.2.4 length . . . 987
8.993.2.5 transactionID . . . 987
8.994 sQosFlowStat Struct Reference . . . 987
8.994.1 Detailed Description . . . 987
8.994.2 Field Documentation . . . 987
8.994.2.1 bearerId . . . 987
8.994.2.2 tx_bytes . . . 987
8.994.2.3 tx_bytes_drp . . . 987
8.994.2.4 tx_pkt . . . 987
8.994.2.5 tx_pkt_drp . . . 987
8.995 sQosStat Struct Reference . . . 987
8.995.1 Detailed Description . . . 988
8.995.2 Field Documentation . . . 988
8.995.2.1 apnId . . . 988
8.995.2.2 numQosFlow . . . 988
8.995.2.3 qosFlow . . . 988
8.995.2.4 total_rx_bytes . . . 989
8.995.2.5 total_rx_pkt . . . 989

8.995.2.6 total_tx_bytes . . . 989
8.995.2.7 total_tx_bytes_drp . . . 989
8.995.2.8 total_tx_pkt . . . 989
8.995.2.9 total_tx_pkt_drp . . . 989
8.996 SrvStatusInfo Struct Reference . . . 989
8.996.1 Detailed Description . . . 989
8.996.2 Field Documentation . . . 989
8.996.2.1 isPrefDataPath . . . 989
8.996.2.2 srvStatus . . . 989
8.997 ssdatasession_params Struct Reference . . . 989
8.997.1 Detailed Description . . . 990
8.997.2 Field Documentation . . . 991
8.997.2.1 action . . . 991
8.997.2.2 failureReason . . . 992
8.997.2.3 failureReasonv4 . . . 992
8.997.2.4 failureReasonv6 . . . 992
8.997.2.5 instanceId . . . 992
8.997.2.6 ipfamily . . . 992
8.997.2.7 pAuthentication . . . 992
8.997.2.8 pPassword . . . 992
8.997.2.9 pProfileId3GPP . . . 992
8.997.2.10 pProfileId3GPP2 . . . 992
8.997.2.11 pTechnology . . . 992
8.997.2.12 pUsername . . . 992
8.997.2.13 rcv4 . . . 992
8.997.2.14 rcv6 . . . 992
8.997.2.15 sessionId . . . 992
8.997.2.16 v4sessionId . . . 992
8.997.2.17 v6sessionId . . . 992
8.997.2.18 verbFailReason . . . 992
8.997.2.19 verbFailReasonType . . . 992
8.998 SupportedMsgList Struct Reference . . . 992
8.998.1 Detailed Description . . . 992
8.998.2 Field Documentation . . . 993
8.998.2.1 supportedMsgLen . . . 993
8.998.2.2 supportedMsgs . . . 993
8.999 SUPSInfo Struct Reference . . . 993
8.999.1 Detailed Description . . . 993
8.999.2 Field Documentation . . . 993
8.999.2.1 isModByCC . . . 993

8.999.2.2 svcType . . . 993
8.1000 SV Struct Reference . . . 993
8.1000.1 Detailed Description . . . 994
8.1000.2 Field Documentation . . . 994
8.1000.2.1 id . . . 994
8.1000.2.2 mask . . . 994
8.1000.2.3 system . . . 994
8.1001 SVInfo Struct Reference . . . 994
8.1001.1 Detailed Description . . . 994
8.1001.2 Field Documentation . . . 995
8.1001.2.1 len . . . 995
8.1001.2.2 pSV . . . 995
8.1002 svUsedforFix_s Struct Reference . . . 995
8.1002.1 Detailed Description . . . 995
8.1002.2 Field Documentation . . . 995
8.1002.2.1 gnssSvUsedList . . . 995
8.1002.2.2 gnssSvUsedList_len . . . 995
8.1003 SWI_STRUCT_CarrierImage Struct Reference . . . 995
8.1003.1 Detailed Description . . . 996
8.1003.2 Field Documentation . . . 996
8.1003.2.1 m_FwBuildId . . . 996
8.1003.2.2 m_FwImageId . . . 996
8.1003.2.3 m_nCarrierId . . . 996
8.1003.2.4 m_nFolderId . . . 996
8.1003.2.5 m_nStorage . . . 996
8.1003.2.6 m_PriBuildId . . . 997
8.1003.2.7 m_PriImageId . . . 997
8.1004 swidms_ehrpdMTUSizeTlv Struct Reference . . . 997
8.1004.1 Detailed Description . . . 997
8.1004.2 Field Documentation . . . 997
8.1004.2.1 ehrpdMTUSize . . . 997
8.1004.2.2 TlvPresent . . . 997
8.1005 swidms_hrpdMTUSizeTlv Struct Reference . . . 997
8.1005.1 Detailed Description . . . 997
8.1005.2 Field Documentation . . . 997
8.1005.2.1 hrpdMTUSize . . . 998
8.1005.2.2 TlvPresent . . . 998
8.1006 swidms_mtuSize3gppTlv Struct Reference . . . 998
8.1006.1 Detailed Description . . . 998
8.1006.2 Field Documentation . . . 998

8.1006.2.1 MTUSize3gpp . . . 998
8.1006.2.2 TlvPresent . . . 998
8.1007 swidms_usbMTUSizeTlv Struct Reference . . . 998
8.1007.1 Detailed Description . . . 998
8.1007.2 Field Documentation . . . 998
8.1007.2.1 TlvPresent . . . 999
8.1007.2.2 UsbMTUSize . . . 999
8.1008 SwiDmsGetMTUResp Struct Reference . . . 999
8.1008.1 Detailed Description . . . 999
8.1008.2 Field Documentation . . . 999
8.1008.2.1 pEhrpdMTUSize . . . 999
8.1008.2.2 pHrpdMTUSize . . . 999
8.1008.2.3 pMTUSize3gpp . . . 999
8.1008.2.4 pUsbMTUSize . . . 999
8.1009 SwiDmsSLQSGetUsbNetNum Struct Reference . . . 999
8.1009.1 Detailed Description . . . 999
8.1009.2 Field Documentation . . . 1000
8.1009.2.1 usbNetNum . . . 1000
8.1010 SwiLocGetAutoStartResp Struct Reference . . . 1000
8.1010.1 Detailed Description . . . 1000
8.1010.2 Field Documentation . . . 1001
8.1010.2.1 fix_rate . . . 1001
8.1010.2.2 fix_rate_reported . . . 1001
8.1010.2.3 fix_type . . . 1001
8.1010.2.4 fix_type_reported . . . 1001
8.1010.2.5 function . . . 1001
8.1010.2.6 function_reported . . . 1001
8.1010.2.7 max_dist . . . 1001
8.1010.2.8 max_dist_reported . . . 1001
8.1010.2.9 max_time . . . 1001
8.1010.2.10 max_time_reported . . . 1001
8.1011 SwiLocSetAutoStartReq Struct Reference . . . 1001
8.1011.1 Detailed Description . . . 1002
8.1011.2 Field Documentation . . . 1003
8.1011.2.1 fix_rate . . . 1003
8.1011.2.2 fix_type . . . 1003
8.1011.2.3 function . . . 1003
8.1011.2.4 max_dist . . . 1003
8.1011.2.5 max_time . . . 1003
8.1011.2.6 set_fix_rate . . . 1003

8.1011.2.7 set_fix_type . . . 1003
8.1011.2.8 set_function . . . 1003
8.1011.2.9 set_max_dist . . . 1003
8.1011.2.10 set_max_time . . . 1003
8.1012 swiModemStatusResp Struct Reference . . . 1003
8.1012.1 Detailed Description . . . 1003
8.1012.2 Field Documentation . . . 1004
8.1012.2.1 commonInfo . . . 1004
8.1012.2.2 pLTEInfo . . . 1004
8.1013 SwiOTAMsg_s Struct Reference . . . 1004
8.1013.1 Detailed Description . . . 1004
8.1013.2 Field Documentation . . . 1004
8.1013.2.1 data . . . 1004
8.1013.2.2 data_len . . . 1004
8.1013.2.3 pLteNasRelInfo . . . 1004
8.1013.2.4 pTime . . . 1005
8.1013.2.5 type . . . 1005
8.1014 swiPDPRuntimeSettingsReq Struct Reference . . . 1005
8.1014.1 Detailed Description . . . 1005
8.1014.2 Field Documentation . . . 1005
8.1014.2.1 contextId . . . 1005
8.1014.2.2 contextType . . . 1005
8.1015 swiPDPRuntimeSettingsResp Struct Reference . . . 1005
8.1015.1 Detailed Description . . . 1006
8.1015.2 Field Documentation . . . 1007
8.1015.2.1 pAPNName . . . 1007
8.1015.2.2 pBearerId . . . 1007
8.1015.2.3 pContextId . . . 1007
8.1015.2.4 pIPv4Address . . . 1007
8.1015.2.5 pIPv4GWAddress . . . 1007
8.1015.2.6 pIPv6Address . . . 1007
8.1015.2.7 pIPv6GWAddress . . . 1007
8.1015.2.8 pPrDNSIPv4Address . . . 1007
8.1015.2.9 pPrDNSIPv6Address . . . 1007
8.1015.2.10 pPrPCSCFIPv4Address . . . 1007
8.1015.2.11 pPrPCSCFIPv6Address . . . 1007
8.1015.2.12 pSeDNSIPv4Address . . . 1007
8.1015.2.13 pSeDNSIPv6Address . . . 1007
8.1015.2.14 pSePCSCFIPv4Address . . . 1007
8.1015.2.15 pSePCSCFIPv6Address . . . 1007

8.1016 swiQosFilter Struct Reference . . . 1007
8.1016.1 Detailed Description . . . 1008
8.1016.2 Field Documentation . . . 1009
8.1016.2.1 index . . . 1009
8.1016.2.2 pEspSpi . . . 1009
8.1016.2.3 pId . . . 1009
8.1016.2.4 pIPv4DstAddr . . . 1009
8.1016.2.5 pIPv4SrcAddr . . . 1009
8.1016.2.6 pIPv6DstAddr . . . 1009
8.1016.2.7 pIPv6Label . . . 1009
8.1016.2.8 pIPv6SrcAddr . . . 1009
8.1016.2.9 pIPv6TrafCls . . . 1009
8.1016.2.10 pNxtHdrProto . . . 1010
8.1016.2.11 pPrecedence . . . 1010
8.1016.2.12 pTCPDstPort . . . 1010
8.1016.2.13 pTCPSrcPort . . . 1010
8.1016.2.14 pTos . . . 1010
8.1016.2.15 pTranDstPort . . . 1010
8.1016.2.16 pTranSrcPort . . . 1010
8.1016.2.17 pUDPDstPort . . . 1010
8.1016.2.18 pUDPSrcPort . . . 1010
8.1016.2.19 version . . . 1010
8.1017 swiQosFlow Struct Reference . . . 1010
8.1017.1 Detailed Description . . . 1010
8.1017.2 Field Documentation . . . 1012
8.1017.2.1 index . . . 1012
8.1017.2.2 p3GPP2Pri . . . 1012
8.1017.2.3 p3GPPImCn . . . 1013
8.1017.2.4 p3GPPResResidualBER . . . 1013
8.1017.2.5 p3GPPSigInd . . . 1013
8.1017.2.6 p3GPPTraHdlPri . . . 1013
8.1017.2.7 pDataRate . . . 1013
8.1017.2.8 pJitter . . . 1013
8.1017.2.9 pLatency . . . 1013
8.1017.2.10 pLteQci . . . 1013
8.1017.2.11 pMaxAllowedPktSz . . . 1013
8.1017.2.12 pMinPolicedPktSz . . . 1013
8.1017.2.13 pPktErrRate . . . 1013
8.1017.2.14 pProfileId3GPP2 . . . 1013
8.1017.2.15 pTokenBucket . . . 1013

8.1017.2.16 bTrafficClass . . . 1013
8.1018 swiQosGranted Struct Reference . . . 1013
8.1018.1 Detailed Description . . . 1013
8.1018.2 Field Documentation . . . 1013
8.1018.2.1 pRxFlow . . . 1013
8.1018.2.2 pTxFlow . . . 1013
8.1019 swiQosIds Struct Reference . . . 1013
8.1019.1 Detailed Description . . . 1014
8.1019.2 Field Documentation . . . 1014
8.1019.2.1 pIds . . . 1014
8.1019.2.2 sz . . . 1014
8.1020 swiQosModifyReq Struct Reference . . . 1014
8.1020.1 Detailed Description . . . 1014
8.1020.2 Field Documentation . . . 1014
8.1020.2.1 id . . . 1014
8.1020.2.2 pRxFilter . . . 1014
8.1020.2.3 pRxFlow . . . 1014
8.1020.2.4 pTxFilter . . . 1015
8.1020.2.5 pTxFlow . . . 1015
8.1021 swiQosReq Struct Reference . . . 1015
8.1021.1 Detailed Description . . . 1015
8.1021.2 Field Documentation . . . 1015
8.1021.2.1 index . . . 1015
8.1021.2.2 pRxFilter . . . 1015
8.1021.2.3 pRxFlow . . . 1015
8.1021.2.4 pTxFilter . . . 1015
8.1021.2.5 pTxFlow . . . 1015
8.1022 swiRMTrasnferStaticsReq Struct Reference . . . 1015
8.1022.1 Detailed Description . . . 1016
8.1022.2 Field Documentation . . . 1016
8.1022.2.1 bResetStatistics . . . 1016
8.1022.2.2 ulMask . . . 1016
8.1023 sysInfoCommon Struct Reference . . . 1016
8.1023.1 Detailed Description . . . 1016
8.1023.2 Field Documentation . . . 1018
8.1023.2.1 isSysForbidden . . . 1018
8.1023.2.2 isSysForbiddenValid . . . 1018
8.1023.2.3 roamStatus . . . 1018
8.1023.2.4 roamStatusValid . . . 1018
8.1023.2.5 srvCapability . . . 1018

8.1023.2.6 srvCapabilityValid . . . 1018
8.1023.2.7 srvDomain . . . 1018
8.1023.2.8 srvDomainValid . . . 1018
8.1024 sysSelectPrefParamsExt Struct Reference . . . 1019
8.1024.1 Detailed Description . . . 1019
8.1024.2 Field Documentation . . . 1022
8.1024.2.1 pCiotAcqOrderPref . . . 1022
8.1024.2.2 pCiotLteOpMode . . . 1022
8.1024.2.3 pCiotLteOpModePref . . . 1022
8.1024.2.4 pLteM1BandPref . . . 1022
8.1024.2.5 pLteNb1BandPref . . . 1022
8.1024.2.6 pNr5gBandPref . . . 1022
8.1024.2.7 pSysSelectPrefParams . . . 1022
8.1025 _gpsTime Struct Reference . . . 1022
8.1025.1 Field Documentation . . . 1022
8.1025.1.1 gpsTimeOfWeekMs . . . 1022
8.1025.1.2 gpsWeek . . . 1022
8.1026 _sensor Struct Reference . . . 1022
8.1026.1 Field Documentation . . . 1023
8.1026.1.1 aidingIndicatorMask . . . 1023
8.1026.1.2 usageMask . . . 1023
8.1027 _Sv Struct Reference . . . 1023
8.1027.1 Field Documentation . . . 1023
8.1027.1.1 entries . . . 1023
8.1027.1.2 len . . . 1023
8.1028 TDSCDMAECIOThresh Struct Reference . . . 1023
8.1028.1 Detailed Description . . . 1023
8.1028.2 Field Documentation . . . 1023
8.1028.2.1 pTDSCDMAECIOThreshList . . . 1023
8.1028.2.2 TDSCDMAECIOThreshListLen . . . 1023
8.1029 TDSCDMARSCPThresh Struct Reference . . . 1024
8.1029.1 Detailed Description . . . 1024
8.1029.2 Field Documentation . . . 1024
8.1029.2.1 pTDSCDMARSCPThreshList . . . 1024
8.1029.2.2 TDSCDMARSCPThreshListLen . . . 1024
8.1030 TDSCDMARSSIThresh Struct Reference . . . 1024
8.1030.1 Detailed Description . . . 1024
8.1030.2 Field Documentation . . . 1024
8.1030.2.1 pTDSCDMARSSIThreshList . . . 1024
8.1030.2.2 TDSCDMARSSIThreshListLen . . . 1025

8.1031 TDSCDMASigInfoExt Struct Reference . . . 1025
8.1031.1 Detailed Description . . . 1025
8.1031.2 Field Documentation . . . 1025
8.1031.2.1 ecio . . . 1025
8.1031.2.2 rscp . . . 1025
8.1031.2.3 rssi . . . 1025
8.1031.2.4 sinr . . . 1025
8.1032 tdscdmaSigInfoExt Struct Reference . . . 1025
8.1032.1 Detailed Description . . . 1025
8.1032.2 Field Documentation . . . 1026
8.1032.2.1 ecio . . . 1026
8.1032.2.2 rscp . . . 1026
8.1032.2.3 rssi . . . 1026
8.1032.2.4 sinr . . . 1026
8.1033 TDSCDMASINRCONFThresh Struct Reference . . . 1026
8.1033.1 Detailed Description . . . 1026
8.1033.2 Field Documentation . . . 1026
8.1033.2.1 pTDSCDMASINRCONFThreshList . . . 1026
8.1033.2.2 TDSCDMASINRCONFThreshListLen . . . 1026
8.1034 TDSCDMASINRThresh Struct Reference . . . 1027
8.1034.1 Detailed Description . . . 1027
8.1034.2 Field Documentation . . . 1027
8.1034.2.1 pTDSCDMASINRThreshList . . . 1027
8.1034.2.2 TDSCDMASINRThreshListLen . . . 1027
8.1035 tempData_t Struct Reference . . . 1027
8.1035.1 Detailed Description . . . 1027
8.1035.2 Field Documentation . . . 1028
8.1035.2.1 temperature . . . 1028
8.1035.2.2 temperatureDataLen . . . 1028
8.1035.2.3 timeOfFirstSample . . . 1028
8.1035.2.4 timeOffset . . . 1028
8.1035.2.5 timeSource . . . 1028
8.1036 tempratureData Struct Reference . . . 1028
8.1036.1 Detailed Description . . . 1028
8.1036.2 Field Documentation . . . 1029
8.1036.2.1 temperature . . . 1029
8.1036.2.2 temperatureDataLen . . . 1029
8.1036.2.3 timeOfFirstSample . . . 1029
8.1036.2.4 timeOffset . . . 1029
8.1036.2.5 timeSource . . . 1029

8.1037 TFTIDParams Struct Reference . . . 1029
8.1037.1 Detailed Description . . . 1030
8.1037.2 Field Documentation . . . 1031
8.1037.2.1 destPortRangeEnd . . . 1031
8.1037.2.2 destPortRangeStart . . . 1031
8.1037.2.3 eValid . . . 1031
8.1037.2.4 filterId . . . 1031
8.1037.2.5 flowLabel . . . 1031
8.1037.2.6 IPSECSPI . . . 1031
8.1037.2.7 ipVersion . . . 1031
8.1037.2.8 nextHeader . . . 1031
8.1037.2.9 pSourceIP . . . 1031
8.1037.2.10 sourceIPMask . . . 1031
8.1037.2.11 srcPortRangeEnd . . . 1031
8.1037.2.12 srcPortRangeStart . . . 1031
8.1037.2.13 tosMask . . . 1031
8.1038 TimeInfo Struct Reference . . . 1031
8.1038.1 Detailed Description . . . 1031
8.1038.2 Field Documentation . . . 1033
8.1038.2.1 day . . . 1033
8.1038.2.2 dayLtSavingAdj . . . 1033
8.1038.2.3 dayOfWeek . . . 1033
8.1038.2.4 hour . . . 1033
8.1038.2.5 minute . . . 1033
8.1038.2.6 month . . . 1033
8.1038.2.7 radioInterface . . . 1033
8.1038.2.8 second . . . 1033
8.1038.2.9 timeZone . . . 1033
8.1038.2.10 TmValPresent . . . 1033
8.1038.2.11 year . . . 1033
8.1039 TmdDeRegNotMitigationLvlReq Struct Reference . . . 1033
8.1039.1 Detailed Description . . . 1033
8.1039.2 Field Documentation . . . 1033
8.1039.2.1 mitigationDevID . . . 1033
8.1039.2.2 mitigationDevIDLen . . . 1033
8.1040 TmdGetMitigationDevListResp Struct Reference . . . 1033
8.1040.1 Detailed Description . . . 1034
8.1040.2 Field Documentation . . . 1034
8.1040.2.1 pMitigationDevList . . . 1034
8.1040.2.2 pMitigationDevListLen . . . 1034

8.1041 TmdGetMitigationLvlReq Struct Reference . . . 1034
8.1041.1 Detailed Description . . . 1034
8.1041.2 Field Documentation . . . 1034
8.1041.2.1 mitigationDevID . . . 1034
8.1041.2.2 mitigationDevIDLen . . . 1035
8.1042 TmdGetMitigationLvlResp Struct Reference . . . 1035
8.1042.1 Detailed Description . . . 1035
8.1042.2 Field Documentation . . . 1035
8.1042.2.1 pCurrentmitigationLvl . . . 1035
8.1042.2.2 pReqMitigationLvl . . . 1035
8.1043 TmdMitigationLvlIndReq Struct Reference . . . 1035
8.1043.1 Detailed Description . . . 1035
8.1043.2 Field Documentation . . . 1036
8.1043.2.1 mitigationDevID . . . 1036
8.1043.2.2 mitigationDevIDLen . . . 1036
8.1044 TmdRegNotMitigationLvlReq Struct Reference . . . 1036
8.1044.1 Detailed Description . . . 1036
8.1044.2 Field Documentation . . . 1036
8.1044.2.1 mitigationDevID . . . 1036
8.1044.2.2 mitigationDevIDLen . . . 1036
8.1045 TokenBucket Struct Reference . . . 1036
8.1045.1 Detailed Description . . . 1036
8.1045.2 Field Documentation . . . 1037
8.1045.2.1 bucketSz . . . 1037
8.1045.2.2 peakRate . . . 1037
8.1045.2.3 tokenRate . . . 1037
8.1046 Tos Struct Reference . . . 1037
8.1046.1 Detailed Description . . . 1037
8.1046.2 Field Documentation . . . 1037
8.1046.2.1 mask . . . 1037
8.1046.2.2 val . . . 1037
8.1047 TransferRouteMessageTlv Struct Reference . . . 1037
8.1047.1 Detailed Description . . . 1037
8.1047.2 Field Documentation . . . 1038
8.1047.2.1 TlvPresent . . . 1038
8.1047.2.2 TransferRouteMTMessageInfo . . . 1038
8.1048 TransferStatInd Struct Reference . . . 1038
8.1048.1 Detailed Description . . . 1038
8.1048.2 Field Documentation . . . 1038
8.1048.2.1 StatsMask . . . 1038

8.1048.2.2 StatsPeriod . . . 1038
8.1049 TransferStatsDataType Struct Reference . . . 1038
8.1049.1 Field Documentation . . . 1038
8.1049.1.1 interval . . . 1039
8.1050 TrStatInd Struct Reference . . . 1039
8.1050.1 Detailed Description . . . 1039
8.1050.2 Field Documentation . . . 1039
8.1050.2.1 statsMask . . . 1039
8.1050.2.2 statsPeriod . . . 1039
8.1051 TrueIMSI Struct Reference . . . 1039
8.1051.1 Detailed Description . . . 1040
8.1051.2 Field Documentation . . . 1040
8.1051.2.1 imsiT1112 . . . 1040
8.1051.2.2 imsiTaddrNum . . . 1040
8.1051.2.3 imsiTS1 . . . 1040
8.1051.2.4 imsiTS2 . . . 1040
8.1051.2.5 mccT . . . 1040
8.1052 TXAGCList Struct Reference . . . 1040
8.1052.1 Detailed Description . . . 1040
8.1052.2 Field Documentation . . . 1041
8.1052.2.1 pTXAIG . . . 1041
8.1052.2.2 pTXComprSlope . . . 1041
8.1052.2.3 pTXComprThres . . . 1041
8.1052.2.4 pTXExpSlope . . . 1041
8.1052.2.5 pTXExpThres . . . 1041
8.1052.2.6 pTXStaticGain . . . 1041
8.1053 TxInfo Struct Reference . . . 1041
8.1053.1 Detailed Description . . . 1041
8.1053.2 Field Documentation . . . 1042
8.1053.2.1 isInTraffic . . . 1042
8.1053.2.2 txPower . . . 1042
8.1054 TXPCMIIRFltr Struct Reference . . . 1042
8.1054.1 Detailed Description . . . 1042
8.1054.2 Field Documentation . . . 1043
8.1054.2.1 pFlag . . . 1043
8.1054.2.2 pStage0Val . . . 1043
8.1054.2.3 pStage1Val . . . 1043
8.1054.2.4 pStage2Val . . . 1043
8.1054.2.5 pStage3Val . . . 1043
8.1054.2.6 pStage4Val . . . 1043

8.1054.2.7 pStageCnt . . . 1043
8.1055 qmi_uim_appStatus Struct Reference . . . 1043
8.1055.1 Detailed Description . . . 1044
8.1055.2 Field Documentation . . . 1046
8.1055.2.1 aidLength . . . 1046
8.1055.2.2 aidVal . . . 1046
8.1055.2.3 appState . . . 1046
8.1055.2.4 appType . . . 1046
8.1055.2.5 persoFeature . . . 1046
8.1055.2.6 persoRetries . . . 1046
8.1055.2.7 persoState . . . 1046
8.1055.2.8 persoUnblockRetries . . . 1046
8.1055.2.9 pin1Retries . . . 1046
8.1055.2.10 pin1State . . . 1046
8.1055.2.11 pin2Retries . . . 1046
8.1055.2.12 pin2State . . . 1046
8.1055.2.13 puk1Retries . . . 1046
8.1055.2.14 puk2Retries . . . 1046
8.1055.2.15 univPin . . . 1046
8.1056 qmi_uim_authenticateResult Struct Reference . . . 1046
8.1056.1 Detailed Description . . . 1047
8.1056.2 Field Documentation . . . 1047
8.1056.2.1 content . . . 1047
8.1056.2.2 contentLen . . . 1047
8.1057 qmi_uim_authenticationData Struct Reference . . . 1047
8.1057.1 Detailed Description . . . 1047
8.1057.2 Field Documentation . . . 1048
8.1057.2.1 context . . . 1048
8.1057.2.2 data . . . 1048
8.1057.2.3 dataLen . . . 1048
8.1058 qmi_uim_cardResult Struct Reference . . . 1048
8.1058.1 Detailed Description . . . 1048
8.1058.2 Field Documentation . . . 1049
8.1058.2.1 sw1 . . . 1049
8.1058.2.2 sw2 . . . 1049
8.1059 qmi_uim_cardStatus Struct Reference . . . 1049
8.1059.1 Detailed Description . . . 1049
8.1059.2 Field Documentation . . . 1050
8.1059.2.1 index1xPri . . . 1050
8.1059.2.2 index1xSec . . . 1050

8.1059.2.3 indexGwPri . . . 1050
8.1059.2.4 indexGwSec . . . 1050
8.1059.2.5 numSlot . . . 1050
8.1059.2.6 SlotInfo . . . 1050
8.1060 uim_changeUIMPIN Struct Reference . . . 1050
8.1060.1 Detailed Description . . . 1050
8.1060.2 Field Documentation . . . 1051
8.1060.2.1 oldPINLen . . . 1051
8.1060.2.2 oldPINVal . . . 1051
8.1060.2.3 pinID . . . 1051
8.1060.2.4 pinLen . . . 1051
8.1060.2.5 pinVal . . . 1051
8.1061 uim_depersonalizationInformation Struct Reference . . . 1051
8.1061.1 Detailed Description . . . 1051
8.1061.2 Field Documentation . . . 1052
8.1061.2.1 ckLen . . . 1052
8.1061.2.2 ckVal . . . 1052
8.1061.2.3 feature . . . 1052
8.1061.2.4 operation . . . 1052
8.1062 uim_encryptedPIN1 Struct Reference . . . 1052
8.1062.1 Detailed Description . . . 1052
8.1062.2 Field Documentation . . . 1053
8.1062.2.1 pin1Len . . . 1053
8.1062.2.2 pin1Val . . . 1053
8.1063 uim_fileAttributes Struct Reference . . . 1053
8.1063.1 Detailed Description . . . 1053
8.1063.2 Field Documentation . . . 1056
8.1063.2.1 fileID . . . 1056
8.1063.2.2 fileSize . . . 1056
8.1063.2.3 fileType . . . 1056
8.1063.2.4 rawLen . . . 1056
8.1063.2.5 rawValue . . . 1056
8.1063.2.6 recordCount . . . 1056
8.1063.2.7 recordSize . . . 1056
8.1063.2.8 secActivate . . . 1056
8.1063.2.9 secActivateMask . . . 1056
8.1063.2.10 secDeactivate . . . 1056
8.1063.2.11 secDeactivateMask . . . 1056
8.1063.2.12 secIncrease . . . 1056
8.1063.2.13 secIncreaseMask . . . 1056

8.1063.2.14 secRead . . . 1056
8.1063.2.15 secReadMask . . . 1056
8.1063.2.16 secWrite . . . 1056
8.1063.2.17 secWriteMask . . . 1056
8.1064 qmi_fileInfo Struct Reference . . . 1056
8.1064.1 Detailed Description . . . 1056
8.1064.2 Field Documentation . . . 1057
8.1064.2.1 fileID . . . 1057
8.1064.2.2 path . . . 1057
8.1064.2.3 pathLen . . . 1057
8.1065 qmi_hotSwapStatus Struct Reference . . . 1057
8.1065.1 Detailed Description . . . 1057
8.1065.2 Field Documentation . . . 1057
8.1065.2.1 hotSwap . . . 1057
8.1065.2.2 hotSwapLength . . . 1057
8.1066 qmi_personalizationStatus Struct Reference . . . 1057
8.1066.1 Detailed Description . . . 1058
8.1066.2 Field Documentation . . . 1058
8.1066.2.1 feature . . . 1058
8.1066.2.2 numFeatures . . . 1058
8.1066.2.3 unblockLeft . . . 1058
8.1066.2.4 verifyLeft . . . 1058
8.1067 qmi_readResult Struct Reference . . . 1058
8.1067.1 Detailed Description . . . 1059
8.1067.2 Field Documentation . . . 1059
8.1067.2.1 content . . . 1059
8.1067.2.2 contentLen . . . 1059
8.1068 qmi_readTransparentInfo Struct Reference . . . 1059
8.1068.1 Detailed Description . . . 1059
8.1068.2 Field Documentation . . . 1059
8.1068.2.1 length . . . 1059
8.1068.2.2 offset . . . 1059
8.1069 qmi_refreshevent Struct Reference . . . 1059
8.1069.1 Detailed Description . . . 1060
8.1069.2 Field Documentation . . . 1061
8.1069.2.1 aid . . . 1061
8.1069.2.2 aidLength . . . 1061
8.1069.2.3 arrfileInfo . . . 1061
8.1069.2.4 mode . . . 1061
8.1069.2.5 numOfFiles . . . 1061

8.1069.2.6 sessionType . . . 1061
8.1069.2.7 stage . . . 1061
8.1070 qim_registerRefresh Struct Reference . . . 1061
8.1070.1 Detailed Description . . . 1061
8.1070.2 Field Documentation . . . 1062
8.1070.2.1 arrfileInfo . . . 1062
8.1070.2.2 numFiles . . . 1062
8.1070.2.3 registerFlag . . . 1062
8.1070.2.4 voteForInit . . . 1062
8.1071 qim_remainingRetries Struct Reference . . . 1062
8.1071.1 Detailed Description . . . 1062
8.1071.2 Field Documentation . . . 1062
8.1071.2.1 unblockLeft . . . 1062
8.1071.2.2 verifyLeft . . . 1063
8.1072 qim_sessionInformation Struct Reference . . . 1063
8.1072.1 Detailed Description . . . 1063
8.1072.2 Field Documentation . . . 1063
8.1072.2.1 aid . . . 1063
8.1072.2.2 aidLength . . . 1063
8.1072.2.3 sessionType . . . 1063
8.1073 qim_setPINProtection Struct Reference . . . 1063
8.1073.1 Detailed Description . . . 1064
8.1073.2 Field Documentation . . . 1064
8.1073.2.1 pinID . . . 1064
8.1073.2.2 pinLength . . . 1064
8.1073.2.3 pinOperation . . . 1064
8.1073.2.4 pinValue . . . 1064
8.1074 qim_slotInfo Struct Reference . . . 1064
8.1074.1 Detailed Description . . . 1065
8.1074.2 Field Documentation . . . 1066
8.1074.2.1 AppStatus . . . 1066
8.1074.2.2 cardState . . . 1066
8.1074.2.3 errorState . . . 1066
8.1074.2.4 numApp . . . 1066
8.1074.2.5 upinRetries . . . 1066
8.1074.2.6 upinState . . . 1066
8.1074.2.7 upukRetries . . . 1066
8.1075 qim_UIMSessionInformation Struct Reference . . . 1066
8.1075.1 Detailed Description . . . 1066
8.1075.2 Field Documentation . . . 1067

8.1075.2.1 aid . . . 1067
8.1075.2.2 aidLength . . . 1067
8.1075.2.3 sessionType . . . 1067
8.1076 uim_unblockUIMPIN Struct Reference . . . 1067
8.1076.1 Detailed Description . . . 1067
8.1076.2 Field Documentation . . . 1067
8.1076.2.1 newPINLen . . . 1067
8.1076.2.2 newPINVal . . . 1067
8.1076.2.3 pinID . . . 1067
8.1076.2.4 pukLen . . . 1068
8.1076.2.5 pukVal . . . 1068
8.1077 uim_verifyUIMPIN Struct Reference . . . 1068
8.1077.1 Detailed Description . . . 1068
8.1077.2 Field Documentation . . . 1068
8.1077.2.1 pinID . . . 1068
8.1077.2.2 pinLen . . . 1068
8.1077.2.3 pinVal . . . 1068
8.1078 UIMAuthenticateReq Struct Reference . . . 1068
8.1078.1 Detailed Description . . . 1068
8.1078.2 Field Documentation . . . 1069
8.1078.2.1 authData . . . 1069
8.1078.2.2 pIndicationToken . . . 1069
8.1078.2.3 sessionInfo . . . 1069
8.1079 UIMAuthenticateResp Struct Reference . . . 1069
8.1079.1 Detailed Description . . . 1069
8.1079.2 Field Documentation . . . 1070
8.1079.2.1 pAuthenticateResult . . . 1070
8.1079.2.2 pCardResult . . . 1070
8.1079.2.3 pIndicationToken . . . 1070
8.1080 UIMChangePinReq Struct Reference . . . 1070
8.1080.1 Detailed Description . . . 1070
8.1080.2 Field Documentation . . . 1070
8.1080.2.1 changePIN . . . 1070
8.1080.2.2 pIndicationToken . . . 1070
8.1080.2.3 pKeyReferenceID . . . 1070
8.1080.2.4 sessionInfo . . . 1070
8.1081 UIMDepersonalizationReq Struct Reference . . . 1071
8.1081.1 Detailed Description . . . 1071
8.1081.2 Field Documentation . . . 1071
8.1081.2.1 depersonilisationInfo . . . 1071

8.1082 UIMDepersonalizationResp Struct Reference . . . 1071
8.1082.1 Detailed Description . . . 1071
8.1082.2 Field Documentation . . . 1071
8.1082.2.1 pRemainingRetries . . . 1071
8.1083 UIMEventRegisterReqResp Struct Reference . . . 1071
8.1083.1 Detailed Description . . . 1072
8.1083.2 Field Documentation . . . 1072
8.1083.2.1 eventMask . . . 1072
8.1084 UIMGetCardStatusResp Struct Reference . . . 1072
8.1084.1 Detailed Description . . . 1072
8.1084.2 Field Documentation . . . 1072
8.1084.2.1 pCardStatus . . . 1072
8.1084.2.2 pHotSwapStatus . . . 1072
8.1085 UIMGetConfigurationReq Struct Reference . . . 1072
8.1085.1 Detailed Description . . . 1073
8.1085.2 Field Documentation . . . 1073
8.1085.2.1 pConfigurationMask . . . 1073
8.1086 UIMGetConfigurationResp Struct Reference . . . 1073
8.1086.1 Detailed Description . . . 1073
8.1086.2 Field Documentation . . . 1074
8.1086.2.1 pAutoSelection . . . 1074
8.1086.2.2 pHaltSubscription . . . 1074
8.1086.2.3 pPersonalizationStatus . . . 1074
8.1087 UIMGetFileAttributesReq Struct Reference . . . 1074
8.1087.1 Detailed Description . . . 1074
8.1087.2 Field Documentation . . . 1074
8.1087.2.1 fileIndex . . . 1074
8.1087.2.2 pIndicationToken . . . 1074
8.1087.2.3 sessionInfo . . . 1074
8.1088 UIMGetFileAttributesResp Struct Reference . . . 1074
8.1088.1 Detailed Description . . . 1075
8.1088.2 Field Documentation . . . 1075
8.1088.2.1 pCardResult . . . 1075
8.1088.2.2 pFileAttributes . . . 1075
8.1088.2.3 pIndicationToken . . . 1075
8.1089 UIMGetSlotsStatusResp Struct Reference . . . 1075
8.1089.1 Detailed Description . . . 1075
8.1089.2 Field Documentation . . . 1075
8.1089.2.1 pNumberOfPhySlot . . . 1075
8.1089.2.2 pUimSlotsStatus . . . 1076

8.1090 UIMPinResp Struct Reference . . . 1076
8.1090.1 Detailed Description . . . 1076
8.1090.2 Field Documentation . . . 1076
8.1090.2.1 pEncryptedPIN1 . . . 1076
8.1090.2.2 pIndicationToken . . . 1076
8.1090.2.3 pRemainingRetries . . . 1076
8.1091 UIMPowerDownReq Struct Reference . . . 1076
8.1091.1 Detailed Description . . . 1076
8.1091.2 Field Documentation . . . 1077
8.1091.2.1 slot . . . 1077
8.1092 UIMPowerUpReq Struct Reference . . . 1077
8.1092.1 Detailed Description . . . 1077
8.1092.2 Field Documentation . . . 1077
8.1092.2.1 pIgnoreHotSwapSwitch . . . 1077
8.1092.2.2 slot . . . 1077
8.1093 UIMReadTransparentReq Struct Reference . . . 1077
8.1093.1 Detailed Description . . . 1078
8.1093.2 Field Documentation . . . 1078
8.1093.2.1 fileIndex . . . 1078
8.1093.2.2 pEncryptData . . . 1078
8.1093.2.3 pIndicationToken . . . 1078
8.1093.2.4 readTransparent . . . 1078
8.1093.2.5 sessionInfo . . . 1078
8.1094 UIMReadTransparentResp Struct Reference . . . 1078
8.1094.1 Detailed Description . . . 1078
8.1094.2 Field Documentation . . . 1079
8.1094.2.1 pCardResult . . . 1079
8.1094.2.2 pEncryptedData . . . 1079
8.1094.2.3 pIndicationToken . . . 1079
8.1094.2.4 pReadResult . . . 1079
8.1095 UIMRefreshCompleteReq Struct Reference . . . 1079
8.1095.1 Detailed Description . . . 1079
8.1095.2 Field Documentation . . . 1080
8.1095.2.1 refreshComplete . . . 1080
8.1095.2.2 sessionInfo . . . 1080
8.1096 UIMRefreshEvent Struct Reference . . . 1080
8.1096.1 Detailed Description . . . 1080
8.1096.2 Field Documentation . . . 1081
8.1096.2.1 aid . . . 1081
8.1096.2.2 aidLength . . . 1081

8.1096.2.3 arrfileInfo . . . 1081
8.1096.2.4 mode . . . 1081
8.1096.2.5 numOfFiles . . . 1081
8.1096.2.6 sessionType . . . 1081
8.1096.2.7 stage . . . 1081
8.1097 UIMRefreshGetLastEventReq Struct Reference . . . 1081
8.1097.1 Detailed Description . . . 1082
8.1097.2 Field Documentation . . . 1082
8.1097.2.1 sessionInfo . . . 1082
8.1098 UIMRefreshGetLastEventResp Struct Reference . . . 1082
8.1098.1 Detailed Description . . . 1082
8.1098.2 Field Documentation . . . 1082
8.1098.2.1 pRefreshEvent . . . 1082
8.1099 UIMRefreshOKReq Struct Reference . . . 1082
8.1099.1 Detailed Description . . . 1082
8.1099.2 Field Documentation . . . 1083
8.1099.2.1 OKtoRefresh . . . 1083
8.1099.2.2 sessionInfo . . . 1083
8.1100 UIMRefreshRegisterReq Struct Reference . . . 1083
8.1100.1 Detailed Description . . . 1083
8.1100.2 Field Documentation . . . 1083
8.1100.2.1 regRefresh . . . 1083
8.1100.2.2 sessionInfo . . . 1083
8.1101 UIMSessionInformation Struct Reference . . . 1083
8.1101.1 Detailed Description . . . 1084
8.1101.2 Field Documentation . . . 1084
8.1101.2.1 aid . . . 1084
8.1101.2.2 aidLength . . . 1084
8.1101.2.3 sessionType . . . 1084
8.1102 UIMSetPinProtectionReq Struct Reference . . . 1084
8.1102.1 Detailed Description . . . 1084
8.1102.2 Field Documentation . . . 1085
8.1102.2.1 pIndicationToken . . . 1085
8.1102.2.2 pinProtection . . . 1085
8.1102.2.3 pKeyReferenceID . . . 1085
8.1102.2.4 sessionInfo . . . 1085
8.1103 UIMSlotsStatus Struct Reference . . . 1085
8.1103.1 Detailed Description . . . 1085
8.1103.2 Field Documentation . . . 1085
8.1103.2.1 uimSlotStatus . . . 1085

8.1104 UIMSlotStatus Struct Reference . . . 1085
8.1104.1 Detailed Description . . . 1086
8.1104.2 Field Documentation . . . 1086
8.1104.2.1 bICCID . . . 1086
8.1104.2.2 bICCIDLength . . . 1086
8.1104.2.3 bLogicalSlot . . . 1086
8.1104.2.4 uPhyCardStatus . . . 1086
8.1104.2.5 uPhySlotStatus . . . 1086
8.1105 UIMSlotStatusChangeInfo Struct Reference . . . 1087
8.1105.1 Detailed Description . . . 1087
8.1105.2 Field Documentation . . . 1087
8.1105.2.1 bNumberOfPhySlots . . . 1087
8.1105.2.2 slotsstatusChange . . . 1087
8.1106 UIMStatusChangeInfo Struct Reference . . . 1087
8.1106.1 Detailed Description . . . 1087
8.1106.2 Field Documentation . . . 1087
8.1106.2.1 statusChange . . . 1087
8.1107 UIMSwitchSlotReq Struct Reference . . . 1087
8.1107.1 Detailed Description . . . 1088
8.1107.2 Field Documentation . . . 1088
8.1107.2.1 bLogicalSlot . . . 1088
8.1107.2.2 ulPhysicalSlot . . . 1088
8.1108 UIMUnblockPinReq Struct Reference . . . 1088
8.1108.1 Detailed Description . . . 1088
8.1108.2 Field Documentation . . . 1089
8.1108.2.1 pIndicationToken . . . 1089
8.1108.2.2 pKeyReferenceID . . . 1089
8.1108.2.3 sessionInfo . . . 1089
8.1108.2.4 unblockPIN . . . 1089
8.1109 UIMVerifyPinReq Struct Reference . . . 1089
8.1109.1 Detailed Description . . . 1089
8.1109.2 Field Documentation . . . 1090
8.1109.2.1 pEncryptedPIN1 . . . 1090
8.1109.2.2 pIndicationToken . . . 1090
8.1109.2.3 pKeyReferenceID . . . 1090
8.1109.2.4 sessionInfo . . . 1090
8.1109.2.5 verifyPIN . . . 1090
8.1110 UMTSInfo Struct Reference . . . 1090
8.1110.1 Detailed Description . . . 1090
8.1110.2 Field Documentation . . . 1091

8.1110.2.1 cellID . . . 1091
8.1110.2.2 ecio . . . 1091
8.1110.2.3 geranInst . . . 1091
8.1110.2.4 GeranInstInfo . . . 1091
8.1110.2.5 lac . . . 1091
8.1110.2.6 plmn . . . 1092
8.1110.2.7 psc . . . 1092
8.1110.2.8 rscp . . . 1092
8.1110.2.9 uarfcn . . . 1092
8.1110.2.10 umtsInst . . . 1092
8.1110.2.11 UMTSInstInfo . . . 1092
8.1111 UMTSinstInfo Struct Reference . . . 1092
8.1111.1 Detailed Description . . . 1092
8.1111.2 Field Documentation . . . 1092
8.1111.2.1 umtsEcio . . . 1092
8.1111.2.2 umtsPsc . . . 1092
8.1111.2.3 umtsRscp . . . 1092
8.1111.2.4 umtsUarfcn . . . 1092
8.1112 umtsLTENbrCell Struct Reference . . . 1092
8.1112.1 Detailed Description . . . 1093
8.1112.2 Field Documentation . . . 1093
8.1112.2.1 cellIsTDD . . . 1093
8.1112.2.2 earfcn . . . 1093
8.1112.2.3 pci . . . 1093
8.1112.2.4 rsrp . . . 1093
8.1112.2.5 rsrq . . . 1093
8.1112.2.6 srxlev . . . 1093
8.1113 UMTSMinQoS Struct Reference . . . 1093
8.1113.1 Detailed Description . . . 1094
8.1113.2 Field Documentation . . . 1095
8.1113.2.1 deliveryErrSDU . . . 1095
8.1113.2.2 grntDownlinkBitrate . . . 1096
8.1113.2.3 grntUplinkBitrate . . . 1096
8.1113.2.4 maxDownlinkBitrate . . . 1096
8.1113.2.5 maxSDUSize . . . 1096
8.1113.2.6 maxUplinkBitrate . . . 1096
8.1113.2.7 qosDeliveryOrder . . . 1096
8.1113.2.8 resBerRatio . . . 1096
8.1113.2.9 sduErrorRatio . . . 1096
8.1113.2.10 trafficClass . . . 1096

8.1113.2.11trafficPriority . . . 1096
8.1113.2.12transferDelay . . . 1096
8.1114UMTSQoS Struct Reference . . . 1096
8.1114.1Detailed Description . . . 1096
8.1114.2Field Documentation . . . 1098
8.1114.2.1deliveryErrSDU . . . 1098
8.1114.2.2grntDownlinkBitrate . . . 1098
8.1114.2.3grntUplinkBitrate . . . 1098
8.1114.2.4maxDownlinkBitrate . . . 1098
8.1114.2.5maxSDUSize . . . 1098
8.1114.2.6maxUplinkBitrate . . . 1098
8.1114.2.7qosDeliveryOrder . . . 1098
8.1114.2.8resBerRatio . . . 1098
8.1114.2.9sduErrorRatio . . . 1098
8.1114.2.10trafficClass . . . 1098
8.1114.2.11trafficPriority . . . 1098
8.1114.2.12transferDelay . . . 1098
8.1115UMTSReqQoSSigInd Struct Reference . . . 1098
8.1115.1Detailed Description . . . 1099
8.1115.2Field Documentation . . . 1099
8.1115.2.1SigInd . . . 1099
8.1115.2.2UMTSReqQoS . . . 1099
8.1116UnblockUIMPIN Struct Reference . . . 1099
8.1116.1Detailed Description . . . 1099
8.1116.2Field Documentation . . . 1100
8.1116.2.1newPINLen . . . 1100
8.1116.2.2newPINVal . . . 1100
8.1116.2.3pinID . . . 1100
8.1116.2.4pukLen . . . 1100
8.1116.2.5pukVal . . . 1100
8.1117UniversalTime Struct Reference . . . 1100
8.1117.1Detailed Description . . . 1100
8.1117.2Field Documentation . . . 1101
8.1117.2.1day . . . 1101
8.1117.2.2dayOfWeek . . . 1101
8.1117.2.3hour . . . 1101
8.1117.2.4minute . . . 1101
8.1117.2.5month . . . 1101
8.1117.2.6second . . . 1101
8.1117.2.7year . . . 1101

8.1118 npack_dms_GetActivationState_t Struct Reference . . . 1101
8.1118.1 Detailed Description . . . 1101
8.1118.2 Field Documentation . . . 1102
8.1118.2.1 state . . . 1102
8.1119 npack_dms_GetBandCapability_t Struct Reference . . . 1102
8.1119.1 Detailed Description . . . 1102
8.1119.2 Field Documentation . . . 1103
8.1119.2.1 BandCapability . . . 1104
8.1119.2.2 Tlvresult . . . 1104
8.1120 npack_dms_GetCrashAction_t Struct Reference . . . 1104
8.1120.1 Detailed Description . . . 1104
8.1120.2 Field Documentation . . . 1104
8.1120.2.1 DevCrashState . . . 1104
8.1120.2.2 Tlvresult . . . 1104
8.1121 npack_dms_GetCustFeature_t Struct Reference . . . 1104
8.1121.1 Detailed Description . . . 1105
8.1121.2 Field Documentation . . . 1106
8.1121.2.1 DHCPRelayEnabled . . . 1106
8.1121.2.2 DisableIMSI . . . 1106
8.1121.2.3 GpsEnable . . . 1106
8.1121.2.4 GPSLPM . . . 1106
8.1121.2.5 GPSSel . . . 1106
8.1121.2.6 IPFamSupport . . . 1106
8.1121.2.7 IsVoiceEnabled . . . 1107
8.1121.2.8 RMAutoConnect . . . 1107
8.1121.2.9 SMSSupport . . . 1107
8.1121.2.10 Tlvresult . . . 1107
8.1122 npack_dms_GetCustFeaturesV2_t Struct Reference . . . 1107
8.1122.1 Detailed Description . . . 1107
8.1122.2 Field Documentation . . . 1107
8.1122.2.1 GetCustomFeatureV2 . . . 1107
8.1122.2.2 Tlvresult . . . 1107
8.1123 npack_dms_GetDeviceCap_t Struct Reference . . . 1107
8.1123.1 Detailed Description . . . 1107
8.1123.2 Field Documentation . . . 1108
8.1123.2.1 DataServiceCapability . . . 1108
8.1123.2.2 MaxRXChannelRate . . . 1108
8.1123.2.3 MaxTXChannelRate . . . 1108
8.1123.2.4 RadioIfaces . . . 1108
8.1123.2.5 RadioIfacesSize . . . 1108

8.1123.2.6 SimCapability . . . 1108
8.1123.2.7 Tlvresult . . . 1108
8.1124 npack_dms_GetDeviceCapabilities_t Struct Reference . . . 1108
8.1124.1 Detailed Description . . . 1109
8.1124.2 Field Documentation . . . 1109
8.1124.2.1 dataServiceCaCapability . . . 1109
8.1124.2.2 maxRxChannelRate . . . 1109
8.1124.2.3 maxTxChannelRate . . . 1110
8.1124.2.4 RadioIfaces . . . 1110
8.1124.2.5 radioIfacesSize . . . 1110
8.1124.2.6 simCapability . . . 1110
8.1125 npack_dms_GetDeviceHardwareRev_t Struct Reference . . . 1110
8.1125.1 Detailed Description . . . 1110
8.1125.2 Field Documentation . . . 1110
8.1125.2.1 String . . . 1110
8.1125.2.2 stringSize . . . 1110
8.1125.2.3 Tlvresult . . . 1110
8.1126 npack_dms_GetDeviceMfr_t Struct Reference . . . 1110
8.1126.1 Detailed Description . . . 1111
8.1126.2 Field Documentation . . . 1111
8.1126.2.1 String . . . 1111
8.1126.2.2 stringSize . . . 1111
8.1126.2.3 Tlvresult . . . 1111
8.1127 npack_dms_GetDeviceSerialNumbers_t Struct Reference . . . 1111
8.1127.1 Detailed Description . . . 1111
8.1127.2 Field Documentation . . . 1112
8.1127.2.1 esnSize . . . 1112
8.1127.2.2 ESNString . . . 1112
8.1127.2.3 meiSize . . . 1112
8.1127.2.4 MEIString . . . 1112
8.1127.2.5 meiSvnSize . . . 1112
8.1127.2.6 meiSvnString . . . 1112
8.1127.2.7 meidSize . . . 1112
8.1127.2.8 MEIDString . . . 1112
8.1127.2.9 Tlvresult . . . 1112
8.1128 npack_dms_GetFirmwareInfo_t Struct Reference . . . 1112
8.1128.1 Detailed Description . . . 1113
8.1128.2 Field Documentation . . . 1114
8.1128.2.1 appversion_str . . . 1114
8.1128.2.2 bootversion_str . . . 1114

8.1128.2.3 carrier_str . . . 1114
8.1128.2.4 cur_carr_name . . . 1114
8.1128.2.5 cur_carr_rev . . . 1114
8.1128.2.6 modelid_str . . . 1114
8.1128.2.7 packageid_str . . . 1114
8.1128.2.8 priversion_str . . . 1114
8.1128.2.9 sku_str . . . 1114
8.1128.2.10 qmiresult . . . 1114
8.1129 qmi_nas_get_firmware_revision_t Struct Reference . . . 1114
8.1129.1 Detailed Description . . . 1114
8.1129.2 Field Documentation . . . 1115
8.1129.2.1 amssSize . . . 1115
8.1129.2.2 AMSSString . . . 1115
8.1129.2.3 PRIString . . . 1115
8.1129.2.4 qmiresult . . . 1115
8.1130 qmi_nas_get_firmware_revisions_t Struct Reference . . . 1115
8.1130.1 Detailed Description . . . 1115
8.1130.2 Field Documentation . . . 1116
8.1130.2.1 amssSize . . . 1116
8.1130.2.2 AMSSString . . . 1116
8.1130.2.3 bootSize . . . 1116
8.1130.2.4 BootString . . . 1116
8.1130.2.5 priSize . . . 1116
8.1130.2.6 PRIString . . . 1116
8.1130.2.7 qmiresult . . . 1116
8.1131 qmi_nas_get_fsn_t Struct Reference . . . 1116
8.1131.1 Detailed Description . . . 1116
8.1131.2 Field Documentation . . . 1117
8.1131.2.1 String . . . 1117
8.1131.2.2 qmiresult . . . 1117
8.1132 qmi_nas_get_hardware_revision_t Struct Reference . . . 1117
8.1132.1 Detailed Description . . . 1117
8.1132.2 Field Documentation . . . 1117
8.1132.2.1 hwVer . . . 1117
8.1133 qmi_nas_get_imsi_t Struct Reference . . . 1117
8.1133.1 Detailed Description . . . 1117
8.1133.2 Field Documentation . . . 1117
8.1133.2.1 imsi . . . 1117
8.1133.2.2 qmiresult . . . 1117
8.1134 qmi_nas_get_manufacturer_t Struct Reference . . . 1118

8.1134.1 Detailed Description . . . 1118
8.1134.2 Field Documentation . . . 1118
8.1134.2.1 manufacturer . . . 1118
8.1134.2.2 Tlvresult . . . 1118
8.1135 npack_dms_GetModelID_t Struct Reference . . . 1118
8.1135.1 Detailed Description . . . 1118
8.1135.2 Field Documentation . . . 1118
8.1135.2.1 modelid . . . 1118
8.1135.2.2 Tlvresult . . . 1118
8.1136 npack_dms_GetNetworkTime_t Struct Reference . . . 1119
8.1136.1 Detailed Description . . . 1119
8.1136.2 Field Documentation . . . 1119
8.1136.2.1 source . . . 1119
8.1136.2.2 timestamp . . . 1119
8.1136.2.3 Tlvresult . . . 1119
8.1137 npack_dms_GetOfflineReason_t Struct Reference . . . 1119
8.1137.1 Detailed Description . . . 1119
8.1137.2 Field Documentation . . . 1120
8.1137.2.1 pbPlatform . . . 1120
8.1137.2.2 pReasonMask . . . 1120
8.1137.2.3 Tlvresult . . . 1120
8.1138 npack_dms_GetPower_t Struct Reference . . . 1120
8.1138.1 Detailed Description . . . 1120
8.1138.2 Field Documentation . . . 1121
8.1138.2.1 HardwareControlledMode . . . 1121
8.1138.2.2 OfflineReason . . . 1121
8.1138.2.3 OperationMode . . . 1121
8.1138.2.4 Tlvresult . . . 1121
8.1139 npack_dms_GetPRLVersion_t Struct Reference . . . 1122
8.1139.1 Detailed Description . . . 1122
8.1139.2 Field Documentation . . . 1122
8.1139.2.1 Tlvresult . . . 1122
8.1139.2.2 u16PRLVersion . . . 1122
8.1139.2.3 u8PRLPreference . . . 1122
8.1140 npack_dms_GetSerialNumbers_t Struct Reference . . . 1122
8.1140.1 Detailed Description . . . 1122
8.1140.2 Field Documentation . . . 1123
8.1140.2.1 esn . . . 1123
8.1140.2.2 mei_no . . . 1123
8.1140.2.3 meisv_svn . . . 1123

8.1140.2.4 meid . . . 1123
8.1141 unpack_dms_GetUSBComp_t Struct Reference . . . 1123
8.1141.1 Detailed Description . . . 1123
8.1141.2 Field Documentation . . . 1125
8.1141.2.1 NumSupUSBComps . . . 1125
8.1141.2.2 SupUSBComps . . . 1125
8.1141.2.3 Tlvresult . . . 1125
8.1141.2.4 USBComp . . . 1125
8.1142 unpack_dms_GetVoiceNumber_t Struct Reference . . . 1125
8.1142.1 Detailed Description . . . 1126
8.1142.2 Field Documentation . . . 1126
8.1142.2.1 MIN . . . 1126
8.1142.2.2 minSize . . . 1126
8.1142.2.3 Tlvresult . . . 1126
8.1142.2.4 VoiceNumber . . . 1126
8.1142.2.5 voiceNumberSize . . . 1126
8.1143 unpack_dms_PSMCfgChange_ind_t Struct Reference . . . 1126
8.1143.1 Detailed Description . . . 1126
8.1143.2 Field Documentation . . . 1127
8.1143.2.1 ActiveTimer . . . 1127
8.1143.2.2 EnableState . . . 1127
8.1143.2.3 PeriodicUpdateTimer . . . 1127
8.1143.2.4 Tlvresult . . . 1127
8.1144 unpack_dms_ResetToFactoryDefaults_t Struct Reference . . . 1127
8.1144.1 Detailed Description . . . 1127
8.1144.2 Field Documentation . . . 1127
8.1144.2.1 Tlvresult . . . 1127
8.1145 unpack_dms_SetActivationStatusCallback_t Struct Reference . . . 1127
8.1145.1 Detailed Description . . . 1128
8.1145.2 Field Documentation . . . 1128
8.1145.2.1 Tlvresult . . . 1128
8.1146 unpack_dms_SetCrashAction_t Struct Reference . . . 1128
8.1146.1 Detailed Description . . . 1128
8.1146.2 Field Documentation . . . 1128
8.1146.2.1 notused . . . 1128
8.1147 unpack_dms_SetCustFeature_t Struct Reference . . . 1128
8.1147.1 Detailed Description . . . 1128
8.1147.2 Field Documentation . . . 1128
8.1147.2.1 Tlvresult . . . 1129
8.1148 unpack_dms_SetCustFeaturesV2_t Struct Reference . . . 1129

8.1148.1 Detailed Description . . . 1129
8.1148.2 Field Documentation . . . 1129
8.1148.2.1 TlvResult . . . 1129
8.1149 qmi_dms_unpack_dms_SetEventReport_ind_t Struct Reference . . . 1129
8.1149.1 Detailed Description . . . 1129
8.1149.2 Field Documentation . . . 1129
8.1149.2.1 ActivationStatusTlv . . . 1129
8.1149.2.2 OperatingModeTlv . . . 1129
8.1149.2.3 TlvResult . . . 1130
8.1150 qmi_dms_unpack_dms_SetEventReport_t Struct Reference . . . 1130
8.1150.1 Detailed Description . . . 1130
8.1150.2 Field Documentation . . . 1130
8.1150.2.1 TlvResult . . . 1130
8.1151 qmi_dms_unpack_dms_SetFirmwarePreference_t Struct Reference . . . 1130
8.1151.1 Detailed Description . . . 1130
8.1151.2 Field Documentation . . . 1130
8.1151.2.1 TlvResult . . . 1130
8.1152 qmi_dms_unpack_dms_SetIndicationRegister_t Struct Reference . . . 1130
8.1152.1 Detailed Description . . . 1130
8.1152.2 Field Documentation . . . 1131
8.1152.2.1 TlvResult . . . 1131
8.1153 qmi_dms_unpack_dms_SetPower_t Struct Reference . . . 1131
8.1153.1 Detailed Description . . . 1131
8.1153.2 Field Documentation . . . 1131
8.1153.2.1 TlvResult . . . 1131
8.1154 qmi_dms_unpack_dms_SetUSBComp_t Struct Reference . . . 1131
8.1154.1 Detailed Description . . . 1131
8.1154.2 Field Documentation . . . 1131
8.1154.2.1 TlvResult . . . 1131
8.1155 qmi_dms_unpack_dms_SLQSDmsSwiGetPCInfo_t Struct Reference . . . 1131
8.1155.1 Detailed Description . . . 1132
8.1155.2 Field Documentation . . . 1133
8.1155.2.1 has_LpmFlag . . . 1133
8.1155.2.2 has_PersistentLpm . . . 1133
8.1155.2.3 has_PowerOffMode . . . 1133
8.1155.2.4 has_Wdisable . . . 1133
8.1155.2.5 LpmFlag . . . 1133
8.1155.2.6 opMode . . . 1133
8.1155.2.7 PersistentLpm . . . 1133
8.1155.2.8 PowerOffMode . . . 1133

8.1155.2.9 Wdisable . . . 1133
8.1156 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference . . . 1133
8.1156.1 Detailed Description . . . 1134
8.1156.2 Field Documentation . . . 1134
8.1156.2.1 source . . . 1134
8.1156.2.2 Tlvresult . . . 1134
8.1156.2.3 type . . . 1134
8.1157 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference . . . 1134
8.1157.1 Detailed Description . . . 1135
8.1157.2 Field Documentation . . . 1135
8.1157.2.1 source . . . 1135
8.1157.2.2 Tlvresult . . . 1135
8.1157.2.3 type . . . 1135
8.1158 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 1135
8.1158.1 Detailed Description . . . 1135
8.1158.2 Field Documentation . . . 1136
8.1158.2.1 Tlvresult . . . 1136
8.1159 unpack_dms_SLQSGetBandCapability_t Struct Reference . . . 1136
8.1159.1 Detailed Description . . . 1136
8.1159.2 Field Documentation . . . 1139
8.1159.2.1 bandCapability . . . 1139
8.1159.2.2 is_LteBandCapability_Available . . . 1139
8.1159.2.3 is_TdsBandCapability_Available . . . 1139
8.1159.2.4 LteBandCapability . . . 1139
8.1159.2.5 TdsBandCapability . . . 1139
8.1160 unpack_dms_SLQSGetERIFile_t Struct Reference . . . 1139
8.1160.1 Detailed Description . . . 1139
8.1160.2 Field Documentation . . . 1139
8.1160.2.1 eriFile . . . 1139
8.1160.2.2 Tlvresult . . . 1139
8.1161 unpack_dms_SLQSGetPowerSaveModeConfig_t Struct Reference . . . 1139
8.1161.1 Detailed Description . . . 1140
8.1161.2 Field Documentation . . . 1140
8.1161.2.1 pActiveTimer . . . 1140
8.1161.2.2 pDurationDueToOOS . . . 1140
8.1161.2.3 pDurationThreshold . . . 1140
8.1161.2.4 pEarlyWakeupTime . . . 1140
8.1161.2.5 pPeriodicUpdateTimer . . . 1140
8.1161.2.6 pPsmEnableState . . . 1140
8.1161.2.7 pRandomizationWindow . . . 1140

8.1162 npack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference . . . 1141
8.1162.1 Detailed Description . . . 1141
8.1162.2 Field Documentation . . . 1141
8.1162.2.1 TlvResult . . . 1141
8.1163 npack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference . . . 1141
8.1163.1 Detailed Description . . . 1141
8.1163.2 Field Documentation . . . 1141
8.1163.2.1 TlvResult . . . 1141
8.1164 npack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 1141
8.1164.1 Detailed Description . . . 1141
8.1164.2 Field Documentation . . . 1142
8.1164.2.1 crashInfoParam . . . 1142
8.1164.2.2 TlvResult . . . 1142
8.1165 npack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference . . . 1142
8.1165.1 Detailed Description . . . 1142
8.1165.2 Field Documentation . . . 1142
8.1165.2.1 pGetDyingGaspCfg . . . 1142
8.1165.2.2 TlvResult . . . 1142
8.1166 npack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference . . . 1142
8.1166.1 Detailed Description . . . 1143
8.1166.2 Field Documentation . . . 1143
8.1166.2.1 pGetDyingGaspStatistics . . . 1143
8.1166.2.2 TlvResult . . . 1143
8.1167 npack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference . . . 1143
8.1167.1 Detailed Description . . . 1143
8.1167.2 Field Documentation . . . 1144
8.1167.2.1 carrier . . . 1144
8.1167.2.2 fwvers . . . 1144
8.1167.2.3 numEntries . . . 1144
8.1167.2.4 pCurrImgInfo . . . 1144
8.1167.2.5 pkgver . . . 1144
8.1167.2.6 priver . . . 1144
8.1168 npack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference . . . 1144
8.1168.1 Detailed Description . . . 1144
8.1168.2 Field Documentation . . . 1145
8.1168.2.1 imgType . . . 1145
8.1168.2.2 logString . . . 1145
8.1168.2.3 refData . . . 1146
8.1168.2.4 refString . . . 1146
8.1168.2.5 ResCode . . . 1146

8.1168.2.6 Tlvresult . . . 1146
8.1169 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference . . . 1146
8.1169.1 Detailed Description . . . 1146
8.1169.2 Field Documentation . . . 1146
8.1169.2.1 manString . . . 1146
8.1169.2.2 modelString . . . 1147
8.1169.2.3 plasmaIDString . . . 1147
8.1169.2.4 swVerString . . . 1147
8.1169.2.5 Tlvresult . . . 1147
8.1170 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference . . . 1147
8.1170.1 Detailed Description . . . 1147
8.1170.2 Field Documentation . . . 1147
8.1170.2.1 nameString . . . 1147
8.1170.2.2 Tlvresult . . . 1147
8.1170.2.3 versionString . . . 1147
8.1171 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference . . . 1147
8.1171.1 Detailed Description . . . 1147
8.1171.2 Field Documentation . . . 1148
8.1171.2.1 meidString . . . 1148
8.1171.2.2 Tlvresult . . . 1148
8.1172 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 1148
8.1172.1 Detailed Description . . . 1148
8.1172.2 Field Documentation . . . 1148
8.1172.2.1 Tlvresult . . . 1148
8.1173 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 1148
8.1173.1 Detailed Description . . . 1148
8.1173.2 Field Documentation . . . 1149
8.1173.2.1 Tlvresult . . . 1149
8.1174 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 1149
8.1174.1 Detailed Description . . . 1149
8.1174.2 Field Documentation . . . 1149
8.1174.2.1 Tlvresult . . . 1149
8.1175 unpack_dms_SLQSUIMGetState_t Struct Reference . . . 1149
8.1175.1 Detailed Description . . . 1149
8.1175.2 Field Documentation . . . 1149
8.1175.2.1 state . . . 1149
8.1175.2.2 Tlvresult . . . 1149
8.1176 unpack_dms_SwiEventReportCallBack_ind_t Struct Reference . . . 1149
8.1176.1 Detailed Description . . . 1150
8.1176.2 Field Documentation . . . 1150

8.1176.2.1 TempTlv . . . 1150
8.1176.2.2 VoltTlv . . . 1150
8.1177 unpack_dms_SwiSetEventReport_t Struct Reference . . . 1150
8.1177.1 Detailed Description . . . 1150
8.1177.2 Field Documentation . . . 1150
8.1177.2.1 Tlvresult . . . 1150
8.1178 unpack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 1150
8.1178.1 Detailed Description . . . 1151
8.1178.2 Field Documentation . . . 1151
8.1178.2.1 facilityState . . . 1151
8.1178.2.2 Tlvresult . . . 1151
8.1178.2.3 unblockRetriesLeft . . . 1151
8.1178.2.4 verifyRetriesLeft . . . 1151
8.1179 unpack_dms_UIMGetICCID_t Struct Reference . . . 1151
8.1179.1 Detailed Description . . . 1152
8.1179.2 Field Documentation . . . 1152
8.1179.2.1 String . . . 1152
8.1179.2.2 stringSize . . . 1152
8.1179.2.3 Tlvresult . . . 1152
8.1180 unpack_dms_UIMGetPINStatus_t Struct Reference . . . 1152
8.1180.1 Detailed Description . . . 1152
8.1180.2 Field Documentation . . . 1153
8.1180.2.1 p1Status . . . 1153
8.1180.2.2 p1UnblockRetriesLeft . . . 1153
8.1180.2.3 p1VerifyRetriesLeft . . . 1153
8.1180.2.4 p2Status . . . 1153
8.1180.2.5 p2UnblockRetriesLeft . . . 1153
8.1180.2.6 p2VerifyRetriesLeft . . . 1153
8.1180.2.7 Tlvresult . . . 1154
8.1181 unpack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 1154
8.1181.1 Detailed Description . . . 1154
8.1181.2 Field Documentation . . . 1154
8.1181.2.1 Tlvresult . . . 1154
8.1181.2.2 verifyRetriesLeft . . . 1154
8.1182 unpack_dms_UIMSetPINProtection_t Struct Reference . . . 1154
8.1182.1 Detailed Description . . . 1154
8.1182.2 Field Documentation . . . 1155
8.1182.2.1 Tlvresult . . . 1155
8.1182.2.2 unblockRetriesLeft . . . 1155
8.1182.2.3 verifyRetriesLeft . . . 1155

8.1182 npack_dms_UIMUnblockControlKey_t Struct Reference . . . 1155
8.1183.1 Detailed Description . . . 1155
8.1183.2 Field Documentation . . . 1155
8.1183.2.1 tlvresult . . . 1155
8.1183.2.2 unblockRetriesLeft . . . 1155
8.1184 npack_fms_GetImagesPreference_t Struct Reference . . . 1155
8.1184.1 Detailed Description . . . 1155
8.1184.2 Field Documentation . . . 1156
8.1184.2.1 imageListSize . . . 1156
8.1184.2.2 pImageList . . . 1156
8.1184.2.3 tlvresult . . . 1156
8.1185 npack_fms_GetStoredImages_t Struct Reference . . . 1156
8.1185.1 Detailed Description . . . 1156
8.1185.2 Field Documentation . . . 1156
8.1185.2.1 imageList . . . 1156
8.1185.2.2 imagelistSize . . . 1156
8.1185.2.3 tlvresult . . . 1156
8.1186 npack_fms_SetImagesPreference_t Struct Reference . . . 1157
8.1186.1 Detailed Description . . . 1157
8.1186.2 Field Documentation . . . 1157
8.1186.2.1 imageTypes . . . 1157
8.1186.2.2 imageTypesSize . . . 1157
8.1186.2.3 tlvresult . . . 1157
8.1187 npack_loc_BestAvailPos_Ind_t Struct Reference . . . 1157
8.1187.1 Detailed Description . . . 1158
8.1187.2 Field Documentation . . . 1162
8.1187.2.1 pAltitudeWrtEllipsoid . . . 1162
8.1187.2.2 pAltitudeWrtMeanSeaLevel . . . 1162
8.1187.2.3 pGpsTime . . . 1162
8.1187.2.4 pHeading . . . 1162
8.1187.2.5 pHeadingUnc . . . 1162
8.1187.2.6 pHorCirConf . . . 1162
8.1187.2.7 pHorEllpConf . . . 1162
8.1187.2.8 pHorReliability . . . 1162
8.1187.2.9 pHorUncCircular . . . 1162
8.1187.2.10 pHorUncEllipseOrientAzimuth . . . 1162
8.1187.2.11 pHorUncEllipseSemiMajor . . . 1162
8.1187.2.12 pHorUncEllipseSemiMinor . . . 1162
8.1187.2.13 pLatitude . . . 1162
8.1187.2.14 pLongitude . . . 1162

8.1187.2.15 MagneticDeviation . . . 1162
8.1187.2.16 PrecisionDilution . . . 1162
8.1187.2.17 SensorDataUsage . . . 1162
8.1187.2.18 SpeedHorizontal . . . 1162
8.1187.2.19 SpeedUnc . . . 1162
8.1187.2.20 SpeedVertical . . . 1162
8.1187.2.21 SpeedVerticalUnc . . . 1162
8.1187.2.22 SvUsedforFix . . . 1162
8.1187.2.23 TechnologyMask . . . 1162
8.1187.2.24 TimeSrc . . . 1162
8.1187.2.25 TimestampUtc . . . 1162
8.1187.2.26 TimeUnc . . . 1162
8.1187.2.27 VertConfidence . . . 1162
8.1187.2.28 VertReliability . . . 1163
8.1187.2.29 VertUnc . . . 1163
8.1187.2.30 Xid . . . 1163
8.1187.2.31 status . . . 1163
8.1187.2.32 Tlvresult . . . 1163
8.1188 unpack_loc_CradleMountCallback_Ind_t Struct Reference . . . 1163
8.1188.1 Detailed Description . . . 1163
8.1188.2 Field Documentation . . . 1163
8.1188.2.1 cradleMountConfigStatus . . . 1163
8.1189 unpack_loc_Delete_Assist_Data_t Struct Reference . . . 1163
8.1189.1 Detailed Description . . . 1163
8.1189.2 Field Documentation . . . 1164
8.1189.2.1 Tlvresult . . . 1164
8.1190 unpack_loc_DeleteAssistData_Ind_t Struct Reference . . . 1164
8.1190.1 Detailed Description . . . 1164
8.1190.2 Field Documentation . . . 1164
8.1190.2.1 status . . . 1164
8.1190.2.2 Tlvresult . . . 1165
8.1191 unpack_loc_EngineState_Ind_t Struct Reference . . . 1165
8.1191.1 Detailed Description . . . 1165
8.1191.2 Field Documentation . . . 1165
8.1191.2.1 engineState . . . 1165
8.1191.2.2 Tlvresult . . . 1165
8.1192 unpack_loc_EventNMEA_Ind_t Struct Reference . . . 1165
8.1192.1 Detailed Description . . . 1165
8.1192.2 Field Documentation . . . 1166
8.1192.2.1 NMEAData . . . 1166

8.1192.2.2 Tlvresult . . . 1166
8.1193 qmi_loc_EventRegister_t Struct Reference . . . 1166
8.1193.1 Detailed Description . . . 1166
8.1193.2 Field Documentation . . . 1166
8.1193.2.1 Tlvresult . . . 1166
8.1194 qmi_loc_EventTimeSyncCallback_Ind_t Struct Reference . . . 1166
8.1194.1 Detailed Description . . . 1166
8.1194.2 Field Documentation . . . 1166
8.1194.2.1 timeSyncRefCounter . . . 1166
8.1195 qmi_loc_GetServer_Ind_t Struct Reference . . . 1166
8.1195.1 Detailed Description . . . 1167
8.1195.2 Field Documentation . . . 1168
8.1195.2.1 pIPv4AddrInfo . . . 1168
8.1195.2.2 pIPv6AddrInfo . . . 1168
8.1195.2.3 pURL . . . 1168
8.1195.2.4 serverStatus . . . 1168
8.1195.2.5 serverType . . . 1168
8.1195.2.6 Tlvresult . . . 1168
8.1196 qmi_loc_GnssSvInfo_Ind_t Struct Reference . . . 1168
8.1196.1 Detailed Description . . . 1168
8.1196.2 Field Documentation . . . 1168
8.1196.2.1 altitudeAssumed . . . 1168
8.1196.2.2 pSatelliteInfo . . . 1168
8.1196.2.3 Tlvresult . . . 1168
8.1197 qmi_loc_InjectPositionCallback_Ind_t Struct Reference . . . 1168
8.1197.1 Detailed Description . . . 1169
8.1197.2 Field Documentation . . . 1169
8.1197.2.1 status . . . 1169
8.1198 qmi_loc_InjectSensorDataCallback_Ind_t Struct Reference . . . 1169
8.1198.1 Detailed Description . . . 1169
8.1198.2 Field Documentation . . . 1170
8.1198.2.1 injectSensorDataStatus . . . 1170
8.1198.2.2 pAccelSamplesAccepted . . . 1170
8.1198.2.3 pAccelTempSamplesAccepted . . . 1170
8.1198.2.4 pGyroSamplesAccepted . . . 1170
8.1198.2.5 pGyroTempSamplesAccepted . . . 1170
8.1198.2.6 pOpaqueIdentifier . . . 1171
8.1199 qmi_loc_InjectTimeSyncDataCallback_Ind_t Struct Reference . . . 1171
8.1199.1 Detailed Description . . . 1171
8.1199.2 Field Documentation . . . 1171

8.1199.2.1 injectTimeSyncStatus . . . 1171
8.1200 unpack_loc_InjectUTCTimeCallback_Ind_t Struct Reference . . . 1171
8.1200.1 Detailed Description . . . 1171
8.1200.2 Field Documentation . . . 1172
8.1200.2.1 status . . . 1172
8.1201 unpack_loc_PositionRpt_Ind_t Struct Reference . . . 1172
8.1201.1 Detailed Description . . . 1173
8.1201.2 Field Documentation . . . 1177
8.1201.2.1 pAltitudeAssumed . . . 1177
8.1201.2.2 pAltitudeWrtEllipsoid . . . 1177
8.1201.2.3 pAltitudeWrtMeanSeaLevel . . . 1178
8.1201.2.4 pFixId . . . 1178
8.1201.2.5 pGpsTime . . . 1178
8.1201.2.6 pHeading . . . 1178
8.1201.2.7 pHeadingUnc . . . 1178
8.1201.2.8 pHorConfidence . . . 1178
8.1201.2.9 pHorReliability . . . 1178
8.1201.2.10 pHorUncCircular . . . 1178
8.1201.2.11 pHorUncEllipseOrientAzimuth . . . 1178
8.1201.2.12 pHorUncEllipseSemiMajor . . . 1178
8.1201.2.13 pHorUncEllipseSemiMinor . . . 1178
8.1201.2.14 pLatitude . . . 1178
8.1201.2.15 pLeapSeconds . . . 1178
8.1201.2.16 pLongitude . . . 1178
8.1201.2.17 pMagneticDeviation . . . 1178
8.1201.2.18 pPrecisionDilution . . . 1178
8.1201.2.19 pSensorDataUsage . . . 1178
8.1201.2.20 pSpeedHorizontal . . . 1178
8.1201.2.21 pSpeedUnc . . . 1178
8.1201.2.22 pSpeedVertical . . . 1178
8.1201.2.23 pSvUsedforFix . . . 1178
8.1201.2.24 pTechnologyMask . . . 1178
8.1201.2.25 pTimeSrc . . . 1178
8.1201.2.26 pTimestampUtc . . . 1178
8.1201.2.27 pTimeUnc . . . 1178
8.1201.2.28 pVertConfidence . . . 1178
8.1201.2.29 pVertReliability . . . 1178
8.1201.2.30 pVertUnc . . . 1178
8.1201.2.31 sessionId . . . 1179
8.1201.2.32 sessionStatus . . . 1179

8.1201.2.3 tlvresult . . . 1179
8.1202 qmi_loc_SensorStreamingCallback_Ind_t Struct Reference . . . 1179
8.1202.1 Detailed Description . . . 1179
8.1202.2 Field Documentation . . . 1179
8.1202.2.1 pAccelAcceptReady . . . 1179
8.1202.2.2 pAccelTempAcceptReady . . . 1179
8.1202.2.3 pGyroAcceptReady . . . 1179
8.1202.2.4 pGyroTempAcceptReady . . . 1179
8.1203 qmi_loc_SetExtPowerConfig_Ind_t Struct Reference . . . 1179
8.1203.1 Detailed Description . . . 1180
8.1203.2 Field Documentation . . . 1180
8.1203.2.1 status . . . 1180
8.1203.2.2 tlvresult . . . 1180
8.1204 qmi_loc_SetExtPowerState_t Struct Reference . . . 1180
8.1204.1 Detailed Description . . . 1180
8.1204.2 Field Documentation . . . 1180
8.1204.2.1 tlvresult . . . 1181
8.1205 qmi_loc_SetOperationMode_Ind_t Struct Reference . . . 1181
8.1205.1 Detailed Description . . . 1181
8.1205.2 Field Documentation . . . 1181
8.1205.2.1 status . . . 1181
8.1205.2.2 tlvresult . . . 1181
8.1206 qmi_loc_SetOperationMode_t Struct Reference . . . 1181
8.1206.1 Detailed Description . . . 1181
8.1206.2 Field Documentation . . . 1182
8.1206.2.1 tlvresult . . . 1182
8.1207 qmi_loc_SetServer_Ind_t Struct Reference . . . 1182
8.1207.1 Detailed Description . . . 1182
8.1207.2 Field Documentation . . . 1182
8.1207.2.1 serverStatus . . . 1182
8.1207.2.2 tlvresult . . . 1182
8.1208 qmi_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 1183
8.1208.1 Detailed Description . . . 1183
8.1208.2 Field Documentation . . . 1183
8.1208.2.1 tlvresult . . . 1183
8.1209 qmi_loc_Start_t Struct Reference . . . 1183
8.1209.1 Detailed Description . . . 1183
8.1209.2 Field Documentation . . . 1183
8.1209.2.1 tlvresult . . . 1183
8.1210 qmi_loc_Stop_t Struct Reference . . . 1183

8.1210.1 Detailed Description . . . 1183
8.1210.2 Field Documentation . . . 1184
8.1210.2.1 TlvResult . . . 1184
8.1211 unpack_nas_GetCDMANetworkParameters_t Struct Reference . . . 1184
8.1211.1 Detailed Description . . . 1184
8.1211.2 Field Documentation . . . 1186
8.1211.2.1 Application . . . 1186
8.1211.2.2 Broadcast . . . 1186
8.1211.2.3 CustomSCP . . . 1186
8.1211.2.4 ForceRev0 . . . 1186
8.1211.2.5 Protocol . . . 1186
8.1211.2.6 RegForeignNID . . . 1186
8.1211.2.7 RegForeignSID . . . 1186
8.1211.2.8 RegHomeSID . . . 1186
8.1211.2.9 Roaming . . . 1186
8.1211.2.10 SCI . . . 1186
8.1211.2.11 SCM . . . 1186
8.1212 unpack_nas_GetHomeNetwork3GPP2_t Struct Reference . . . 1186
8.1212.1 Detailed Description . . . 1186
8.1212.2 Field Documentation . . . 1188
8.1212.2.1 nameSize . . . 1188
8.1212.2.2 pMCC . . . 1188
8.1212.2.3 pMNC . . . 1188
8.1212.2.4 pName . . . 1188
8.1212.2.5 pNID . . . 1188
8.1212.2.6 pNw2DescDisp . . . 1188
8.1212.2.7 pNw2DescEnc . . . 1188
8.1212.2.8 pNw2DescLen . . . 1188
8.1212.2.9 pNw2MCC . . . 1188
8.1212.2.10 pNw2MNC . . . 1188
8.1212.2.11 pNw2Name . . . 1188
8.1212.2.12 pSID . . . 1188
8.1213 unpack_nas_GetHomeNetwork_t Struct Reference . . . 1188
8.1213.1 Detailed Description . . . 1188
8.1213.2 Field Documentation . . . 1189
8.1213.2.1 mcc . . . 1189
8.1213.2.2 mnc . . . 1189
8.1213.2.3 name . . . 1189
8.1213.2.4 nid . . . 1189
8.1213.2.5 sid . . . 1189

8.1214 nnpack_nas_GetNetworkPreference_t Struct Reference . . . 1189
8.1214.1 Detailed Description . . . 1189
8.1214.2 Field Documentation . . . 1190
8.1214.2.1 ActiveTechPref . . . 1190
8.1214.2.2 Duration . . . 1190
8.1214.2.3 PersistentTechPref . . . 1190
8.1214.2.4 Tlvresult . . . 1190
8.1215 nnpack_nas_GetRFInfo_t Struct Reference . . . 1190
8.1215.1 Detailed Description . . . 1191
8.1215.2 Field Documentation . . . 1191
8.1215.2.1 instancesSize . . . 1191
8.1215.2.2 RFBandInfoElements . . . 1191
8.1216 nnpack_nas_GetServingNetwork_t Struct Reference . . . 1191
8.1216.1 Detailed Description . . . 1191
8.1216.2 Field Documentation . . . 1193
8.1216.2.1 CSDomain . . . 1193
8.1216.2.2 DataCaps . . . 1193
8.1216.2.3 DataCapsLen . . . 1193
8.1216.2.4 MCC . . . 1193
8.1216.2.5 MNC . . . 1193
8.1216.2.6 Name . . . 1193
8.1216.2.7 nameSize . . . 1193
8.1216.2.8 PSDomain . . . 1193
8.1216.2.9 RadioIfaces . . . 1193
8.1216.2.10 RadioIfacesSize . . . 1193
8.1216.2.11 RAN . . . 1193
8.1216.2.12 RegistrationState . . . 1193
8.1216.2.13 Roaming . . . 1193
8.1217 nnpack_nas_GetServingNetworkCapabilities_t Struct Reference . . . 1193
8.1217.1 Detailed Description . . . 1194
8.1217.2 Field Documentation . . . 1194
8.1217.2.1 DataCaps . . . 1194
8.1217.2.2 DataCapsLen . . . 1194
8.1218 nnpack_nas_GetSignalStrengths_t Struct Reference . . . 1194
8.1218.1 Detailed Description . . . 1194
8.1218.2 Field Documentation . . . 1194
8.1218.2.1 len . . . 1195
8.1218.2.2 radio . . . 1195
8.1218.2.3 ssi . . . 1195
8.1219 nnpack_nas_PerformNetworkScan_t Struct Reference . . . 1195

8.1219.1 Detailed Description . . . 1195
8.1219.2 Field Documentation . . . 1196
8.1219.2.1 p3GppNetworkInfoInstances . . . 1196
8.1219.2.2 p3GppNetworkInstanceSize . . . 1196
8.1219.2.3 pPCSInstance . . . 1196
8.1219.2.4 pPCSInstanceSize . . . 1196
8.1219.2.5 pRATINstance . . . 1196
8.1219.2.6 pRATInstanceSize . . . 1196
8.1219.2.7 pScanResult . . . 1196
8.1220 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference . . . 1196
8.1220.1 Detailed Description . . . 1196
8.1220.2 Field Documentation . . . 1196
8.1220.2.1 dataCaps . . . 1196
8.1220.2.2 dataCapsSize . . . 1196
8.1221 unpack_nas_SetEventReportInd_t Struct Reference . . . 1196
8.1221.1 Detailed Description . . . 1196
8.1221.2 Field Documentation . . . 1197
8.1221.2.1 RFTlv . . . 1197
8.1221.2.2 RRTlv . . . 1197
8.1221.2.3 SLQSSSTlv . . . 1197
8.1221.2.4 SSTlv . . . 1197
8.1222 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference . . . 1197
8.1222.1 Detailed Description . . . 1197
8.1222.2 Field Documentation . . . 1197
8.1222.2.1 sPhyCaAggPcellInfo . . . 1197
8.1222.2.2 sPhyCaAggScellDlBw . . . 1198
8.1222.2.3 sPhyCaAggScellIndex . . . 1198
8.1222.2.4 sPhyCaAggScellIndType . . . 1198
8.1222.2.5 sPhyCaAggScellInfo . . . 1198
8.1223 unpack_nas_SetNetworkPreference_t Struct Reference . . . 1198
8.1223.1 Detailed Description . . . 1198
8.1223.2 Field Documentation . . . 1198
8.1223.2.1 Tlvresult . . . 1198
8.1224 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . 1198
8.1224.1 Detailed Description . . . 1198
8.1224.2 Field Documentation . . . 1198
8.1224.2.1 roaming . . . 1198
8.1225 unpack_nas_SetServingSystemCallback_ind_t Struct Reference . . . 1199
8.1225.1 Detailed Description . . . 1199
8.1225.2 Field Documentation . . . 1199

8.1225.2.1 SSInfo . . . 1199
8.1225.2.2 Tlvresult . . . 1199
8.1226 npack_nas_SLQSGetErrorRate_t Struct Reference . . . 1199
8.1226.1 Detailed Description . . . 1199
8.1226.2 Field Documentation . . . 1200
8.1226.2.1 pCDMAFrameErrRate . . . 1200
8.1226.2.2 pGSMBER . . . 1200
8.1226.2.3 pHDRPackErrRate . . . 1200
8.1226.2.4 pWCDMABER . . . 1200
8.1227 npack_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . 1200
8.1227.1 Detailed Description . . . 1200
8.1227.2 Field Documentation . . . 1201
8.1227.2.1 LTECphyCaInfo . . . 1201
8.1227.2.2 Tlvresult . . . 1201
8.1228 npack_nas_SLQSGetNetworkTime_t Struct Reference . . . 1201
8.1228.1 Detailed Description . . . 1201
8.1228.2 Field Documentation . . . 1201
8.1228.2.1 p3GPP2TimeInfo . . . 1201
8.1228.2.2 p3GPPTimeInfo . . . 1201
8.1229 npack_nas_SLQSGetOperatorNameData_t Struct Reference . . . 1201
8.1229.1 Detailed Description . . . 1201
8.1229.2 Field Documentation . . . 1202
8.1229.2.1 pNITZInformation . . . 1202
8.1229.2.2 pOperatorNameString . . . 1202
8.1229.2.3 pOperatorPLMNList . . . 1202
8.1229.2.4 pPLMNNetworkName . . . 1202
8.1229.2.5 pSrvcProviderName . . . 1202
8.1230 npack_nas_SLQSGetPLMNName_t Struct Reference . . . 1202
8.1230.1 Detailed Description . . . 1203
8.1230.2 Field Documentation . . . 1204
8.1230.2.1 longName . . . 1204
8.1230.2.2 longNameCI . . . 1204
8.1230.2.3 longNameEn . . . 1204
8.1230.2.4 longNameLen . . . 1204
8.1230.2.5 longNameSB . . . 1204
8.1230.2.6 shortName . . . 1204
8.1230.2.7 shortNameCI . . . 1205
8.1230.2.8 shortNameEn . . . 1205
8.1230.2.9 shortNameLen . . . 1205
8.1230.2.10 shortNameSB . . . 1205

8.1230.2.1 spn . . . 1205
8.1230.2.2 spnEncoding . . . 1205
8.1230.2.3 spnLength . . . 1205
8.1231 unpack_nas_SLQSGetServingSystem_t Struct Reference . . . 1205
8.1231.1 Detailed Description . . . 1205
8.1231.2 Field Documentation . . . 1208
8.1231.2.1 BasestationID . . . 1208
8.1231.2.2 BasestationLatitude . . . 1208
8.1231.2.3 BasestationLongitude . . . 1208
8.1231.2.4 CallBarStatus . . . 1208
8.1231.2.5 CDMA_P_Rev . . . 1208
8.1231.2.6 CDMASystemInfoExt . . . 1208
8.1231.2.7 CellID . . . 1208
8.1231.2.8 ConcSvcInfo . . . 1208
8.1231.2.9 CurrentPLMN . . . 1208
8.1231.2.10 DataSrvCapabilities . . . 1208
8.1231.2.11 DefaultRoamInd . . . 1208
8.1231.2.12 DetailedSvcInfo . . . 1208
8.1231.2.13 DTMInd . . . 1208
8.1231.2.14 Gpp2TimeZone . . . 1208
8.1231.2.15 GppNetworkDSTAdjustment . . . 1208
8.1231.2.16 GppTimeZone . . . 1208
8.1231.2.17 HdrPersonality . . . 1208
8.1231.2.18 Hdrac . . . 1208
8.1231.2.19 NetworkID . . . 1209
8.1231.2.20 PRLInd . . . 1209
8.1231.2.21 RoamIndicatorVal . . . 1209
8.1231.2.22 RoamingIndicatorList . . . 1209
8.1231.2.23 ServingSystem . . . 1209
8.1231.2.24 SystemID . . . 1209
8.1231.2.25 TrackAreaCode . . . 1209
8.1232 unpack_nas_SLQSGetSignalStrength_t Struct Reference . . . 1209
8.1232.1 Detailed Description . . . 1209
8.1232.2 Field Documentation . . . 1211
8.1232.2.1 ecioList . . . 1211
8.1232.2.2 ecioListLen . . . 1211
8.1232.2.3 errorRateList . . . 1211
8.1232.2.4 errorRateListLen . . . 1211
8.1232.2.5 io . . . 1211
8.1232.2.6 tersrp . . . 1211

8.1232.2.7 lteSnr . . . 1211
8.1232.2.8 rsrqInfo . . . 1211
8.1232.2.9 rxSignalStrengthList . . . 1211
8.1232.2.10 rxSignalStrengthListLen . . . 1211
8.1232.2.11 signalStrengthReqMask . . . 1211
8.1232.2.12 snr . . . 1211
8.1233 qmi_nas_SLQSGetSysInfo_t Struct Reference . . . 1211
8.1233.1 Detailed Description . . . 1212
8.1233.2 Field Documentation . . . 1214
8.1233.2.1 pAddCDMASysInfo . . . 1214
8.1233.2.2 pAddGSMSysInfo . . . 1214
8.1233.2.3 pAddHDRSysInfo . . . 1214
8.1233.2.4 pAddLTESysInfo . . . 1214
8.1233.2.5 pAddWCDMASysInfo . . . 1214
8.1233.2.6 pCDMASrvStatusInfo . . . 1214
8.1233.2.7 pCDMASysInfo . . . 1214
8.1233.2.8 pGSMCallBarringSysInfo . . . 1214
8.1233.2.9 pGSMCipherDomainSysInfo . . . 1214
8.1233.2.10 pGSMSrvStatusInfo . . . 1214
8.1233.2.11 pGSMSysInfo . . . 1214
8.1233.2.12 pHDRSrvStatusInfo . . . 1214
8.1233.2.13 pHDRSysInfo . . . 1214
8.1233.2.14 pLTESrvStatusInfo . . . 1214
8.1233.2.15 pLTESysInfo . . . 1214
8.1233.2.16 pLTEVoiceSupportSysInfo . . . 1214
8.1233.2.17 pWCDMACallBarringSysInfo . . . 1214
8.1233.2.18 pWCDMACipherDomainSysInfo . . . 1215
8.1233.2.19 pWCDMASrvStatusInfo . . . 1215
8.1233.2.20 pWCDMASysInfo . . . 1215
8.1234 qmi_nas_SLQSGetSysSelectionPref_t Struct Reference . . . 1215
8.1234.1 Detailed Description . . . 1215
8.1234.2 Field Documentation . . . 1219
8.1234.2.1 pBandPref . . . 1219
8.1234.2.2 pEmerMode . . . 1219
8.1234.2.3 pGWAcqOrderPref . . . 1219
8.1234.2.4 pLTEBandPref . . . 1219
8.1234.2.5 pModePref . . . 1219
8.1234.2.6 pNetSelPref . . . 1219
8.1234.2.7 pPRLPref . . . 1219
8.1234.2.8 pRoamPref . . . 1219

8.1234.2.9 pSrvDomainPref . . . 1219
8.1235 qmi_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 1219
8.1235.1 Detailed Description . . . 1220
8.1235.2 Field Documentation . . . 1220
8.1235.2.1 pCDMAChannel . . . 1220
8.1235.2.2 pDirNum . . . 1220
8.1235.2.3 pHomeSIDNID . . . 1220
8.1235.2.4 pMinBasedIMSI . . . 1220
8.1235.2.5 pNAMNameInfo . . . 1220
8.1235.2.6 pTrueIMSI . . . 1220
8.1236 qmi_nas_SLQSNasGetCellLocationInfo_t Struct Reference . . . 1220
8.1236.1 Detailed Description . . . 1221
8.1236.2 Field Documentation . . . 1221
8.1236.2.1 pCDMAInfo . . . 1221
8.1236.2.2 pGERANInfo . . . 1222
8.1236.2.3 pLTEInfoInterfreq . . . 1222
8.1236.2.4 pLTEInfoIntrafreq . . . 1222
8.1236.2.5 pLTEInfoNeighboringGSM . . . 1222
8.1236.2.6 pLTEInfoNeighboringWCDMA . . . 1222
8.1236.2.7 pUMTSCellID . . . 1222
8.1236.2.8 pUMTSInfo . . . 1222
8.1236.2.9 pWCDMAInfoLTENeighborCell . . . 1222
8.1237 qmi_nas_SLQSNASGeteDRXParams_t Struct Reference . . . 1222
8.1237.1 Detailed Description . . . 1222
8.1237.2 Field Documentation . . . 1222
8.1237.2.1 pCycleLen . . . 1222
8.1237.2.2 pEdrxEnable . . . 1222
8.1237.2.3 pPagingTimeWindow . . . 1223
8.1238 qmi_nas_SLQSNasGetHDRColorCode_t Struct Reference . . . 1223
8.1238.1 Detailed Description . . . 1223
8.1238.2 Field Documentation . . . 1223
8.1238.2.1 pColorCode . . . 1223
8.1239 qmi_nas_SLQSNasGetRFInfo_t Struct Reference . . . 1223
8.1239.1 Detailed Description . . . 1223
8.1239.2 Field Documentation . . . 1224
8.1239.2.1 pLTEOperationMode . . . 1224
8.1239.2.2 pRfBandInfoExtFormat . . . 1224
8.1239.2.3 pRfBandwidthInfo . . . 1224
8.1239.2.4 pRfDedicatedBandInfo . . . 1224
8.1239.2.5 rfbandInfoList . . . 1224

8.1239.2.6 Tlvresult . . . 1224
8.1240 unpack_nas_SLQSNasGetSigInfo_t Struct Reference . . . 1224
8.1240.1 Detailed Description . . . 1224
8.1240.2 Field Documentation . . . 1225
8.1240.2.1 CDMASSInfo . . . 1225
8.1240.2.2 GSMSSInfo . . . 1225
8.1240.2.3 HDRSSInfo . . . 1225
8.1240.2.4 LTESSInfo . . . 1225
8.1240.2.5 WCDMASSInfo . . . 1225
8.1241 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 1225
8.1241.1 Detailed Description . . . 1225
8.1241.2 Field Documentation . . . 1225
8.1241.2.1 pRXChain0Info . . . 1225
8.1241.2.2 pRXChain1Info . . . 1225
8.1241.2.3 pTXInfo . . . 1225
8.1242 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . 1225
8.1242.1 Detailed Description . . . 1226
8.1242.2 Field Documentation . . . 1226
8.1242.2.1 pDayltSavAdj . . . 1226
8.1242.2.2 pRadioInterface . . . 1226
8.1242.2.3 pTimeZone . . . 1226
8.1242.2.4 universalTime . . . 1226
8.1243 unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . 1226
8.1243.1 Detailed Description . . . 1227
8.1243.2 Field Documentation . . . 1227
8.1243.2.1 pCDMASigInfo . . . 1227
8.1243.2.2 pGSMSigInfo . . . 1227
8.1243.2.3 pHDRSigInfo . . . 1227
8.1243.2.4 pLTESigInfo . . . 1227
8.1243.2.5 pRscp . . . 1227
8.1243.2.6 pTDSCDMASigInfoExt . . . 1228
8.1243.2.7 pWCDMASigInfo . . . 1228
8.1244 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference . . . 1228
8.1244.1 Detailed Description . . . 1228
8.1244.2 Field Documentation . . . 1228
8.1244.2.1 pLteEARFCN . . . 1228
8.1244.2.2 pLtePCI . . . 1228
8.1244.2.3 pWcdmaUARFCN . . . 1228
8.1245 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference . . . 1228
8.1245.1 Detailed Description . . . 1228

8.1245.2 Field Documentation . . . 1229
8.1245.2.1 commonInfo . . . 1229
8.1245.2.2 pLTEInfo . . . 1229
8.1246 npack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . 1229
8.1246.1 Detailed Description . . . 1229
8.1246.2 Field Documentation . . . 1229
8.1246.2.1 Info . . . 1229
8.1246.2.2 Tlvresult . . . 1229
8.1247 npack_nas_SLQSNasTimerCallback_ind_t Struct Reference . . . 1229
8.1247.1 Detailed Description . . . 1230
8.1247.2 Field Documentation . . . 1230
8.1247.2.1 t3396_apn . . . 1230
8.1247.2.2 t3396_plmn_id . . . 1230
8.1247.2.3 t3396_val . . . 1230
8.1248 npack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . 1230
8.1248.1 Detailed Description . . . 1230
8.1248.2 Field Documentation . . . 1230
8.1248.2.1 Info . . . 1230
8.1248.2.2 Tlvresult . . . 1230
8.1249 npack_nas_SLQSSwiGetHDRPersonality_t Struct Reference . . . 1230
8.1249.1 Detailed Description . . . 1231
8.1249.2 Field Documentation . . . 1231
8.1249.2.1 pCurrentPersonality . . . 1231
8.1249.2.2 pPersonalityListLength . . . 1231
8.1249.2.3 pProtocolSubtypeElement . . . 1231
8.1250 npack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference . . . 1231
8.1250.1 Detailed Description . . . 1231
8.1250.2 Field Documentation . . . 1232
8.1250.2.1 pAppSubType . . . 1232
8.1250.2.2 pCurrentPrsnlty . . . 1232
8.1250.2.3 pPersonalityListLength . . . 1232
8.1250.2.4 pProtoSubTypElmnt . . . 1232
8.1251 npack_nas_SLQSSwiGetHRPDStats_t Struct Reference . . . 1232
8.1251.1 Detailed Description . . . 1232
8.1251.2 Field Documentation . . . 1233
8.1251.2.1 pDRCParams . . . 1233
8.1251.2.2 pPilotSetData . . . 1233
8.1251.2.3 pUATI . . . 1233
8.1252 npack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . 1233
8.1252.1 Detailed Description . . . 1233

8.1252.2 Field Documentation . . . 1233
8.1252.2.1 CQIValueCW0 . . . 1233
8.1252.2.2 CQIValueCW1 . . . 1233
8.1252.2.3 ValidityCW0 . . . 1233
8.1252.2.4 ValidityCW1 . . . 1233
8.1253 qmi_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference . . . 1234
8.1253.1 Detailed Description . . . 1234
8.1253.2 Field Documentation . . . 1234
8.1253.2.1 pSccRxInfo . . . 1234
8.1254 qmi_nas_SLQSSwiHDRPersonalityCallback_Ind_t Struct Reference . . . 1234
8.1254.1 Detailed Description . . . 1234
8.1254.2 Field Documentation . . . 1234
8.1254.2.1 pHDRPersInd . . . 1234
8.1255 qmi_nas_SLQSSwiNetworkDebug_t Struct Reference . . . 1234
8.1255.1 Detailed Description . . . 1235
8.1255.2 Field Documentation . . . 1235
8.1255.2.1 pDataStatusDetail . . . 1235
8.1255.2.2 pDeviceConfigDetail . . . 1235
8.1255.2.3 pNetworkStat1x . . . 1235
8.1255.2.4 pNetworkStatEVDO . . . 1235
8.1255.2.5 pObjectVer . . . 1235
8.1256 qmi_nas_SLQSSwiRandIndicatorCallback_Ind_t Struct Reference . . . 1235
8.1256.1 Detailed Description . . . 1235
8.1256.2 Field Documentation . . . 1236
8.1256.2.1 rankIndicatorTlv . . . 1236
8.1257 qmi_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . 1236
8.1257.1 Detailed Description . . . 1236
8.1257.2 Field Documentation . . . 1238
8.1257.2.1 pAddCDMASysInfo . . . 1238
8.1257.2.2 pAddGSMSysInfo . . . 1238
8.1257.2.3 pAddHDRSysInfo . . . 1238
8.1257.2.4 pAddLTESysInfo . . . 1238
8.1257.2.5 pAddWCDMASysInfo . . . 1238
8.1257.2.6 pCDMASrvStatusInfo . . . 1238
8.1257.2.7 pCDMASysInfo . . . 1238
8.1257.2.8 pGSMCallBarringSysInfo . . . 1238
8.1257.2.9 pGSMCipherDomainSysInfo . . . 1238
8.1257.2.10 pGSMSrvStatusInfo . . . 1238
8.1257.2.11 pGSMSysInfo . . . 1238
8.1257.2.12 pHDRSrvStatusInfo . . . 1238

8.1257.2.13 HDRSysInfo . . . 1238
8.1257.2.14 LTESrvStatusInfo . . . 1238
8.1257.2.15 LTESysInfo . . . 1238
8.1257.2.16 LTEVoiceSupportSysInfo . . . 1239
8.1257.2.17 SysInfoNoChange . . . 1239
8.1257.2.18 WCDMACallBarringSysInfo . . . 1239
8.1257.2.19 WCDMACipherDomainSysInfo . . . 1239
8.1257.2.20 WCDMASrvStatusInfo . . . 1239
8.1257.2.21 WCDMASysInfo . . . 1239
8.1258 unpack_omaDmConfigTlv_t Struct Reference . . . 1239
8.1258.1 Detailed Description . . . 1239
8.1258.2 Field Documentation . . . 1239
8.1258.2.1 alertmsg . . . 1239
8.1258.2.2 alertmsglength . . . 1239
8.1258.2.3 state . . . 1240
8.1258.2.4 userInputReq . . . 1240
8.1258.2.5 userInputTimeout . . . 1240
8.1259 unpack_omaDmFotaTlv_t Struct Reference . . . 1240
8.1259.1 Detailed Description . . . 1240
8.1259.2 Field Documentation . . . 1241
8.1259.2.1 description . . . 1241
8.1259.2.2 descriptionlength . . . 1241
8.1259.2.3 fwdloadsize . . . 1241
8.1259.2.4 fwloadComplete . . . 1241
8.1259.2.5 namelength . . . 1241
8.1259.2.6 package_name . . . 1241
8.1259.2.7 sessionType . . . 1241
8.1259.2.8 severity . . . 1241
8.1259.2.9 state . . . 1241
8.1259.2.10 updateCompleteStatus . . . 1241
8.1259.2.11 userInputReq . . . 1242
8.1259.2.12 userInputTimeout . . . 1242
8.1259.2.13 version . . . 1242
8.1259.2.14 versionlength . . . 1242
8.1260 unpack_omaDmNotificationsTlv_t Struct Reference . . . 1242
8.1260.1 Detailed Description . . . 1242
8.1260.2 Field Documentation . . . 1242
8.1260.2.1 notification . . . 1242
8.1260.2.2 sessionStatus . . . 1242
8.1261 unpack_pds_GetPDSDefaults_t Struct Reference . . . 1243

8.1261.1 Detailed Description . . . 1243
8.1261.2 Field Documentation . . . 1243
8.1261.2.1 pAccuracy . . . 1243
8.1261.2.2 pInterval . . . 1243
8.1261.2.3 pOperation . . . 1243
8.1261.2.4 pTimeout . . . 1243
8.1262 qmi_pds_GetPDSState_t Struct Reference . . . 1243
8.1262.1 Detailed Description . . . 1244
8.1262.2 Field Documentation . . . 1244
8.1262.2.1 pEnabledStatus . . . 1244
8.1262.2.2 pTrackingStatus . . . 1244
8.1263 qmi_pds_GetPortAutomaticTracking_t Struct Reference . . . 1244
8.1263.1 Detailed Description . . . 1244
8.1263.2 Field Documentation . . . 1244
8.1263.2.1 pbAuto . . . 1244
8.1264 qmi_pds_GetServiceAutomaticTracking_t Struct Reference . . . 1245
8.1264.1 Detailed Description . . . 1245
8.1264.2 Field Documentation . . . 1245
8.1264.2.1 pbAuto . . . 1245
8.1265 qmi_pds_GetXTRAAutomaticDownload_t Struct Reference . . . 1245
8.1265.1 Detailed Description . . . 1245
8.1265.2 Field Documentation . . . 1245
8.1265.2.1 pbEnabled . . . 1245
8.1265.2.2 pInterval . . . 1246
8.1266 qmi_pds_GetXTRANetwork_t Struct Reference . . . 1246
8.1266.1 Detailed Description . . . 1246
8.1266.2 Field Documentation . . . 1246
8.1266.2.1 pPreference . . . 1246
8.1267 qmi_pds_GetXTRAValidity_t Struct Reference . . . 1246
8.1267.1 Detailed Description . . . 1246
8.1267.2 Field Documentation . . . 1247
8.1267.2.1 pDuration . . . 1247
8.1267.2.2 pGPSWeek . . . 1247
8.1267.2.3 pGPSWeekOffset . . . 1247
8.1268 qmi_pds_SetEventReport_Ind_t Struct Reference . . . 1247
8.1268.1 Detailed Description . . . 1247
8.1268.2 Field Documentation . . . 1248
8.1268.2.1 dLatitude . . . 1248
8.1268.2.2 dLongitude . . . 1248
8.1268.2.3 has_dLatitude . . . 1248

8.1268.2.4 has_dLongitude . . . 1248
8.1268.2.5 has_PositionDataNMEA . . . 1248
8.1268.2.6 has_posSrc . . . 1248
8.1268.2.7 has_SessionStatus . . . 1248
8.1268.2.8 PositionDataNMEA . . . 1248
8.1268.2.9 posSrc . . . 1248
8.1268.2.10 SessionStatus . . . 1248
8.1269 unpack_pds_SetPdsState_Ind_t Struct Reference . . . 1249
8.1269.1 Detailed Description . . . 1249
8.1269.2 Field Documentation . . . 1249
8.1269.2.1 EnabledStatus . . . 1249
8.1269.2.2 TlvPresent . . . 1249
8.1269.2.3 TrackingStatus . . . 1249
8.1270 unpack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 1249
8.1270.1 Detailed Description . . . 1249
8.1270.2 Field Documentation . . . 1250
8.1270.2.1 pServerAddress . . . 1250
8.1270.2.2 pServerPort . . . 1250
8.1270.2.3 pServerURL . . . 1250
8.1270.2.4 pServerURLLength . . . 1250
8.1271 unpack_pds_SLQSGetGPSStateInfo_t Struct Reference . . . 1250
8.1271.1 Detailed Description . . . 1251
8.1271.2 Field Documentation . . . 1253
8.1271.2.1 Altitude . . . 1253
8.1271.2.2 EngineState . . . 1253
8.1271.2.3 glo_almanac_sv_msk . . . 1253
8.1271.2.4 glo_ephemeris_sv_msk . . . 1253
8.1271.2.5 glo_health_sv_msk . . . 1253
8.1271.2.6 glo_visible_sv_msk . . . 1253
8.1271.2.7 gps_almanac_sv_msk . . . 1253
8.1271.2.8 gps_ephemeris_sv_msk . . . 1253
8.1271.2.9 gps_health_sv_msk . . . 1253
8.1271.2.10 gps_visible_sv_msk . . . 1253
8.1271.2.11 HorizontalUncertainty . . . 1253
8.1271.2.12 no_valid . . . 1253
8.1271.2.13 atitude . . . 1253
8.1271.2.14 ongitude . . . 1253
8.1271.2.15 bas_almanac_sv_msk . . . 1253
8.1271.2.16 bas_ephemeris_sv_msk . . . 1253
8.1271.2.17 bas_health_sv_msk . . . 1254

8.1271.2.18 sbas_visible_sv_msk . . . 1254
8.1271.2.19 time_uncert_ms . . . 1254
8.1271.2.20 timeStmp_gps_week . . . 1254
8.1271.2.21 timeStmp_tow_ms . . . 1254
8.1271.2.22 validMask . . . 1254
8.1271.2.23 verticalUncertainty . . . 1254
8.1271.2.24 wra_start_gps_minutes . . . 1254
8.1271.2.25 wra_start_gps_week . . . 1254
8.1271.2.26 wra_valid_duration_hours . . . 1254
8.1272 unpack_qmi_t Struct Reference . . . 1254
8.1272.1 Detailed Description . . . 1254
8.1272.2 Field Documentation . . . 1254
8.1272.2.1 msgid . . . 1254
8.1272.2.2 type . . . 1254
8.1272.2.3 xid . . . 1254
8.1273 unpack_qos_dataRate_t Struct Reference . . . 1254
8.1273.1 Detailed Description . . . 1255
8.1273.2 Field Documentation . . . 1255
8.1273.2.1 dataRateMax . . . 1255
8.1273.2.2 guaranteedRate . . . 1255
8.1274 unpack_qos_IPv4Addr_t Struct Reference . . . 1255
8.1274.1 Detailed Description . . . 1255
8.1274.2 Field Documentation . . . 1255
8.1274.2.1 addr . . . 1255
8.1274.2.2 subnetMask . . . 1255
8.1275 unpack_qos_IPv6Addr_t Struct Reference . . . 1255
8.1275.1 Detailed Description . . . 1255
8.1275.2 Field Documentation . . . 1256
8.1275.2.1 addr . . . 1256
8.1275.2.2 prefixLen . . . 1256
8.1276 unpack_qos_IPv6TrafCls_t Struct Reference . . . 1256
8.1276.1 Detailed Description . . . 1256
8.1276.2 Field Documentation . . . 1256
8.1276.2.1 mask . . . 1256
8.1276.2.2 val . . . 1256
8.1277 unpack_qos_pktErrRate_t Struct Reference . . . 1256
8.1277.1 Detailed Description . . . 1256
8.1277.2 Field Documentation . . . 1257
8.1277.2.1 exponent . . . 1257
8.1277.2.2 multiplier . . . 1257

8.1278 unpack_qos_Port_t Struct Reference . . . 1257
8.1278.1 Detailed Description . . . 1257
8.1278.2 Field Documentation . . . 1257
8.1278.2.1 port . . . 1257
8.1278.2.2 range . . . 1257
8.1279 unpack_qos_QosFlowInfo_t Struct Reference . . . 1257
8.1279.1 Detailed Description . . . 1257
8.1279.2 Field Documentation . . . 1258
8.1279.2.1 BearerID . . . 1258
8.1279.2.2 is_RxQFlowGranted_Available . . . 1258
8.1279.2.3 is_TxQFlowGranted_Available . . . 1258
8.1279.2.4 NumRxFilters . . . 1258
8.1279.2.5 NumTxFilters . . . 1258
8.1279.2.6 QFlowState . . . 1258
8.1279.2.7 RxQFilter . . . 1259
8.1279.2.8 RxQFlowGranted . . . 1259
8.1279.2.9 TxQFilter . . . 1259
8.1279.2.10 TxQFlowGranted . . . 1259
8.1280 unpack_qos_QosFlowInfoState_t Struct Reference . . . 1259
8.1280.1 Detailed Description . . . 1259
8.1280.2 Field Documentation . . . 1259
8.1280.2.1 id . . . 1259
8.1280.2.2 isNewFlow . . . 1259
8.1280.2.3 state . . . 1259
8.1281 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . 1259
8.1281.1 Detailed Description . . . 1260
8.1281.2 Field Documentation . . . 1260
8.1281.2.1 NWQoSStatus . . . 1260
8.1282 unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 1260
8.1282.1 Detailed Description . . . 1260
8.1282.2 Field Documentation . . . 1261
8.1282.2.1 ambr_dl . . . 1261
8.1282.2.2 ambr_dl_ext . . . 1261
8.1282.2.3 ambr_dl_ext2 . . . 1261
8.1282.2.4 ambr_ul . . . 1261
8.1282.2.5 ambr_ul_ext . . . 1261
8.1282.2.6 ambr_ul_ext2 . . . 1261
8.1282.2.7 apnId . . . 1261
8.1283 unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 1261
8.1283.1 Detailed Description . . . 1261

8.1283.2 Field Documentation . . . 1262
8.1283.2.1 apnId . . . 1262
8.1283.2.2 numQosFlow . . . 1262
8.1283.2.3 qosFlow . . . 1262
8.1283.2.4 total_rx_bytes . . . 1262
8.1283.2.5 total_rx_pkt . . . 1262
8.1283.2.6 total_tx_bytes . . . 1262
8.1283.2.7 total_tx_bytes_drp . . . 1262
8.1283.2.8 total_tx_pkt . . . 1262
8.1283.2.9 total_tx_pkt_drp . . . 1262
8.1284 qmi_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . 1262
8.1284.1 Detailed Description . . . 1263
8.1284.2 Field Documentation . . . 1263
8.1284.2.1 NumFlows . . . 1263
8.1284.2.2 QosFlowInfo . . . 1263
8.1285 qmi_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . 1263
8.1285.1 Detailed Description . . . 1263
8.1285.2 Field Documentation . . . 1263
8.1285.2.1 status . . . 1263
8.1286 qmi_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . 1263
8.1286.1 Detailed Description . . . 1264
8.1286.2 Field Documentation . . . 1264
8.1286.2.1 event . . . 1264
8.1287 qmi_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . 1264
8.1287.1 Detailed Description . . . 1264
8.1287.2 Field Documentation . . . 1265
8.1287.2.1 event . . . 1265
8.1287.2.2 id . . . 1265
8.1287.2.3 reason . . . 1265
8.1287.2.4 status . . . 1265
8.1288 qmi_qos_swiQosFilter_t Struct Reference . . . 1265
8.1288.1 Detailed Description . . . 1266
8.1288.2 Field Documentation . . . 1267
8.1288.2.1 EspSpi . . . 1267
8.1288.2.2 id . . . 1267
8.1288.2.3 index . . . 1267
8.1288.2.4 IPv4DstAddr . . . 1267
8.1288.2.5 IPv4SrcAddr . . . 1267
8.1288.2.6 IPv4Tos . . . 1267
8.1288.2.7 IPv6DstAddr . . . 1268

8.1288.2.8 IPv6Label . . . 1268
8.1288.2.9 IPv6SrcAddr . . . 1268
8.1288.2.10 IPv6TrafCls . . . 1268
8.1288.2.11 is_EspSpi_Available . . . 1268
8.1288.2.12 is_Id_Available . . . 1268
8.1288.2.13 is_IPv4DstAddr_Available . . . 1268
8.1288.2.14 is_IPv4SrcAddr_Available . . . 1268
8.1288.2.15 is_IPv4Tos_Available . . . 1268
8.1288.2.16 is_IPv6DstAddr_Available . . . 1268
8.1288.2.17 is_IPv6Label_Available . . . 1268
8.1288.2.18 is_IPv6SrcAddr_Available . . . 1268
8.1288.2.19 is_IPv6TrafCls_Available . . . 1268
8.1288.2.20 is_NxtHdrProto_Available . . . 1268
8.1288.2.21 is_Precedence_Available . . . 1268
8.1288.2.22 is_TCPDstPort_Available . . . 1268
8.1288.2.23 is_TCPSrcPort_Available . . . 1268
8.1288.2.24 is_TranDstPort_Available . . . 1268
8.1288.2.25 is_TranSrcPort_Available . . . 1268
8.1288.2.26 is_UDPDstPort_Available . . . 1268
8.1288.2.27 is_UDPSrcPort_Available . . . 1268
8.1288.2.28 NxtHdrProto . . . 1268
8.1288.2.29 Precedence . . . 1268
8.1288.2.30 TCPDstPort . . . 1268
8.1288.2.31 TCPSrcPort . . . 1268
8.1288.2.32 TranDstPort . . . 1268
8.1288.2.33 TranSrcPort . . . 1268
8.1288.2.34 UDPDstPort . . . 1268
8.1288.2.35 UDPSrcPort . . . 1269
8.1288.2.36 Version . . . 1269
8.1289 unpack_qos_swiQosFlow_t Struct Reference . . . 1269
8.1289.1 Detailed Description . . . 1269
8.1289.2 Field Documentation . . . 1271
8.1289.2.1 DataRate . . . 1271
8.1289.2.2 Index . . . 1271
8.1289.2.3 is_DataRate_Available . . . 1272
8.1289.2.4 is_Jitter_Available . . . 1272
8.1289.2.5 is_Latency_Available . . . 1272
8.1289.2.6 is_LteQci_Available . . . 1272
8.1289.2.7 is_MaxAllowedPktSz_Available . . . 1272
8.1289.2.8 is_MinPolicedPktSz_Available . . . 1272

8.1289.2.9 is_PktErrRate_Available . . . 1272
8.1289.2.10 is_ProfileId3GPP2_Available . . . 1272
8.1289.2.11 is_TokenBucket_Available . . . 1272
8.1289.2.12 is_TrafficClass_Available . . . 1272
8.1289.2.13 is_val_3GPP2Pri_Available . . . 1272
8.1289.2.14 is_val_3GPPImCn_Available . . . 1272
8.1289.2.15 is_val_3GPPResResidualBER_Available . . . 1272
8.1289.2.16 is_val_3GPPSigInd_Available . . . 1272
8.1289.2.17 is_val_3GPPTraHdlPri_Available . . . 1272
8.1289.2.18 Jitter . . . 1272
8.1289.2.19 Latency . . . 1272
8.1289.2.20 LteQci . . . 1272
8.1289.2.21 MaxAllowedPktSz . . . 1272
8.1289.2.22 MinPolicedPktSz . . . 1272
8.1289.2.23 PktErrRate . . . 1272
8.1289.2.24 ProfileId3GPP2 . . . 1272
8.1289.2.25 TokenBucket . . . 1272
8.1289.2.26 TrafficClass . . . 1272
8.1289.2.27 Val_3GPP2Pri . . . 1272
8.1289.2.28 Val_3GPPImCn . . . 1272
8.1289.2.29 Val_3GPPResResidualBER . . . 1272
8.1289.2.30 Val_3GPPSigInd . . . 1272
8.1289.2.31 Val_3GPPTraHdlPri . . . 1273
8.1290 unpack_qos_tokenBucket_t Struct Reference . . . 1273
8.1290.1 Detailed Description . . . 1273
8.1290.2 Field Documentation . . . 1273
8.1290.2.1 bucketSz . . . 1273
8.1290.2.2 peakRate . . . 1273
8.1290.2.3 tokenRate . . . 1273
8.1291 unpack_qos_Tos_t Struct Reference . . . 1273
8.1291.1 Detailed Description . . . 1273
8.1291.2 Field Documentation . . . 1274
8.1291.2.1 mask . . . 1274
8.1291.2.2 val . . . 1274
8.1292 unpack_QosFlowStat_t Struct Reference . . . 1274
8.1292.1 Detailed Description . . . 1274
8.1292.2 Field Documentation . . . 1274
8.1292.2.1 bearerId . . . 1274
8.1292.2.2 tx_bytes . . . 1274
8.1292.2.3 tx_bytes_drp . . . 1274

8.1292.2.4 tx_pkt . . . 1274
8.1292.2.5 tx_pkt_drp . . . 1274
8.1293 unpack_rms_GetSMSWake_t Struct Reference . . . 1274
8.1293.1 Detailed Description . . . 1275
8.1293.2 Field Documentation . . . 1275
8.1293.2.1 enabled . . . 1275
8.1293.2.2 wake_mask . . . 1275
8.1294 unpack_RMTransferStatistics_ind_t Struct Reference . . . 1275
8.1294.1 Detailed Description . . . 1275
8.1294.2 Field Documentation . . . 1276
8.1294.2.1 RxDropConutTlv . . . 1276
8.1294.2.2 RxOkByteCountTlv . . . 1276
8.1294.2.3 RxOkConutTlv . . . 1276
8.1294.2.4 TxDropConutTlv . . . 1276
8.1294.2.5 TxOkByteCountTlv . . . 1276
8.1294.2.6 TxOkConutTlv . . . 1276
8.1295 unpack_sar_SLQSGetRfSarState_t Struct Reference . . . 1276
8.1295.1 Detailed Description . . . 1276
8.1295.2 Field Documentation . . . 1277
8.1295.2.1 pSarRFState . . . 1277
8.1296 unpack_sms_GetSMSCAddress_t Struct Reference . . . 1277
8.1296.1 Detailed Description . . . 1277
8.1296.2 Field Documentation . . . 1277
8.1296.2.1 addressSize . . . 1277
8.1296.2.2 pSMSCAddress . . . 1277
8.1296.2.3 pSMSCType . . . 1277
8.1296.2.4 typeSize . . . 1277
8.1297 unpack_sms_SaveSMS_t Struct Reference . . . 1277
8.1297.1 Detailed Description . . . 1277
8.1297.2 Field Documentation . . . 1278
8.1297.2.1 pMessageIndex . . . 1278
8.1298 unpack_sms_SendSMS_t Struct Reference . . . 1278
8.1298.1 Detailed Description . . . 1278
8.1298.2 Field Documentation . . . 1278
8.1298.2.1 messageFailureCode . . . 1278
8.1298.2.2 messageID . . . 1278
8.1299 unpack_sms_SetNewSMSCallback_ind_t Struct Reference . . . 1278
8.1299.1 Detailed Description . . . 1278
8.1299.2 Field Documentation . . . 1279
8.1299.2.1 ETWSPLMNTlv . . . 1279

8.1299.2.2 ETWSTlv . . . 1279
8.1299.2.3 IMSTlv . . . 1279
8.1299.2.4 MMTlv . . . 1279
8.1299.2.5 NewMMTlv . . . 1279
8.1299.2.6 SMSCTlv . . . 1279
8.1299.2.7 TRMessageTlv . . . 1279
8.1300 unpack_sms_SetNewSMSCallback_t Struct Reference . . . 1279
8.1300.1 Detailed Description . . . 1279
8.1301 unpack_sms_SLQSDeleteSMS_t Struct Reference . . . 1280
8.1301.1 Detailed Description . . . 1280
8.1302 unpack_sms_SLQSGetIndicationRegister_t Struct Reference . . . 1280
8.1302.1 Detailed Description . . . 1280
8.1302.2 Field Documentation . . . 1280
8.1302.2.1 pGetIndicationRegInfo . . . 1280
8.1303 unpack_sms_SLQSGetMessageWaiting_t Struct Reference . . . 1280
8.1303.1 Detailed Description . . . 1280
8.1303.2 Field Documentation . . . 1281
8.1303.2.1 pGetMsgWaitingInfoResp . . . 1281
8.1304 unpack_sms_SLQSGetSMS_t Struct Reference . . . 1281
8.1304.1 Detailed Description . . . 1281
8.1304.2 Field Documentation . . . 1281
8.1304.2.1 message . . . 1281
8.1304.2.2 messageFormat . . . 1281
8.1304.2.3 messageSize . . . 1281
8.1304.2.4 messageTag . . . 1282
8.1305 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 1282
8.1305.1 Detailed Description . . . 1282
8.1305.2 Field Documentation . . . 1282
8.1305.2.1 pBroadcastConfig . . . 1282
8.1305.2.2 pCDMABroadcastConfig . . . 1282
8.1306 unpack_sms_SLQSGetSMSList_t Struct Reference . . . 1282
8.1306.1 Detailed Description . . . 1282
8.1306.2 Field Documentation . . . 1283
8.1306.2.1 messageList . . . 1283
8.1306.2.2 messageListSize . . . 1283
8.1307 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference . . . 1283
8.1307.1 Detailed Description . . . 1283
8.1307.2 Field Documentation . . . 1283
8.1307.2.1 pGetTransLayerInfo . . . 1283
8.1308 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference . . . 1283

8.1308.1 Detailed Description . . . 1283
8.1308.2 Field Documentation . . . 1283
8.1308.2.1 pGetTransNWRegInfo . . . 1283
8.1309 unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . 1283
8.1309.1 Detailed Description . . . 1284
8.1310 unpack_sms_SLQSNWRegInfoCallback_ind_t Struct Reference . . . 1284
8.1310.1 Detailed Description . . . 1284
8.1310.2 Field Documentation . . . 1284
8.1310.2.1 NWRegStat . . . 1284
8.1311 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 1284
8.1311.1 Detailed Description . . . 1284
8.1311.2 Field Documentation . . . 1285
8.1311.2.1 pMaxStorageSizeResp . . . 1285
8.1312 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference . . . 1285
8.1312.1 Detailed Description . . . 1285
8.1312.2 Field Documentation . . . 1285
8.1312.2.1 pMessageProtocol . . . 1285
8.1313 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference . . . 1285
8.1313.1 Detailed Description . . . 1285
8.1313.2 Field Documentation . . . 1286
8.1313.2.1 pSmsStorage . . . 1286
8.1314 unpack_sms_SLQSTransLayerInfoCallback_ind_t Struct Reference . . . 1286
8.1314.1 Detailed Description . . . 1286
8.1314.2 Field Documentation . . . 1286
8.1314.2.1 pTransLayerInfo . . . 1286
8.1314.2.2 regInd . . . 1286
8.1315 unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t Struct Reference . . . 1286
8.1315.1 Detailed Description . . . 1287
8.1315.2 Field Documentation . . . 1288
8.1315.2.1 alphaIDLen . . . 1288
8.1315.2.2 causeCode . . . 1288
8.1315.2.3 errorClass . . . 1288
8.1315.2.4 messageID . . . 1288
8.1315.2.5 msgDelFailureCause . . . 1288
8.1315.2.6 msgDelFailureType . . . 1288
8.1315.2.7 pAlphaID . . . 1288
8.1315.2.8 RPCause . . . 1288
8.1315.2.9 sendStatus . . . 1288
8.1315.2.10 TPCause . . . 1288
8.1315.2.11 userData . . . 1288

8.1316 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . 1288
8.1316.1 Detailed Description . . . 1288
8.1316.2 Field Documentation . . . 1288
8.1316.2.1 messageMode . . . 1288
8.1316.2.2 storageType . . . 1288
8.1317 unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t Struct Reference . . . 1289
8.1317.1 Detailed Description . . . 1289
8.1317.2 Field Documentation . . . 1289
8.1317.2.1 msgWaitInfo . . . 1289
8.1317.2.2 numInstances . . . 1289
8.1318 unpack_swidms_SLQSSwiDmsGetMTU_t Struct Reference . . . 1289
8.1318.1 Detailed Description . . . 1289
8.1318.2 Field Documentation . . . 1290
8.1318.2.1 pEhrpdMTUSize . . . 1290
8.1318.2.2 pHrpdMTUSize . . . 1290
8.1318.2.3 pMTUSize3gpp . . . 1290
8.1318.2.4 pUsbMTUSize . . . 1290
8.1319 unpack_swidms_SLQSSwiDmsGetUsbNetNum_t Struct Reference . . . 1290
8.1319.1 Detailed Description . . . 1290
8.1319.2 Field Documentation . . . 1290
8.1319.2.1 usbNetNum . . . 1290
8.1320 unpack_swidms_SLQSSwiDmsSetMTU_t Struct Reference . . . 1290
8.1320.1 Detailed Description . . . 1290
8.1320.2 Field Documentation . . . 1291
8.1320.2.1 Tlvresult . . . 1291
8.1321 unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . 1291
8.1321.1 Detailed Description . . . 1291
8.1321.2 Field Documentation . . . 1292
8.1321.2.1 fix_rate . . . 1292
8.1321.2.2 fix_rate_reported . . . 1292
8.1321.2.3 fix_type . . . 1292
8.1321.2.4 fix_type_reported . . . 1292
8.1321.2.5 function . . . 1292
8.1321.2.6 function_reported . . . 1292
8.1321.2.7 max_dist . . . 1292
8.1321.2.8 max_dist_reported . . . 1292
8.1321.2.9 max_time . . . 1292
8.1321.2.10 max_time_reported . . . 1292
8.1322 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference . . . 1292
8.1322.1 Detailed Description . . . 1293

8.1322.2 Field Documentation . . . 1293
8.1322.2.1 eventType . . . 1293
8.1322.2.2 SessionInfoConfig . . . 1293
8.1322.2.3 SessionInfoFota . . . 1293
8.1322.2.4 SessionInfoNotification . . . 1293
8.1323 npack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 1293
8.1323.1 Detailed Description . . . 1294
8.1323.2 Field Documentation . . . 1295
8.1323.2.1 Date . . . 1295
8.1323.2.2 DateLength . . . 1295
8.1323.2.3 PkgDescLength . . . 1295
8.1323.2.4 PkgDescription . . . 1295
8.1323.2.5 PkgName . . . 1295
8.1323.2.6 PkgNameLength . . . 1295
8.1323.2.7 RetryCount . . . 1295
8.1323.2.8 SessionState . . . 1295
8.1323.2.9 SessionType . . . 1295
8.1323.2.10 Severity . . . 1295
8.1323.2.11 Source . . . 1295
8.1323.2.12 SourceLength . . . 1296
8.1323.2.13 Status . . . 1296
8.1323.2.14 Time . . . 1296
8.1323.2.15 TimeLength . . . 1296
8.1323.2.16 UpdateCompleteStatus . . . 1296
8.1324 npack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . 1296
8.1324.1 Detailed Description . . . 1296
8.1324.2 Field Documentation . . . 1297
8.1324.2.1 Autosdm . . . 1297
8.1324.2.2 FOTAdownload . . . 1297
8.1324.2.3 FOTAUpdate . . . 1297
8.1324.2.4 FwAutoCheck . . . 1297
8.1324.2.5 OMADMEnabled . . . 1297
8.1325 npack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 1297
8.1325.1 Detailed Description . . . 1297
8.1325.2 Field Documentation . . . 1298
8.1325.2.1 FwAvailability . . . 1298
8.1326 npack_uim_ChangePin_t Struct Reference . . . 1298
8.1326.1 Detailed Description . . . 1298
8.1326.2 Field Documentation . . . 1298
8.1326.2.1 pEncryptedPIN1 . . . 1298

8.1326.2.2 pIndicationToken . . . 1299
8.1326.2.3 pRemainingRetries . . . 1299
8.1326.2.4 Tlvresult . . . 1299
8.1327 unpack_uim_GetCardStatus_t Struct Reference . . . 1299
8.1327.1 Detailed Description . . . 1299
8.1327.2 Field Documentation . . . 1299
8.1327.2.1 pCardStatus . . . 1299
8.1327.2.2 pHotSwapStatus . . . 1299
8.1327.2.3 Tlvresult . . . 1299
8.1328 unpack_uim_ReadTransparent_t Struct Reference . . . 1299
8.1328.1 Detailed Description . . . 1300
8.1328.2 Field Documentation . . . 1300
8.1328.2.1 pCardResult . . . 1300
8.1328.2.2 pEncryptedData . . . 1300
8.1328.2.3 pIndicationToken . . . 1300
8.1328.2.4 pReadResult . . . 1300
8.1328.2.5 Tlvresult . . . 1300
8.1329 unpack_uim_SetPinProtection_t Struct Reference . . . 1300
8.1329.1 Detailed Description . . . 1300
8.1329.2 Field Documentation . . . 1301
8.1329.2.1 pEncryptedPIN1 . . . 1301
8.1329.2.2 pIndicationToken . . . 1301
8.1329.2.3 pRemainingRetries . . . 1301
8.1329.2.4 Tlvresult . . . 1301
8.1330 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference . . . 1301
8.1330.1 Detailed Description . . . 1301
8.1330.2 Field Documentation . . . 1301
8.1330.2.1 bNumberOfPhySlots . . . 1302
8.1330.2.2 slotsstatusChange . . . 1302
8.1331 unpack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 1302
8.1331.1 Detailed Description . . . 1302
8.1331.2 Field Documentation . . . 1302
8.1331.2.1 pAuthenticateResult . . . 1302
8.1331.2.2 pCardResult . . . 1302
8.1331.2.3 pIndicationToken . . . 1302
8.1332 unpack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 1302
8.1332.1 Detailed Description . . . 1302
8.1332.2 Field Documentation . . . 1303
8.1332.2.1 pRemainingRetries . . . 1303
8.1333 unpack_uim_SLQSUIMEventRegister_t Struct Reference . . . 1303

8.1333.1 Detailed Description . . . 1303
8.1333.2 Field Documentation . . . 1303
8.1333.2.1 eventMask . . . 1303
8.1334 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 1303
8.1334.1 Detailed Description . . . 1303
8.1334.2 Field Documentation . . . 1304
8.1334.2.1 pAutoSelection . . . 1304
8.1334.2.2 pHaltSubscription . . . 1304
8.1334.2.3 pPersonalizationStatus . . . 1304
8.1335 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 1304
8.1335.1 Detailed Description . . . 1304
8.1335.2 Field Documentation . . . 1305
8.1335.2.1 pCardResult . . . 1305
8.1335.2.2 pFileAttributes . . . 1305
8.1335.2.3 pIndicationToken . . . 1305
8.1336 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference . . . 1305
8.1336.1 Detailed Description . . . 1305
8.1336.2 Field Documentation . . . 1305
8.1336.2.1 pNumberOfPhySlot . . . 1305
8.1336.2.2 pUimSlotsStatus . . . 1305
8.1337 unpack_uim_SLQSUIMRefreshCallback_Ind_t Struct Reference . . . 1305
8.1337.1 Detailed Description . . . 1306
8.1337.2 Field Documentation . . . 1306
8.1337.2.1 refreshEvent . . . 1306
8.1337.2.2 TlvPresent . . . 1306
8.1338 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 1306
8.1338.1 Detailed Description . . . 1306
8.1338.2 Field Documentation . . . 1306
8.1338.2.1 pRefreshEvent . . . 1306
8.1339 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference . . . 1306
8.1339.1 Detailed Description . . . 1306
8.1339.2 Field Documentation . . . 1307
8.1339.2.1 pCardStatus . . . 1307
8.1340 unpack_uim_UnblockPin_t Struct Reference . . . 1307
8.1340.1 Detailed Description . . . 1307
8.1340.2 Field Documentation . . . 1307
8.1340.2.1 pEncryptedPIN1 . . . 1307
8.1340.2.2 pIndicationToken . . . 1307
8.1340.2.3 pRemainingRetries . . . 1307
8.1340.2.4 Tlvresult . . . 1307

8.1341 unpack_uim_VerifyPin_t Struct Reference . . . 1307
8.1341.1 Detailed Description . . . 1308
8.1341.2 Field Documentation . . . 1308
8.1341.2.1 pEncryptedPIN1 . . . 1308
8.1341.2.2 pIndicationToken . . . 1308
8.1341.2.3 pRemainingRetries . . . 1308
8.1341.2.4 Tlvresult . . . 1308
8.1342 unpack_voice_allCallStatusCallback_ind_t Struct Reference . . . 1308
8.1342.1 Detailed Description . . . 1309
8.1342.2 Field Documentation . . . 1311
8.1342.2.1 arrCallInfomation . . . 1311
8.1342.2.2 pArrAlertingPattern . . . 1311
8.1342.2.3 pArrAlertingType . . . 1311
8.1342.2.4 pArrAlphaID . . . 1311
8.1342.2.5 pArrCalledPartyNum . . . 1311
8.1342.2.6 pArrCallEndReason . . . 1311
8.1342.2.7 pArrConnectPartyNum . . . 1311
8.1342.2.8 pArrDiagInfo . . . 1311
8.1342.2.9 pArrRedirPartyNum . . . 1311
8.1342.2.10 pArrRemotePartyName . . . 1311
8.1342.2.11 pArrRemotePartyNum . . . 1311
8.1342.2.12 pArrSvcOption . . . 1311
8.1343 unpack_voice_DTMFEventCallback_ind_t Struct Reference . . . 1311
8.1343.1 Detailed Description . . . 1312
8.1343.2 Field Documentation . . . 1312
8.1343.2.1 DTMFInformation . . . 1312
8.1343.2.2 pOffLength . . . 1312
8.1343.2.3 pOnLength . . . 1312
8.1344 unpack_voice_OTASPStatusCallback_ind_t Struct Reference . . . 1312
8.1344.1 Detailed Description . . . 1312
8.1344.2 Field Documentation . . . 1313
8.1344.2.1 callID . . . 1313
8.1344.2.2 OTASPStatus . . . 1313
8.1345 unpack_voice_SLQSOriginateUSSD_t Struct Reference . . . 1313
8.1345.1 Detailed Description . . . 1313
8.1345.2 Field Documentation . . . 1316
8.1345.2.1 pAlphaIDInfo . . . 1316
8.1345.2.2 pCallId . . . 1316
8.1345.2.3 pCcResultType . . . 1316
8.1345.2.4 pCCSuppsType . . . 1316

8.1345.2.5 pfailureCause . . . 1316
8.1345.2.6 pUSSDInfo . . . 1316
8.1346 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 1316
8.1346.1 Detailed Description . . . 1316
8.1346.2 Field Documentation . . . 1317
8.1346.2.1 pCallId . . . 1317
8.1347 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 1317
8.1347.1 Detailed Description . . . 1317
8.1347.2 Field Documentation . . . 1317
8.1347.2.1 pCallID . . . 1317
8.1348 unpack_voice_SLQSVoiceDialCall_t Struct Reference . . . 1317
8.1348.1 Detailed Description . . . 1317
8.1348.2 Field Documentation . . . 1318
8.1348.2.1 pAlphaIDInfo . . . 1318
8.1348.2.2 pCallID . . . 1318
8.1348.2.3 pCCResultType . . . 1318
8.1348.2.4 pCCSUPSType . . . 1318
8.1349 unpack_voice_SLQSVoiceEndCall_t Struct Reference . . . 1318
8.1349.1 Detailed Description . . . 1318
8.1349.2 Field Documentation . . . 1319
8.1349.2.1 pCallId . . . 1319
8.1350 unpack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference . . . 1319
8.1350.1 Detailed Description . . . 1319
8.1350.2 Field Documentation . . . 1321
8.1350.2.1 pArrAlertingPattern . . . 1321
8.1350.2.2 pArrAlertingType . . . 1321
8.1350.2.3 pArrAlphaID . . . 1321
8.1350.2.4 pArrCalledPartyNum . . . 1321
8.1350.2.5 pArrCallEndReason . . . 1321
8.1350.2.6 pArrCallInfo . . . 1321
8.1350.2.7 pArrConnectPartyNum . . . 1321
8.1350.2.8 pArrDiagInfo . . . 1321
8.1350.2.9 pArrRedirPartyNum . . . 1321
8.1350.2.10 pArrRemotePartyName . . . 1321
8.1350.2.11 pArrRemotePartyNum . . . 1321
8.1350.2.12 pArrSvcOption . . . 1321
8.1350.2.13 pArrUUSInfo . . . 1321
8.1350.2.14 pOTASPStatus . . . 1321
8.1350.2.15 pVoicePrivacy . . . 1321
8.1351 unpack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 1321

8.1351.1 Detailed Description . . . 1322
8.1351.2 Field Documentation . . . 1323
8.1351.2.1 pAlphaIDInfo . . . 1323
8.1351.2.2 pCallID . . . 1323
8.1351.2.3 pCCResType . . . 1323
8.1351.2.4 pCCSUPSType . . . 1323
8.1351.2.5 pFailCause . . . 1323
8.1351.2.6 pSvcClass . . . 1323
8.1352 npack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 1323
8.1352.1 Detailed Description . . . 1323
8.1352.2 Field Documentation . . . 1324
8.1352.2.1 pAlphaIDInfo . . . 1324
8.1352.2.2 pCallID . . . 1324
8.1352.2.3 pCCResType . . . 1324
8.1352.2.4 pCCSUPSType . . . 1324
8.1352.2.5 pFailCause . . . 1324
8.1352.2.6 pGetCallFWExtInfo . . . 1324
8.1352.2.7 pGetCallFWInfo . . . 1324
8.1353 npack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 1324
8.1353.1 Detailed Description . . . 1325
8.1353.2 Field Documentation . . . 1327
8.1353.2.1 pAlertingPattern . . . 1327
8.1353.2.2 pAlertType . . . 1327
8.1353.2.3 pAlphaIDInfo . . . 1327
8.1353.2.4 pCallInfo . . . 1327
8.1353.2.5 pConnectNumInfo . . . 1327
8.1353.2.6 pDiagInfo . . . 1327
8.1353.2.7 pOTASPStatus . . . 1327
8.1353.2.8 pRemotePartyName . . . 1327
8.1353.2.9 pRemotePartyNum . . . 1327
8.1353.2.10 pSrvOpt . . . 1327
8.1353.2.11 pUUSInfo . . . 1327
8.1353.2.12 pVoicePrivacy . . . 1327
8.1354 npack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 1328
8.1354.1 Detailed Description . . . 1328
8.1354.2 Field Documentation . . . 1329
8.1354.2.1 pAlphaIDInfo . . . 1329
8.1354.2.2 pCallID . . . 1329
8.1354.2.3 pCCResType . . . 1329
8.1354.2.4 pCCSUPSType . . . 1329

8.1354.2.5 pFailCause . . . 1329
8.1354.2.6 pSvcClass . . . 1329
8.1355 unpack_voice_SLQSVoiceGetCLIP_t Struct Reference . . . 1329
8.1355.1 Detailed Description . . . 1329
8.1355.2 Field Documentation . . . 1330
8.1355.2.1 pAlphaIDInfo . . . 1330
8.1355.2.2 pCallID . . . 1330
8.1355.2.3 pCCResType . . . 1330
8.1355.2.4 pCCSUPSType . . . 1330
8.1355.2.5 pCLIPResp . . . 1330
8.1355.2.6 pFailCause . . . 1330
8.1356 unpack_voice_SLQSVoiceGetCLIR_t Struct Reference . . . 1330
8.1356.1 Detailed Description . . . 1331
8.1356.2 Field Documentation . . . 1331
8.1356.2.1 pAlphaIDInfo . . . 1331
8.1356.2.2 pCallID . . . 1331
8.1356.2.3 pCCResType . . . 1332
8.1356.2.4 pCCSUPSType . . . 1332
8.1356.2.5 pCLIRResp . . . 1332
8.1356.2.6 pFailCause . . . 1332
8.1357 unpack_voice_SLQSVoiceGetCNAP_t Struct Reference . . . 1332
8.1357.1 Detailed Description . . . 1332
8.1357.2 Field Documentation . . . 1333
8.1357.2.1 pAlphaIDInfo . . . 1333
8.1357.2.2 pCallID . . . 1333
8.1357.2.3 pCCResType . . . 1333
8.1357.2.4 pCCSUPSType . . . 1333
8.1357.2.5 pCNAPResp . . . 1333
8.1357.2.6 pFailCause . . . 1333
8.1358 unpack_voice_SLQSVoiceGetCOLP_t Struct Reference . . . 1333
8.1358.1 Detailed Description . . . 1333
8.1358.2 Field Documentation . . . 1334
8.1358.2.1 pAlphaIDInfo . . . 1334
8.1358.2.2 pCallID . . . 1334
8.1358.2.3 pCCResType . . . 1334
8.1358.2.4 pCCSUPSType . . . 1334
8.1358.2.5 pCOLPResp . . . 1334
8.1358.2.6 pFailCause . . . 1334
8.1359 unpack_voice_SLQSVoiceGetCOLR_t Struct Reference . . . 1334
8.1359.1 Detailed Description . . . 1335

8.1359.2 Field Documentation . . . 1336
8.1359.2.1 pAlphaIDInfo . . . 1336
8.1359.2.2 pCallID . . . 1336
8.1359.2.3 pCCResType . . . 1336
8.1359.2.4 pCCSUPSType . . . 1336
8.1359.2.5 pCOLRResp . . . 1336
8.1359.2.6 pFailCause . . . 1336
8.1360 unpack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 1336
8.1360.1 Detailed Description . . . 1336
8.1360.2 Field Documentation . . . 1338
8.1360.2.1 pAirTimerCnt . . . 1338
8.1360.2.2 pAutoAnswerStat . . . 1338
8.1360.2.3 pCurAMRConfig . . . 1338
8.1360.2.4 pCurPrefVoiceSO . . . 1338
8.1360.2.5 pCurrTTYMode . . . 1338
8.1360.2.6 pCurVoiceDomainPref . . . 1338
8.1360.2.7 pCurVoicePrivacyPref . . . 1338
8.1360.2.8 pRoamTimerCnt . . . 1338
8.1361 unpack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 1338
8.1361.1 Detailed Description . . . 1338
8.1361.2 Field Documentation . . . 1338
8.1361.2.1 pFailCause . . . 1338
8.1362 unpack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 1338
8.1362.1 Detailed Description . . . 1338
8.1362.2 Field Documentation . . . 1339
8.1362.2.1 pCallID . . . 1339
8.1363 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 1339
8.1363.1 Detailed Description . . . 1339
8.1363.2 Field Documentation . . . 1340
8.1363.2.1 pAlphaIDInfo . . . 1340
8.1363.2.2 pCallID . . . 1340
8.1363.2.3 pCCResType . . . 1340
8.1363.2.4 pCCSUPSType . . . 1340
8.1363.2.5 pFailCause . . . 1340
8.1364 unpack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 1340
8.1364.1 Detailed Description . . . 1340
8.1364.2 Field Documentation . . . 1341
8.1364.2.1 pAirTimerStatus . . . 1341
8.1364.2.2 pAutoAnsStatus . . . 1341
8.1364.2.3 pPrefVoiceSOStatus . . . 1342

8.1364.2.4 pRoamTimerStatus . . . 1342
8.1364.2.5 pTTYConfigStatus . . . 1342
8.1364.2.6 pVoiceDomainPrefStatus . . . 1342
8.1365 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 1342
8.1365.1 Detailed Description . . . 1342
8.1365.2 Field Documentation . . . 1343
8.1365.2.1 pAlphaIDInfo . . . 1343
8.1365.2.2 pCallID . . . 1343
8.1365.2.3 pCCResultType . . . 1343
8.1365.2.4 pCCSUPSType . . . 1343
8.1365.2.5 pFailCause . . . 1343
8.1366 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 1343
8.1366.1 Detailed Description . . . 1343
8.1366.2 Field Documentation . . . 1343
8.1366.2.1 pCallID . . . 1343
8.1367 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 1343
8.1367.1 Detailed Description . . . 1344
8.1367.2 Field Documentation . . . 1344
8.1367.2.1 callID . . . 1344
8.1368 unpack_voice_SLQSVoiceSUPSCallback_ind_t Struct Reference . . . 1344
8.1368.1 Detailed Description . . . 1344
8.1368.2 Field Documentation . . . 1346
8.1368.2.1 pAlphaIDInfo . . . 1346
8.1368.2.2 pCallBarPasswd . . . 1346
8.1368.2.3 pCallFwdInfo . . . 1346
8.1368.2.4 pCallFWNum . . . 1346
8.1368.2.5 pCallFWTimerVal . . . 1346
8.1368.2.6 pCallID . . . 1346
8.1368.2.7 pCLIPstatus . . . 1346
8.1368.2.8 pCLIRstatus . . . 1346
8.1368.2.9 pCNAPstatus . . . 1346
8.1368.2.10 pCOLPstatus . . . 1346
8.1368.2.11 pCOLRstatus . . . 1346
8.1368.2.12 pDataSrc . . . 1346
8.1368.2.13 pFailCause . . . 1346
8.1368.2.14 pNewPwdData . . . 1346
8.1368.2.15 pReason . . . 1346
8.1368.2.16 pSvcClass . . . 1346
8.1368.2.17 pUSSInfo . . . 1346
8.1368.2.18 pUPSInformation . . . 1346

8.1369 unpack_voice_SUPSNotificationCallback_ind_t Struct Reference . . . 1346
8.1369.1 Detailed Description . . . 1347
8.1369.2 Field Documentation . . . 1348
8.1369.2.1 callID . . . 1348
8.1369.2.2 notifType . . . 1348
8.1369.2.3 pCUGIndex . . . 1348
8.1369.2.4 pECTNum . . . 1348
8.1370 unpack_voice_USSDNotificationCallback_ind_t Struct Reference . . . 1348
8.1370.1 Detailed Description . . . 1348
8.1370.2 Field Documentation . . . 1348
8.1370.2.1 notification_Type . . . 1348
8.1370.2.2 USSDNotificationNetworkInfo . . . 1348
8.1371 unpack_voice_VoiceInfoRecCallback_ind_t Struct Reference . . . 1348
8.1371.1 Detailed Description . . . 1349
8.1371.2 Field Documentation . . . 1350
8.1371.2.1 callID . . . 1350
8.1371.2.2 pCalledPartyInfo . . . 1350
8.1371.2.3 pCallerIDInfo . . . 1350
8.1371.2.4 pCallerNameInfo . . . 1350
8.1371.2.5 pCallingPartyInfo . . . 1350
8.1371.2.6 pCallWaitInd . . . 1350
8.1371.2.7 pCLIRCause . . . 1350
8.1371.2.8 pConnectNumInfo . . . 1350
8.1371.2.9 pDispInfo . . . 1350
8.1371.2.10 pExtDispInfo . . . 1350
8.1371.2.11 pExtDispRecInfo . . . 1350
8.1371.2.12 pLineCtrlInfo . . . 1350
8.1371.2.13 pNSSAudioCtrl . . . 1351
8.1371.2.14 pNSSRelease . . . 1351
8.1371.2.15 pRedirNumInfo . . . 1351
8.1371.2.16 pSignalInfo . . . 1351
8.1372 unpack_voice_voicePrivacyChangeCallback_ind_t Struct Reference . . . 1351
8.1372.1 Detailed Description . . . 1351
8.1372.2 Field Documentation . . . 1351
8.1372.2.1 callID . . . 1351
8.1372.2.2 voicePrivacy . . . 1351
8.1373 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference . . . 1351
8.1373.1 Detailed Description . . . 1351
8.1373.2 Field Documentation . . . 1352
8.1373.2.1 DHCPv4LeaseOptTlv . . . 1352

8.1373.2.2 DHCPv4LeaseStateTlv . . . 1352
8.1373.2.3 IPv4AddrTlv . . . 1352
8.1373.2.4 ProfileIdTlv . . . 1352
8.1374 unpack_wds_GetAutoconnect_t Struct Reference . . . 1352
8.1374.1 Detailed Description . . . 1352
8.1374.2 Field Documentation . . . 1352
8.1374.2.1 psetting . . . 1352
8.1375 unpack_wds_GetByteTotals_t Struct Reference . . . 1352
8.1375.1 Detailed Description . . . 1352
8.1375.2 Field Documentation . . . 1353
8.1375.2.1 pRXTotalBytes . . . 1353
8.1375.2.2 pTXTotalBytes . . . 1353
8.1376 unpack_wds_GetConnectionRate_t Struct Reference . . . 1353
8.1376.1 Detailed Description . . . 1353
8.1376.2 Field Documentation . . . 1353
8.1376.2.1 currentChannelRXRate . . . 1353
8.1376.2.2 currentChannelTXRate . . . 1353
8.1376.2.3 maxChannelRXRate . . . 1353
8.1376.2.4 maxChannelTXRate . . . 1354
8.1377 unpack_wds_GetDataBearerTechnology_t Struct Reference . . . 1354
8.1377.1 Detailed Description . . . 1354
8.1377.2 Field Documentation . . . 1354
8.1377.2.1 pDataBearer . . . 1354
8.1378 unpack_wds_GetDefaultProfile_t Struct Reference . . . 1354
8.1378.1 Detailed Description . . . 1355
8.1378.2 Field Documentation . . . 1356
8.1378.2.1 apnname . . . 1356
8.1378.2.2 apnsize . . . 1356
8.1378.2.3 auth . . . 1356
8.1378.2.4 paddr . . . 1356
8.1378.2.5 paddrv6 . . . 1356
8.1378.2.6 name . . . 1356
8.1378.2.7 namesize . . . 1357
8.1378.2.8 pdptype . . . 1357
8.1378.2.9 pridns . . . 1357
8.1378.2.10 pridnsv6 . . . 1357
8.1378.2.11 secdns . . . 1357
8.1378.2.12 secdnsv6 . . . 1357
8.1378.2.13 username . . . 1357
8.1378.2.14 usersize . . . 1357

8.1379 unpack_wds_GetDefaultProfileNum_t Struct Reference . . . 1357
8.1379.1 Detailed Description . . . 1357
8.1379.2 Field Documentation . . . 1357
8.1379.2.1 index . . . 1357
8.1380 unpack_wds_GetDormancyState_t Struct Reference . . . 1357
8.1380.1 Detailed Description . . . 1357
8.1380.2 Field Documentation . . . 1358
8.1380.2.1 dormancyState . . . 1358
8.1381 unpack_wds_GetLastMobileIPError_t Struct Reference . . . 1358
8.1381.1 Detailed Description . . . 1358
8.1381.2 Field Documentation . . . 1358
8.1381.2.1 error . . . 1358
8.1382 unpack_wds_GetMobileIP_t Struct Reference . . . 1358
8.1382.1 Detailed Description . . . 1358
8.1382.2 Field Documentation . . . 1358
8.1382.2.1 mipMode . . . 1358
8.1383 unpack_wds_GetMobileIPProfile_t Struct Reference . . . 1359
8.1383.1 Detailed Description . . . 1359
8.1383.2 Field Documentation . . . 1360
8.1383.2.1 AAASPI . . . 1360
8.1383.2.2 AAAState . . . 1360
8.1383.2.3 address . . . 1360
8.1383.2.4 enabled . . . 1360
8.1383.2.5 HASPI . . . 1360
8.1383.2.6 HAState . . . 1360
8.1383.2.7 NAI . . . 1360
8.1383.2.8 naiSize . . . 1360
8.1383.2.9 primaryHA . . . 1360
8.1383.2.10 revTunneling . . . 1360
8.1383.2.11 secondaryHA . . . 1360
8.1384 unpack_wds_GetPacketStatistics_t Struct Reference . . . 1360
8.1384.1 Detailed Description . . . 1361
8.1384.2 Field Documentation . . . 1362
8.1384.2.1 pRXDroppedCount . . . 1362
8.1384.2.2 pRXOkBytesCount . . . 1362
8.1384.2.3 pRXOKBytesLastCall . . . 1362
8.1384.2.4 pRXPacketErrors . . . 1362
8.1384.2.5 pRXPacketOverflows . . . 1362
8.1384.2.6 pRXPacketSuccesses . . . 1362
8.1384.2.7 pTXDroppedCount . . . 1362

8.1384.2.8 pTXOkBytesCount . . . 1362
8.1384.2.9 pTXOKBytesLastCall . . . 1362
8.1384.2.10 pTXPacketErrors . . . 1362
8.1384.2.11 pTXPacketOverflows . . . 1362
8.1384.2.12 pTXPacketSuccesses . . . 1362
8.1385 unpack_wds_GetPacketStatus_t Struct Reference . . . 1362
8.1385.1 Detailed Description . . . 1363
8.1385.2 Field Documentation . . . 1364
8.1385.2.1 rXDroppedCount . . . 1364
8.1385.2.2 rXOkBytesCount . . . 1364
8.1385.2.3 rXOKBytesLastCall . . . 1364
8.1385.2.4 rXPacketErrors . . . 1364
8.1385.2.5 rXPacketOverflows . . . 1364
8.1385.2.6 rXPacketSuccesses . . . 1364
8.1385.2.7 tXDroppedCount . . . 1364
8.1385.2.8 tXOkBytesCount . . . 1364
8.1385.2.9 tXOKBytesLastCall . . . 1364
8.1385.2.10 tXPacketErrors . . . 1364
8.1385.2.11 tXPacketOverflows . . . 1364
8.1385.2.12 tXPacketSuccesses . . . 1364
8.1386 unpack_wds_GetSessionDuration_t Struct Reference . . . 1364
8.1386.1 Detailed Description . . . 1364
8.1386.2 Field Documentation . . . 1365
8.1386.2.1 callDuration . . . 1365
8.1387 unpack_wds_GetSessionState_t Struct Reference . . . 1365
8.1387.1 Detailed Description . . . 1365
8.1387.2 Field Documentation . . . 1365
8.1387.2.1 connectionStatus . . . 1365
8.1388 unpack_wds_RMSetTransferStatistics_t Struct Reference . . . 1365
8.1388.1 Detailed Description . . . 1365
8.1389 unpack_wds_SetMobileIPProfile_t Struct Reference . . . 1365
8.1389.1 Detailed Description . . . 1365
8.1390 unpack_wds_SLQSCreateProfile_t Struct Reference . . . 1366
8.1390.1 Detailed Description . . . 1366
8.1390.2 Field Documentation . . . 1366
8.1390.2.1 pCreateProfileOut . . . 1366
8.1390.2.2 pProfileID . . . 1366
8.1390.2.3 Tlvresult . . . 1366
8.1391 unpack_wds_SLQSDeleteProfile_t Struct Reference . . . 1366
8.1391.1 Detailed Description . . . 1366

8.1391.2 Field Documentation . . . 1367
8.1391.2.1 extendedErrorCode . . . 1367
8.1392 npack_wds_SLQSDUNCallInfoCallBack_ind_t Struct Reference . . . 1367
8.1392.1 Detailed Description . . . 1367
8.1392.2 Field Documentation . . . 1368
8.1392.2.1 CRTlv . . . 1368
8.1392.2.2 CSTlv . . . 1368
8.1392.2.3 DBTTlv . . . 1368
8.1392.2.4 DSTlv . . . 1368
8.1392.2.5 LMCERTlv . . . 1368
8.1392.2.6 RXBOTlv . . . 1368
8.1392.2.7 TXBOTlv . . . 1368
8.1393 npack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . 1368
8.1393.1 Detailed Description . . . 1368
8.1393.2 Field Documentation . . . 1369
8.1393.2.1 _3gppRelease . . . 1369
8.1393.2.2 defaultPDNEnabled . . . 1369
8.1393.2.3 LTEAttachProfile . . . 1369
8.1393.2.4 LTEAttachProfileList . . . 1369
8.1393.2.5 LTEAttachProfileListLen . . . 1370
8.1393.2.6 profileList . . . 1370
8.1394 npack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 1370
8.1394.1 Detailed Description . . . 1370
8.1394.2 Field Documentation . . . 1370
8.1394.2.1 currNetworkInfo . . . 1370
8.1394.2.2 networkInfoLen . . . 1370
8.1394.2.3 prefNetwork . . . 1370
8.1395 npack_wds_SLQSGetCurrentChannelRate_t Struct Reference . . . 1370
8.1395.1 Detailed Description . . . 1371
8.1395.2 Field Documentation . . . 1371
8.1395.2.1 current_channel_rx_rate . . . 1371
8.1395.2.2 current_channel_tx_rate . . . 1371
8.1395.2.3 max_channel_rx_rate . . . 1371
8.1395.2.4 max_channel_tx_rate . . . 1371
8.1396 npack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 1371
8.1396.1 Detailed Description . . . 1371
8.1396.2 Field Documentation . . . 1372
8.1396.2.1 curDataBearerTechnology . . . 1372
8.1396.2.2 dataBearerMask . . . 1372
8.1396.2.3 lastCallDataBearerTechnology . . . 1372

8.1397 unpack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 1372
8.1397.1 Detailed Description . . . 1372
8.1397.2 Field Documentation . . . 1374
8.1397.2.1 callEndReason . . . 1374
8.1397.2.2 channelRate . . . 1374
8.1397.2.3 connectionStatus . . . 1374
8.1397.2.4 dataBearerTech . . . 1375
8.1397.2.5 dormancyStatus . . . 1375
8.1397.2.6 lastCallDataBearerTech . . . 1375
8.1397.2.7 lastCallRXOKBytesCnt . . . 1375
8.1397.2.8 lastCallTXOKBytesCnt . . . 1375
8.1397.2.9 mdmCallDurationActive . . . 1375
8.1397.2.10 rxOKBytesCount . . . 1375
8.1397.2.11 txOKBytesCount . . . 1375
8.1398 unpack_wds_SLQSGetProfileSettings_t Struct Reference . . . 1375
8.1398.1 Detailed Description . . . 1375
8.1398.2 Field Documentation . . . 1375
8.1398.2.1 pProfileSettings . . . 1375
8.1398.2.2 ProfileType . . . 1375
8.1398.2.3 Tlvresult . . . 1375
8.1399 unpack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 1375
8.1399.1 Detailed Description . . . 1376
8.1399.2 Field Documentation . . . 1377
8.1399.2.1 APNName . . . 1377
8.1399.2.2 Authentication . . . 1377
8.1399.2.3 DomainList . . . 1377
8.1399.2.4 GPRSGrantedQoS . . . 1378
8.1399.2.5 GWAddressV4 . . . 1378
8.1399.2.6 IMCNflag . . . 1378
8.1399.2.7 IPFamilyPreference . . . 1378
8.1399.2.8 IPv4 . . . 1378
8.1399.2.9 IPV6AddrInfo . . . 1378
8.1399.2.10 IPV6GWAddrInfo . . . 1378
8.1399.2.11 Mtu . . . 1378
8.1399.2.12 PCSCFAddrPCO . . . 1378
8.1399.2.13 PCSCFFQDNAddrList . . . 1378
8.1399.2.14 PDPType . . . 1378
8.1399.2.15 PrimaryDNSV4 . . . 1378
8.1399.2.16 PrimaryDNSV6 . . . 1378
8.1399.2.17 ProfileID . . . 1378

8.1399.2.18 ProfileName . . . 1378
8.1399.2.19 SecondaryDNSV4 . . . 1378
8.1399.2.20 SecondaryDNSV6 . . . 1378
8.1399.2.21 ServerAddrList . . . 1378
8.1399.2.22 SubnetMaskV4 . . . 1378
8.1399.2.23 Technology . . . 1378
8.1399.2.24 UMTSGrantedQoS . . . 1378
8.1399.2.25 Username . . . 1378
8.1400 qmi_wds_SLQSModifyProfile_t Struct Reference . . . 1378
8.1400.1 Detailed Description . . . 1378
8.1400.2 Field Documentation . . . 1379
8.1400.2.1 ExtErrorCode . . . 1379
8.1401 qmi_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 1379
8.1401.1 Detailed Description . . . 1379
8.1401.2 Field Documentation . . . 1379
8.1401.2.1 Tlvresult . . . 1379
8.1402 qmi_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference . . . 1379
8.1402.1 Detailed Description . . . 1379
8.1402.2 Field Documentation . . . 1381
8.1402.2.1 bearerID . . . 1381
8.1402.2.2 conn_status . . . 1381
8.1402.2.3 ipFamily . . . 1381
8.1402.2.4 reconfigReqd . . . 1381
8.1402.2.5 sessionEndReason . . . 1381
8.1402.2.6 techName . . . 1381
8.1402.2.7 verboseSessnEndReason . . . 1381
8.1402.2.8 verboseSessnEndReasonType . . . 1381
8.1403 qmi_wds_SLQSSetWdsEventCallback_ind_t Struct Reference . . . 1381
8.1403.1 Detailed Description . . . 1382
8.1403.2 Field Documentation . . . 1382
8.1403.2.1 currDBTechAvail . . . 1382
8.1403.2.2 currNWInfo . . . 1382
8.1403.2.3 dataSysStatAvail . . . 1382
8.1403.2.4 dBTechAvail . . . 1382
8.1403.2.5 dBtechExtAvail . . . 1382
8.1403.2.6 dBTechExtRatValue . . . 1382
8.1403.2.7 dBTechExtSoMask . . . 1382
8.1403.2.8 dBTechnology . . . 1382
8.1403.2.9 dBtechnologyExt . . . 1382
8.1403.2.10 dormancyStatAvail . . . 1382

8.1403.2.11 dormancyStatus . . . 1382
8.1403.2.12 ipstatAvail . . . 1382
8.1403.2.13 ipStatus . . . 1383
8.1403.2.14 etInfoLen . . . 1383
8.1403.2.15 refNetwork . . . 1383
8.1403.2.16 atMask . . . 1383
8.1403.2.17 x_bytes . . . 1383
8.1403.2.18 x_pkts . . . 1383
8.1403.2.19 pMask . . . 1383
8.1403.2.20 x_bytes . . . 1383
8.1403.2.21 x_pkts . . . 1383
8.1403.2.22 erStatAvail . . . 1383
8.1404 npack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 1383
8.1404.1 Detailed Description . . . 1383
8.1404.2 Field Documentation . . . 1383
8.1404.2.1 pHwConfig . . . 1383
8.1404.2.2 pRequestOptionList . . . 1383
8.1405 npack_wds_SLQSSGetLoopback_t Struct Reference . . . 1383
8.1405.1 Detailed Description . . . 1384
8.1405.2 Field Documentation . . . 1384
8.1405.2.1 ByteLoopbackMode . . . 1384
8.1405.2.2 ByteLoopbackMultiplier . . . 1384
8.1406 npack_wds_SLQSStartDataSession_t Struct Reference . . . 1384
8.1406.1 Detailed Description . . . 1384
8.1406.2 Field Documentation . . . 1385
8.1406.2.1 pFailureReason . . . 1385
8.1406.2.2 psid . . . 1385
8.1406.2.3 pVerboseFailReasonType . . . 1385
8.1406.2.4 pVerboseFailureReason . . . 1385
8.1407 npack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 1385
8.1407.1 Detailed Description . . . 1385
8.1407.2 Field Documentation . . . 1387
8.1407.2.1 apnName . . . 1387
8.1407.2.2 bearerId . . . 1387
8.1407.2.3 contextId . . . 1387
8.1407.2.4 ipv4Address . . . 1387
8.1407.2.5 pv4GWAddress . . . 1387
8.1407.2.6 pv6Address . . . 1387
8.1407.2.7 pv6GWAddress . . . 1387
8.1407.2.8 prDNSIPv4Address . . . 1387

8.1407.2.9 prDNSIPv6Address . . . 1387
8.1407.2.10 prPCSCFIPv4Address . . . 1387
8.1407.2.11 prPCSCFIPv6Address . . . 1387
8.1407.2.12 seDNSIPv4Address . . . 1387
8.1407.2.13 seDNSIPv6Address . . . 1387
8.1407.2.14 sePCSCFIPv4Address . . . 1387
8.1407.2.15 sePCSCFIPv6Address . . . 1387
8.1408 UnPackGetProfileSettingOut Struct Reference . . . 1387
8.1408.1 Detailed Description . . . 1387
8.1408.2 Field Documentation . . . 1388
8.1408.2.1 curProfile . . . 1388
8.1408.2.2 pExtErrCode . . . 1388
8.1409 UnpackWdsProfileParam Union Reference . . . 1388
8.1409.1 Detailed Description . . . 1388
8.1409.2 Field Documentation . . . 1388
8.1409.2.1 SlqsProfile3GPP . . . 1388
8.1409.2.2 SlqsProfile3GPP2 . . . 1388
8.1410 UrlAddr Struct Reference . . . 1388
8.1410.1 Detailed Description . . . 1389
8.1410.2 Field Documentation . . . 1389
8.1410.2.1 address . . . 1389
8.1410.2.2 TlvPresent . . . 1389
8.1411 URLAddrInfo Struct Reference . . . 1389
8.1411.1 Detailed Description . . . 1389
8.1411.2 Field Documentation . . . 1389
8.1411.2.1 urlAddr . . . 1389
8.1412 USBCompConfig Struct Reference . . . 1389
8.1412.1 Detailed Description . . . 1389
8.1412.2 Field Documentation . . . 1390
8.1412.2.1 pUSBComp . . . 1390
8.1413 USBCompParams Struct Reference . . . 1390
8.1413.1 Detailed Description . . . 1390
8.1413.2 Field Documentation . . . 1392
8.1413.2.1 pNumSupUSBComps . . . 1392
8.1413.2.2 pSupUSBComps . . . 1392
8.1413.2.3 pUSBComp . . . 1392
8.1414 usbMTUSizeTlv Struct Reference . . . 1392
8.1414.1 Detailed Description . . . 1392
8.1414.2 Field Documentation . . . 1393
8.1414.2.1 TlvPresent . . . 1393

8.1414.2.2 UsbMTUSize . . . 1393
8.1415 USSDNoWaitIndicationInfo Struct Reference . . . 1393
8.1415.1 Detailed Description . . . 1393
8.1415.2 Field Documentation . . . 1393
8.1415.2.1 pAlphaIdentifier . . . 1393
8.1415.2.2 pError . . . 1393
8.1415.2.3 pFailureCause . . . 1393
8.1415.2.4 pUSSDData . . . 1393
8.1416 USSDRespFNetwork Struct Reference . . . 1393
8.1416.1 Detailed Description . . . 1394
8.1416.2 Field Documentation . . . 1394
8.1416.2.1 pRespData . . . 1394
8.1416.2.2 pTypeCode . . . 1394
8.1417 USSInfo Struct Reference . . . 1394
8.1417.1 Detailed Description . . . 1394
8.1417.2 Field Documentation . . . 1394
8.1417.2.1 ussData . . . 1394
8.1417.2.2 ussDCS . . . 1394
8.1417.2.3 ussLen . . . 1395
8.1418 USSResp Struct Reference . . . 1395
8.1418.1 Field Documentation . . . 1395
8.1418.1.1 pAlphaIDInfo . . . 1395
8.1418.1.2 pCallId . . . 1395
8.1418.1.3 pCcResultType . . . 1395
8.1418.1.4 pCCSuppsType . . . 1395
8.1418.1.5 pfailureCause . . . 1395
8.1418.1.6 pUSSDInfo . . . 1395
8.1419 UUSInfo Struct Reference . . . 1395
8.1419.1 Detailed Description . . . 1395
8.1419.2 Field Documentation . . . 1396
8.1419.2.1 UUSData . . . 1396
8.1419.2.2 UUSDatalen . . . 1396
8.1419.2.3 UUSDcs . . . 1396
8.1419.2.4 UUSType . . . 1396
8.1420 VerifyUIMPIN Struct Reference . . . 1396
8.1420.1 Detailed Description . . . 1396
8.1420.2 Field Documentation . . . 1397
8.1420.2.1 pinID . . . 1397
8.1420.2.2 pinLen . . . 1397
8.1420.2.3 pinVal . . . 1397

8.1421 voice_airTimer Struct Reference . . . 1397
8.1421.1 Detailed Description . . . 1397
8.1421.2 Field Documentation . . . 1397
8.1421.2.1 airTimerValue . . . 1397
8.1421.2.2 namID . . . 1397
8.1422 voice_allCallsAlphaIDInfo Struct Reference . . . 1397
8.1422.1 Detailed Description . . . 1398
8.1422.2 Field Documentation . . . 1398
8.1422.2.1 AlphaIDInfo . . . 1398
8.1422.2.2 callID . . . 1398
8.1423 voice_allCallsDiagInfo Struct Reference . . . 1398
8.1423.1 Detailed Description . . . 1398
8.1423.2 Field Documentation . . . 1398
8.1423.2.1 callID . . . 1398
8.1423.2.2 DiagInfo . . . 1398
8.1424 voice_allCallsUUSInfo Struct Reference . . . 1398
8.1424.1 Detailed Description . . . 1399
8.1424.2 Field Documentation . . . 1399
8.1424.2.1 callID . . . 1399
8.1424.2.2 uusInfo . . . 1399
8.1425 voice_alphaIDInfo Struct Reference . . . 1399
8.1425.1 Detailed Description . . . 1399
8.1425.2 Field Documentation . . . 1400
8.1425.2.1 alphaDcs . . . 1400
8.1425.2.2 alphaLen . . . 1400
8.1425.2.3 alphaText . . . 1400
8.1426 voice_arrAlertingPattern Struct Reference . . . 1400
8.1426.1 Detailed Description . . . 1400
8.1426.2 Field Documentation . . . 1400
8.1426.2.1 alertingPattern . . . 1400
8.1426.2.2 callID . . . 1400
8.1426.2.3 numInstances . . . 1400
8.1427 voice_arrAlertingType Struct Reference . . . 1400
8.1427.1 Detailed Description . . . 1401
8.1427.2 Field Documentation . . . 1401
8.1427.2.1 AlertingType . . . 1401
8.1427.2.2 callID . . . 1401
8.1427.2.3 numInstances . . . 1401
8.1428 voice_arrAlphaID Struct Reference . . . 1401
8.1428.1 Detailed Description . . . 1401

8.1428.2 Field Documentation . . . 1402
8.1428.2.1 allCallsAlphaIDInfoArr . . . 1402
8.1428.2.2 numInstances . . . 1402
8.1429 voice_arrCalledPartyNum Struct Reference . . . 1402
8.1429.1 Detailed Description . . . 1402
8.1429.2 Field Documentation . . . 1402
8.1429.2.1 CalledPartyNum . . . 1402
8.1429.2.2 numInstances . . . 1402
8.1430 voice_arrCallEndReason Struct Reference . . . 1402
8.1430.1 Detailed Description . . . 1402
8.1430.2 Field Documentation . . . 1403
8.1430.2.1 callEndReason . . . 1403
8.1430.2.2 callID . . . 1403
8.1430.2.3 numInstances . . . 1403
8.1431 voice_arrCallInfo Struct Reference . . . 1403
8.1431.1 Detailed Description . . . 1403
8.1431.2 Field Documentation . . . 1403
8.1431.2.1 getAllCallInfo . . . 1403
8.1431.2.2 numInstances . . . 1403
8.1432 voice_arrConnectPartyNum Struct Reference . . . 1403
8.1432.1 Detailed Description . . . 1404
8.1432.2 Field Documentation . . . 1404
8.1432.2.1 ConnectedPartyNum . . . 1404
8.1432.2.2 numInstances . . . 1404
8.1433 voice_arrDiagInfo Struct Reference . . . 1404
8.1433.1 Detailed Description . . . 1404
8.1433.2 Field Documentation . . . 1404
8.1433.2.1 DiagInfo . . . 1404
8.1433.2.2 numInstances . . . 1404
8.1434 voice_arrRedirPartyNum Struct Reference . . . 1404
8.1434.1 Detailed Description . . . 1405
8.1434.2 Field Documentation . . . 1405
8.1434.2.1 numInstances . . . 1405
8.1434.2.2 RedirPartyNum . . . 1405
8.1435 voice_arrRemotePartyName Struct Reference . . . 1405
8.1435.1 Detailed Description . . . 1405
8.1435.2 Field Documentation . . . 1405
8.1435.2.1 GetAllCallRmtPtyName . . . 1405
8.1435.2.2 numInstances . . . 1406
8.1436 voice_arrRemotePartyNum Struct Reference . . . 1406

8.1436.1 Detailed Description . . . 1406
8.1436.2 Field Documentation . . . 1406
8.1436.2.1 numInstances . . . 1406
8.1436.2.2 RmtPtyNum . . . 1406
8.1437 voice_arrSvcOption Struct Reference . . . 1406
8.1437.1 Detailed Description . . . 1406
8.1437.2 Field Documentation . . . 1407
8.1437.2.1 callID . . . 1407
8.1437.2.2 numInstances . . . 1407
8.1437.2.3 srvOption . . . 1407
8.1438 voice_arrUUSInfo Struct Reference . . . 1407
8.1438.1 Detailed Description . . . 1407
8.1438.2 Field Documentation . . . 1407
8.1438.2.1 AllCallsUUSInfo . . . 1407
8.1438.2.2 numInstances . . . 1407
8.1439 voice_burstDTMFInfo Struct Reference . . . 1407
8.1439.1 Detailed Description . . . 1408
8.1439.2 Field Documentation . . . 1408
8.1439.2.1 digitCnt . . . 1408
8.1439.2.2 pCallID . . . 1408
8.1439.2.3 pDigitBuff . . . 1408
8.1440 voice_calledPartyInfo Struct Reference . . . 1408
8.1440.1 Detailed Description . . . 1408
8.1440.2 Field Documentation . . . 1409
8.1440.2.1 number . . . 1409
8.1440.2.2 numLen . . . 1409
8.1440.2.3 numPlan . . . 1409
8.1440.2.4 numType . . . 1410
8.1440.2.5 PI . . . 1410
8.1440.2.6 SI . . . 1410
8.1441 voice_calledPartySubAdd Struct Reference . . . 1410
8.1441.1 Detailed Description . . . 1410
8.1441.2 Field Documentation . . . 1410
8.1441.2.1 extBit . . . 1410
8.1441.2.2 oddEvenInd . . . 1410
8.1441.2.3 subAddr . . . 1410
8.1441.2.4 subAddrLen . . . 1410
8.1441.2.5 subAddrType . . . 1411
8.1442 voice_callerIDInfo Struct Reference . . . 1411
8.1442.1 Detailed Description . . . 1411

8.1442.2 Field Documentation . . . 1411
8.1442.2.1 callerID . . . 1411
8.1442.2.2 callerIDLen . . . 1411
8.1442.2.3 PI . . . 1411
8.1443 voice_callFwdTypeAndPlan Struct Reference . . . 1411
8.1443.1 Detailed Description . . . 1411
8.1443.2 Field Documentation . . . 1412
8.1443.2.1 numberPlan . . . 1412
8.1443.2.2 numberType . . . 1412
8.1444 voice_callFWExtInfo Struct Reference . . . 1412
8.1444.1 Detailed Description . . . 1412
8.1444.2 Field Documentation . . . 1414
8.1444.2.1 noReplyTimer . . . 1414
8.1444.2.2 number . . . 1414
8.1444.2.3 numLen . . . 1414
8.1444.2.4 numPlan . . . 1414
8.1444.2.5 numType . . . 1414
8.1444.2.6 PI . . . 1414
8.1444.2.7 SI . . . 1414
8.1444.2.8 SvcClass . . . 1414
8.1444.2.9 SvcStatus . . . 1414
8.1445 voice_callFWInfo Struct Reference . . . 1414
8.1445.1 Detailed Description . . . 1414
8.1445.2 Field Documentation . . . 1415
8.1445.2.1 noReplyTimer . . . 1415
8.1445.2.2 number . . . 1415
8.1445.2.3 numLen . . . 1415
8.1445.2.4 SvcClass . . . 1415
8.1445.2.5 SvcStatus . . . 1415
8.1446 voice_callInfo Struct Reference . . . 1415
8.1446.1 Detailed Description . . . 1415
8.1446.2 Field Documentation . . . 1416
8.1446.2.1 callID . . . 1416
8.1446.2.2 callState . . . 1417
8.1446.2.3 callType . . . 1417
8.1446.2.4 direction . . . 1417
8.1446.2.5 mode . . . 1417
8.1447 voice_callingPartyInfo Struct Reference . . . 1417
8.1447.1 Detailed Description . . . 1417
8.1447.2 Field Documentation . . . 1418

8.1447.2.1 number . . . 1418
8.1447.2.2 numLen . . . 1418
8.1447.2.3 numPlan . . . 1418
8.1447.2.4 numType . . . 1418
8.1447.2.5 PI . . . 1418
8.1447.2.6 SI . . . 1418
8.1448 voice_ccSUPSType Struct Reference . . . 1418
8.1448.1 Detailed Description . . . 1419
8.1448.2 Field Documentation . . . 1419
8.1448.2.1 reason . . . 1419
8.1448.2.2 svcType . . . 1419
8.1449 voice_CLIPResp Struct Reference . . . 1419
8.1449.1 Detailed Description . . . 1419
8.1449.2 Field Documentation . . . 1420
8.1449.2.1 ActiveStatus . . . 1420
8.1449.2.2 ProvisionStatus . . . 1420
8.1450 voice_CLIRResp Struct Reference . . . 1420
8.1450.1 Detailed Description . . . 1420
8.1450.2 Field Documentation . . . 1420
8.1450.2.1 ActiveStatus . . . 1420
8.1450.2.2 ProvisionStatus . . . 1420
8.1451 voice_CNAPResp Struct Reference . . . 1421
8.1451.1 Detailed Description . . . 1421
8.1451.2 Field Documentation . . . 1421
8.1451.2.1 ActiveStatus . . . 1421
8.1451.2.2 ProvisionStatus . . . 1421
8.1452 voice_COLPResp Struct Reference . . . 1421
8.1452.1 Detailed Description . . . 1421
8.1452.2 Field Documentation . . . 1422
8.1452.2.1 ActiveStatus . . . 1422
8.1452.2.2 ProvisionStatus . . . 1422
8.1453 voice_COLRResp Struct Reference . . . 1422
8.1453.1 Detailed Description . . . 1422
8.1453.2 Field Documentation . . . 1423
8.1453.2.1 ActiveStatus . . . 1423
8.1453.2.2 ProvisionStatus . . . 1423
8.1454 voice_connectNumInfo Struct Reference . . . 1423
8.1454.1 Detailed Description . . . 1423
8.1454.2 Field Documentation . . . 1424
8.1454.2.1 callerID . . . 1424

8.1454.2.2 callerIDLen . . . 1424
8.1454.2.3 numPlan . . . 1424
8.1454.2.4 numPresInd . . . 1424
8.1454.2.5 numType . . . 1424
8.1454.2.6 screeningInd . . . 1424
8.1455 voice_CUGInfo Struct Reference . . . 1424
8.1455.1 Detailed Description . . . 1425
8.1455.2 Field Documentation . . . 1425
8.1455.2.1 CUGIndex . . . 1425
8.1455.2.2 SuppOA . . . 1425
8.1455.2.3 SuppPrefCUG . . . 1425
8.1456 voice_curAMRConfig Struct Reference . . . 1425
8.1456.1 Detailed Description . . . 1425
8.1456.2 Field Documentation . . . 1426
8.1456.2.1 gsmAmrStat . . . 1426
8.1456.2.2 wcdmaAmrStat . . . 1426
8.1457 voice_diagInfo Struct Reference . . . 1426
8.1457.1 Detailed Description . . . 1426
8.1457.2 Field Documentation . . . 1426
8.1457.2.1 diagInfoLen . . . 1426
8.1457.2.2 diagnosticInfo . . . 1426
8.1458 voice_DTMFInfo Struct Reference . . . 1426
8.1458.1 Detailed Description . . . 1427
8.1458.2 Field Documentation . . . 1427
8.1458.2.1 callID . . . 1427
8.1458.2.2 digitBuff . . . 1427
8.1458.2.3 digitCnt . . . 1427
8.1458.2.4 DTMFEvent . . . 1427
8.1459 voice_DTMFLengths Struct Reference . . . 1427
8.1459.1 Detailed Description . . . 1427
8.1459.2 Field Documentation . . . 1428
8.1459.2.1 DTMFInterdigitInterval . . . 1428
8.1459.2.2 DTMFPulseWidth . . . 1428
8.1460 voice_ECTNum Struct Reference . . . 1428
8.1460.1 Detailed Description . . . 1428
8.1460.2 Field Documentation . . . 1429
8.1460.2.1 ECTCallState . . . 1429
8.1460.2.2 number . . . 1429
8.1460.2.3 presentationInd . . . 1429
8.1461 voice_extDispRecInfo Struct Reference . . . 1429

8.1461.1 Detailed Description . . . 1429
8.1461.2 Field Documentation . . . 1429
8.1461.2.1 dispType . . . 1429
8.1461.2.2 extDispInfo . . . 1429
8.1461.2.3 extDispInfoLen . . . 1429
8.1462 voice_getAllCallInformation Struct Reference . . . 1429
8.1462.1 Detailed Description . . . 1430
8.1462.2 Field Documentation . . . 1430
8.1462.2.1 ALS . . . 1430
8.1462.2.2 CallInfo . . . 1430
8.1462.2.3 isEmpty . . . 1430
8.1463 voice_getAllCallRmtPtyName Struct Reference . . . 1430
8.1463.1 Detailed Description . . . 1430
8.1463.2 Field Documentation . . . 1430
8.1463.2.1 callID . . . 1431
8.1463.2.2 RemotePartyName . . . 1431
8.1464 voice_getAllCallRmtPtyNum Struct Reference . . . 1431
8.1464.1 Detailed Description . . . 1431
8.1464.2 Field Documentation . . . 1431
8.1464.2.1 callID . . . 1431
8.1464.2.2 RemotePartyNum . . . 1431
8.1465 voice_getCallFWExtInfo Struct Reference . . . 1431
8.1465.1 Detailed Description . . . 1431
8.1465.2 Field Documentation . . . 1432
8.1465.2.1 CallFWExtInfo . . . 1432
8.1465.2.2 numInstances . . . 1432
8.1466 voice_getCallFWInfo Struct Reference . . . 1432
8.1466.1 Detailed Description . . . 1432
8.1466.2 Field Documentation . . . 1432
8.1466.2.1 CallFWInfo . . . 1432
8.1466.2.2 numInstances . . . 1432
8.1467 voice_lineCtrlInfo Struct Reference . . . 1432
8.1467.1 Detailed Description . . . 1432
8.1467.2 Field Documentation . . . 1433
8.1467.2.1 polarityIncluded . . . 1433
8.1467.2.2 pwrDenialTime . . . 1433
8.1467.2.3 revPolarity . . . 1433
8.1467.2.4 toggleMode . . . 1433
8.1468 voice_newPwdData Struct Reference . . . 1433
8.1468.1 Detailed Description . . . 1433

8.1468.2 Field Documentation . . . 1433
8.1468.2.1 newPwd . . . 1433
8.1468.2.2 newPwdAgain . . . 1433
8.1469 voice_NSSAudioCtrl Struct Reference . . . 1434
8.1469.1 Detailed Description . . . 1434
8.1469.2 Field Documentation . . . 1434
8.1469.2.1 downLink . . . 1434
8.1469.2.2 upLink . . . 1434
8.1470 voice_peerNumberInfo Struct Reference . . . 1434
8.1470.1 Detailed Description . . . 1434
8.1470.2 Field Documentation . . . 1436
8.1470.2.1 callID . . . 1436
8.1470.2.2 number . . . 1436
8.1470.2.3 numLen . . . 1436
8.1470.2.4 numPI . . . 1436
8.1470.2.5 numPlan . . . 1436
8.1470.2.6 numSI . . . 1436
8.1470.2.7 numType . . . 1436
8.1471 voice_prefVoiceSO Struct Reference . . . 1436
8.1471.1 Detailed Description . . . 1436
8.1471.2 Field Documentation . . . 1438
8.1471.2.1 evrcCapability . . . 1438
8.1471.2.2 homeOrigVoiceSO . . . 1438
8.1471.2.3 homePageVoiceSO . . . 1438
8.1471.2.4 namID . . . 1438
8.1471.2.5 roamOrigVoiceSO . . . 1438
8.1472 voice_redirNumInfo Struct Reference . . . 1438
8.1472.1 Detailed Description . . . 1438
8.1472.2 Field Documentation . . . 1439
8.1472.2.1 number . . . 1439
8.1472.2.2 numLen . . . 1439
8.1472.2.3 numPlan . . . 1439
8.1472.2.4 numType . . . 1439
8.1472.2.5 PI . . . 1439
8.1472.2.6 reason . . . 1439
8.1472.2.7 SI . . . 1439
8.1473 voice_remotePartyName Struct Reference . . . 1440
8.1473.1 Detailed Description . . . 1440
8.1473.2 Field Documentation . . . 1440
8.1473.2.1 callerName . . . 1440

8.1473.2.2 codingScheme . . . 1440
8.1473.2.3 nameLen . . . 1440
8.1473.2.4 namePI . . . 1440
8.1474 voice_remotePartyNum Struct Reference . . . 1440
8.1474.1 Detailed Description . . . 1441
8.1474.2 Field Documentation . . . 1441
8.1474.2.1 numLen . . . 1441
8.1474.2.2 presentationInd . . . 1441
8.1474.2.3 remPartyNumber . . . 1441
8.1475 voice_roamTimer Struct Reference . . . 1441
8.1475.1 Detailed Description . . . 1441
8.1475.2 Field Documentation . . . 1442
8.1475.2.1 namID . . . 1442
8.1475.2.2 roamTimerValue . . . 1442
8.1476 voice_signalInfo Struct Reference . . . 1442
8.1476.1 Detailed Description . . . 1442
8.1476.2 Field Documentation . . . 1442
8.1476.2.1 alertPitch . . . 1442
8.1476.2.2 signal . . . 1442
8.1476.2.3 signalType . . . 1442
8.1477 voice_SUPSInfo Struct Reference . . . 1443
8.1477.1 Detailed Description . . . 1443
8.1477.2 Field Documentation . . . 1443
8.1477.2.1 isModByCC . . . 1443
8.1477.2.2 svcType . . . 1443
8.1478 voice_USSDNotificationNetworkInfo Struct Reference . . . 1443
8.1478.1 Detailed Description . . . 1443
8.1478.2 Field Documentation . . . 1444
8.1478.2.1 networkInfo . . . 1444
8.1478.2.2 lvPresent . . . 1444
8.1479 voice_USSInfo Struct Reference . . . 1444
8.1479.1 Detailed Description . . . 1444
8.1479.2 Field Documentation . . . 1444
8.1479.2.1 ussData . . . 1444
8.1479.2.2 ussDCS . . . 1444
8.1479.2.3 ussLen . . . 1444
8.1480 voice_UUSInfo Struct Reference . . . 1444
8.1480.1 Detailed Description . . . 1445
8.1480.2 Field Documentation . . . 1445
8.1480.2.1 UUSData . . . 1445

8.1480.2.2 UUSDatalen . . . 1445
8.1480.2.3 UUSDcs . . . 1445
8.1480.2.4 UUSType . . . 1445
8.1481 VoiceALSSelectLineInfo Struct Reference . . . 1446
8.1481.1 Detailed Description . . . 1446
8.1481.2 Field Documentation . . . 1446
8.1481.2.1 lineValue . . . 1446
8.1482 VoiceALSSetLineSwitchInfo Struct Reference . . . 1446
8.1482.1 Detailed Description . . . 1446
8.1482.2 Field Documentation . . . 1446
8.1482.2.1 switchOption . . . 1446
8.1483 VoiceAnswerCall Struct Reference . . . 1446
8.1483.1 Detailed Description . . . 1447
8.1483.2 Field Documentation . . . 1447
8.1483.2.1 pCallId . . . 1447
8.1484 VoiceBindSubscriptionInfo Struct Reference . . . 1447
8.1484.1 Detailed Description . . . 1447
8.1484.2 Field Documentation . . . 1447
8.1484.2.1 subsType . . . 1447
8.1485 VoiceBurstDTMFInfo Struct Reference . . . 1447
8.1485.1 Detailed Description . . . 1447
8.1485.2 Field Documentation . . . 1448
8.1485.2.1 BurstDTMFInfo . . . 1448
8.1485.2.2 pBurstDTMFLengths . . . 1448
8.1486 VoiceCallInfoReq Struct Reference . . . 1448
8.1486.1 Detailed Description . . . 1448
8.1486.2 Field Documentation . . . 1448
8.1486.2.1 callID . . . 1448
8.1487 VoiceCallInfoResp Struct Reference . . . 1448
8.1487.1 Detailed Description . . . 1448
8.1487.2 Field Documentation . . . 1450
8.1487.2.1 pAlertingPattern . . . 1450
8.1487.2.2 pAlertType . . . 1450
8.1487.2.3 pAlphaIDInfo . . . 1450
8.1487.2.4 pCallInfo . . . 1450
8.1487.2.5 pConnectNumInfo . . . 1450
8.1487.2.6 pDiagInfo . . . 1450
8.1487.2.7 pOTASPStatus . . . 1450
8.1487.2.8 pRemotePartyName . . . 1450
8.1487.2.9 pRemotePartyNum . . . 1451

8.1487.2.10 pSrvOpt . . . 1451
8.1487.2.11 pUUSInfo . . . 1451
8.1487.2.12 pVoicePrivacy . . . 1451
8.1488 VoiceCallRequestParams Struct Reference . . . 1451
8.1488.1 Detailed Description . . . 1451
8.1488.2 Field Documentation . . . 1452
8.1488.2.1 callNumber . . . 1452
8.1488.2.2 pCallPartySubAdd . . . 1452
8.1488.2.3 pCallType . . . 1452
8.1488.2.4 pCLIRType . . . 1452
8.1488.2.5 pCUGInfo . . . 1452
8.1488.2.6 pEmergencyCategory . . . 1452
8.1488.2.7 pSvcType . . . 1452
8.1488.2.8 pUUSInFo . . . 1452
8.1489 VoiceCallResponseParams Struct Reference . . . 1452
8.1489.1 Detailed Description . . . 1453
8.1489.2 Field Documentation . . . 1453
8.1489.2.1 pAlphaIDInfo . . . 1453
8.1489.2.2 pCallID . . . 1453
8.1489.2.3 pCCResultType . . . 1453
8.1489.2.4 pCCSUPSType . . . 1453
8.1490 VoiceContDTMFinfo Struct Reference . . . 1453
8.1490.1 Detailed Description . . . 1453
8.1490.2 Field Documentation . . . 1454
8.1490.2.1 DTMFdigit . . . 1454
8.1490.2.2 pCallID . . . 1454
8.1491 VoiceDTMFEventInfo Struct Reference . . . 1454
8.1491.1 Detailed Description . . . 1454
8.1491.2 Field Documentation . . . 1455
8.1491.2.1 DTMFInformation . . . 1455
8.1491.2.2 pOffLength . . . 1455
8.1491.2.3 pOnLength . . . 1455
8.1492 VoiceFlashInfo Struct Reference . . . 1455
8.1492.1 Detailed Description . . . 1455
8.1492.2 Field Documentation . . . 1455
8.1492.2.1 pCallID . . . 1455
8.1492.2.2 pFlashPayLd . . . 1455
8.1492.2.3 pFlashType . . . 1455
8.1493 VoiceGetAllCallInfo Struct Reference . . . 1455
8.1493.1 Detailed Description . . . 1456

8.1493.2 Field Documentation . . . 1458
8.1493.2.1 pArrAlertingPattern . . . 1458
8.1493.2.2 pArrAlertingType . . . 1458
8.1493.2.3 pArrAlphaID . . . 1458
8.1493.2.4 pArrCalledPartyNum . . . 1458
8.1493.2.5 pArrCallEndReason . . . 1458
8.1493.2.6 pArrCallInfo . . . 1458
8.1493.2.7 pArrConnectPartyNum . . . 1458
8.1493.2.8 pArrDiagInfo . . . 1458
8.1493.2.9 pArrRedirPartyNum . . . 1458
8.1493.2.10 pArrRemotePartyName . . . 1458
8.1493.2.11 pArrRemotePartyNum . . . 1458
8.1493.2.12 pArrSvcOption . . . 1458
8.1493.2.13 pArrUUSInfo . . . 1458
8.1493.2.14 pOTASPStatus . . . 1458
8.1493.2.15 pVoicePrivacy . . . 1458
8.1494 VoiceGetCallBarringReq Struct Reference . . . 1458
8.1494.1 Detailed Description . . . 1458
8.1494.2 Field Documentation . . . 1459
8.1494.2.1 pSvcClass . . . 1459
8.1494.2.2 reason . . . 1459
8.1495 VoiceGetCallBarringResp Struct Reference . . . 1459
8.1495.1 Detailed Description . . . 1459
8.1495.2 Field Documentation . . . 1460
8.1495.2.1 pAlphaIDInfo . . . 1460
8.1495.2.2 pCallID . . . 1460
8.1495.2.3 pCCResType . . . 1460
8.1495.2.4 pCCSUPSType . . . 1460
8.1495.2.5 pFailCause . . . 1460
8.1495.2.6 pSvcClass . . . 1460
8.1496 VoiceGetCallFWReq Struct Reference . . . 1460
8.1496.1 Detailed Description . . . 1461
8.1496.2 Field Documentation . . . 1461
8.1496.2.1 pSvcClass . . . 1461
8.1496.2.2 Reason . . . 1461
8.1497 VoiceGetCallFWResp Struct Reference . . . 1461
8.1497.1 Detailed Description . . . 1461
8.1497.2 Field Documentation . . . 1462
8.1497.2.1 pAlphaIDInfo . . . 1462
8.1497.2.2 pCallID . . . 1462

8.1497.2.3 pCCResType . . . 1462
8.1497.2.4 pCCSUPSType . . . 1462
8.1497.2.5 pFailCause . . . 1462
8.1497.2.6 pGetCallFWExtInfo . . . 1462
8.1497.2.7 pGetCallFWInfo . . . 1462
8.1498 VoiceGetCallWaitInfo Struct Reference . . . 1462
8.1498.1 Detailed Description . . . 1463
8.1498.2 Field Documentation . . . 1463
8.1498.2.1 pAlphaIDInfo . . . 1464
8.1498.2.2 pCallID . . . 1464
8.1498.2.3 pCCResType . . . 1464
8.1498.2.4 pCCSUPSType . . . 1464
8.1498.2.5 pFailCause . . . 1464
8.1498.2.6 pSvcClass . . . 1464
8.1499 VoiceGetCLIPResp Struct Reference . . . 1464
8.1499.1 Detailed Description . . . 1464
8.1499.2 Field Documentation . . . 1465
8.1499.2.1 pAlphaIDInfo . . . 1465
8.1499.2.2 pCallID . . . 1465
8.1499.2.3 pCCResType . . . 1465
8.1499.2.4 pCCSUPSType . . . 1465
8.1499.2.5 pCLIPResp . . . 1465
8.1499.2.6 pFailCause . . . 1465
8.1500 VoiceGetCLIRResp Struct Reference . . . 1465
8.1500.1 Detailed Description . . . 1465
8.1500.2 Field Documentation . . . 1466
8.1500.2.1 pAlphaIDInfo . . . 1466
8.1500.2.2 pCallID . . . 1466
8.1500.2.3 pCCResType . . . 1466
8.1500.2.4 pCCSUPSType . . . 1466
8.1500.2.5 pCLIRResp . . . 1466
8.1500.2.6 pFailCause . . . 1466
8.1501 VoiceGetCNAPResp Struct Reference . . . 1466
8.1501.1 Detailed Description . . . 1466
8.1501.2 Field Documentation . . . 1467
8.1501.2.1 pAlphaIDInfo . . . 1467
8.1501.2.2 pCallID . . . 1467
8.1501.2.3 pCCResType . . . 1467
8.1501.2.4 pCCSUPSType . . . 1467
8.1501.2.5 pCNAPResp . . . 1467

8.1501.2.6 pFailCause . . . 1467
8.1502 VoiceGetCOLPResp Struct Reference . . . 1467
8.1502.1 Detailed Description . . . 1468
8.1502.2 Field Documentation . . . 1468
8.1502.2.1 pAlphaIDInfo . . . 1468
8.1502.2.2 pCallID . . . 1469
8.1502.2.3 pCCResType . . . 1469
8.1502.2.4 pCCSUPSType . . . 1469
8.1502.2.5 pCOLPResp . . . 1469
8.1502.2.6 pFailCause . . . 1469
8.1503 VoiceGetCOLRResp Struct Reference . . . 1469
8.1503.1 Detailed Description . . . 1469
8.1503.2 Field Documentation . . . 1470
8.1503.2.1 pAlphaIDInfo . . . 1470
8.1503.2.2 pCallID . . . 1470
8.1503.2.3 pCCResType . . . 1470
8.1503.2.4 pCCSUPSType . . . 1470
8.1503.2.5 pCOLRResp . . . 1470
8.1503.2.6 pFailCause . . . 1470
8.1504 VoiceGetConfigReq Struct Reference . . . 1470
8.1504.1 Detailed Description . . . 1470
8.1504.2 Field Documentation . . . 1471
8.1504.2.1 pAirTimer . . . 1471
8.1504.2.2 pAMRStatus . . . 1471
8.1504.2.3 pAutoAnswer . . . 1471
8.1504.2.4 pNamID . . . 1471
8.1504.2.5 pPrefVoicePrivacy . . . 1471
8.1504.2.6 pPrefVoiceSO . . . 1471
8.1504.2.7 pRoamTimer . . . 1472
8.1504.2.8 pTTYMode . . . 1472
8.1504.2.9 pVoiceDomainPref . . . 1472
8.1505 VoiceGetConfigResp Struct Reference . . . 1472
8.1505.1 Detailed Description . . . 1472
8.1505.2 Field Documentation . . . 1473
8.1505.2.1 pAirTimerCnt . . . 1473
8.1505.2.2 pAutoAnswerStat . . . 1473
8.1505.2.3 pCurAMRConfig . . . 1473
8.1505.2.4 pCurPrefVoiceSO . . . 1473
8.1505.2.5 pCurrTTYMode . . . 1473
8.1505.2.6 pCurVoiceDomainPref . . . 1473

8.1505.2.7 pCurVoicePrivacyPref . . . 1473
8.1505.2.8 pRoamTimerCnt . . . 1473
8.1506 VoiceIndicationRegisterInfo Struct Reference . . . 1473
8.1506.1 Detailed Description . . . 1474
8.1506.2 Field Documentation . . . 1474
8.1506.2.1 pRegDTMFEvents . . . 1474
8.1506.2.2 pRegVoicePrivacyEvents . . . 1474
8.1506.2.3 pSuppsNotifEvents . . . 1474
8.1507 VoiceInfoRec Struct Reference . . . 1474
8.1507.1 Detailed Description . . . 1475
8.1507.2 Field Documentation . . . 1476
8.1507.2.1 callID . . . 1476
8.1507.2.2 pCalledPartyInfo . . . 1476
8.1507.2.3 pCallerIDInfo . . . 1476
8.1507.2.4 pCallerNameInfo . . . 1476
8.1507.2.5 pCallingPartyInfo . . . 1476
8.1507.2.6 pCallWaitInd . . . 1476
8.1507.2.7 pCLIRCause . . . 1476
8.1507.2.8 pConnectNumInfo . . . 1476
8.1507.2.9 pDispInfo . . . 1476
8.1507.2.10 pExtDispInfo . . . 1476
8.1507.2.11 pExtDispRecInfo . . . 1476
8.1507.2.12 pLineCtrlInfo . . . 1476
8.1507.2.13 pNSSAudioCtrl . . . 1476
8.1507.2.14 pNSSRelease . . . 1476
8.1507.2.15 pRedirNumInfo . . . 1477
8.1507.2.16 pSignalInfo . . . 1477
8.1508 VoiceManageCallsReq Struct Reference . . . 1477
8.1508.1 Detailed Description . . . 1477
8.1508.2 Field Documentation . . . 1477
8.1508.2.1 pCallID . . . 1477
8.1508.2.2 SUPSType . . . 1477
8.1509 VoiceManageCallsResp Struct Reference . . . 1477
8.1509.1 Detailed Description . . . 1478
8.1509.2 Field Documentation . . . 1478
8.1509.2.1 pFailCause . . . 1478
8.1510 VoiceOrigUSSDNoWaitInfo Struct Reference . . . 1478
8.1510.1 Detailed Description . . . 1478
8.1510.2 Field Documentation . . . 1478
8.1510.2.1 USSInformation . . . 1478

8.1511 VoiceOTASPStatusInfo Struct Reference . . . 1478
8.1511.1 Detailed Description . . . 1478
8.1511.2 Field Documentation . . . 1479
8.1511.2.1 callID . . . 1479
8.1511.2.2 OTASPStatus . . . 1479
8.1512 VoicePrivacyInfo Struct Reference . . . 1479
8.1512.1 Detailed Description . . . 1479
8.1512.2 Field Documentation . . . 1480
8.1512.2.1 callID . . . 1480
8.1512.2.2 voicePrivacy . . . 1480
8.1513 VoiceSetAllCallStatusCbkInfo Struct Reference . . . 1480
8.1513.1 Detailed Description . . . 1480
8.1513.2 Field Documentation . . . 1482
8.1513.2.1 arrCallInfomation . . . 1482
8.1513.2.2 pArrAlertingPattern . . . 1482
8.1513.2.3 pArrAlertingType . . . 1482
8.1513.2.4 pArrAlphaID . . . 1482
8.1513.2.5 pArrCalledPartyNum . . . 1482
8.1513.2.6 pArrCallEndReason . . . 1482
8.1513.2.7 pArrConnectPartyNum . . . 1482
8.1513.2.8 pArrDiagInfo . . . 1482
8.1513.2.9 pArrRedirPartyNum . . . 1482
8.1513.2.10 pArrRemotePartyName . . . 1482
8.1513.2.11 pArrRemotePartyNum . . . 1482
8.1513.2.12 pArrSvcOption . . . 1482
8.1514 VoiceSetCallBarringPwdInfo Struct Reference . . . 1482
8.1514.1 Detailed Description . . . 1482
8.1514.2 Field Documentation . . . 1483
8.1514.2.1 newPasswd . . . 1483
8.1514.2.2 newPasswdAgain . . . 1483
8.1514.2.3 oldPasswd . . . 1483
8.1514.2.4 Reason . . . 1483
8.1515 VoiceSetCallBarringPwdResp Struct Reference . . . 1483
8.1515.1 Detailed Description . . . 1484
8.1515.2 Field Documentation . . . 1484
8.1515.2.1 pAlphaIDInfo . . . 1484
8.1515.2.2 pCallID . . . 1484
8.1515.2.3 pCCResType . . . 1484
8.1515.2.4 pCCSUPSType . . . 1484
8.1515.2.5 pFailCause . . . 1484

8.1516 VoiceSetConfigReq Struct Reference . . . 1484
8.1516.1 Detailed Description . . . 1485
8.1516.2 Field Documentation . . . 1485
8.1516.2.1 pAirTimerConfig . . . 1486
8.1516.2.2 pAutoAnswer . . . 1486
8.1516.2.3 pPrefVoiceDomain . . . 1486
8.1516.2.4 pPrefVoiceSO . . . 1486
8.1516.2.5 pRoamTimerConfig . . . 1486
8.1516.2.6 pTTYMode . . . 1486
8.1517 VoiceSetConfigResp Struct Reference . . . 1486
8.1517.1 Detailed Description . . . 1486
8.1517.2 Field Documentation . . . 1487
8.1517.2.1 pAirTimerStatus . . . 1487
8.1517.2.2 pAutoAnsStatus . . . 1487
8.1517.2.3 pPrefVoiceSOStatus . . . 1487
8.1517.2.4 pRoamTimerStatus . . . 1487
8.1517.2.5 pTTYConfigStatus . . . 1487
8.1517.2.6 pVoiceDomainPrefStatus . . . 1487
8.1518 VoiceSetPrefPrivacy Struct Reference . . . 1487
8.1518.1 Detailed Description . . . 1487
8.1518.2 Field Documentation . . . 1488
8.1518.2.1 privacyPref . . . 1488
8.1519 VoiceSetSUPSServiceReq Struct Reference . . . 1488
8.1519.1 Detailed Description . . . 1488
8.1519.2 Field Documentation . . . 1490
8.1519.2.1 pCallBarringPasswd . . . 1490
8.1519.2.2 pCallForwardingNumber . . . 1490
8.1519.2.3 pCallFwdTypeAndPlan . . . 1490
8.1519.2.4 pServiceClass . . . 1490
8.1519.2.5 pTimerVal . . . 1490
8.1519.2.6 reason . . . 1490
8.1519.2.7 voiceSvc . . . 1490
8.1520 VoiceSetSUPSServiceResp Struct Reference . . . 1490
8.1520.1 Detailed Description . . . 1490
8.1520.2 Field Documentation . . . 1491
8.1520.2.1 pAlphaIDInfo . . . 1491
8.1520.2.2 pCallID . . . 1491
8.1520.2.3 pCCResultType . . . 1491
8.1520.2.4 pCCSUPSType . . . 1491
8.1520.2.5 pFailCause . . . 1491

8.1521 VoiceStopContDTMFInfo Struct Reference . . . 1491
8.1521.1 Detailed Description . . . 1491
8.1521.2 Field Documentation . . . 1492
8.1521.2.1 callID . . . 1492
8.1522 VoiceSUPSInfo Struct Reference . . . 1492
8.1522.1 Detailed Description . . . 1492
8.1522.2 Field Documentation . . . 1494
8.1522.2.1 pAlphaIDInfo . . . 1494
8.1522.2.2 pCallBarPasswd . . . 1494
8.1522.2.3 pCallFwdInfo . . . 1494
8.1522.2.4 pCallFWNum . . . 1494
8.1522.2.5 pCallFWTimerVal . . . 1494
8.1522.2.6 pCallID . . . 1494
8.1522.2.7 pCLIPstatus . . . 1494
8.1522.2.8 pCLIRstatus . . . 1494
8.1522.2.9 pCNAPstatus . . . 1494
8.1522.2.10 pCOLPstatus . . . 1494
8.1522.2.11 pCOLRstatus . . . 1494
8.1522.2.12 pDataSrc . . . 1494
8.1522.2.13 pFailCause . . . 1494
8.1522.2.14 pNewPwdData . . . 1494
8.1522.2.15 pReason . . . 1494
8.1522.2.16 pSvcClass . . . 1494
8.1522.2.17 pUSSInfo . . . 1494
8.1522.2.18 SUPSInformation . . . 1494
8.1523 VoiceSUPSNotification Struct Reference . . . 1494
8.1523.1 Detailed Description . . . 1495
8.1523.2 Field Documentation . . . 1496
8.1523.2.1 callID . . . 1496
8.1523.2.2 notifType . . . 1496
8.1523.2.3 pCUGIndex . . . 1496
8.1523.2.4 pECTNum . . . 1496
8.1524 WcdmaCellInfo Struct Reference . . . 1496
8.1524.1 Detailed Description . . . 1496
8.1524.2 Field Documentation . . . 1496
8.1524.2.1 cpich_ecno . . . 1496
8.1524.2.2 cpich_rscp . . . 1496
8.1524.2.3 psc . . . 1497
8.1524.2.4 srxlev . . . 1497
8.1525 WCDMAECIOThresh Struct Reference . . . 1497

8.1525.1 Detailed Description . . . 1497
8.1525.2 Field Documentation . . . 1497
8.1525.2.1 pWCDMAECIOThreshList . . . 1497
8.1525.2.2 WCDMAECIOThreshListLen . . . 1497
8.1526 WCDMAInfoLTENeighborCell Struct Reference . . . 1497
8.1526.1 Detailed Description . . . 1497
8.1526.2 Field Documentation . . . 1498
8.1526.2.1 UMTSLTENbrCell . . . 1498
8.1526.2.2 umtsLTENbrCellLen . . . 1498
8.1526.2.3 wcdmaRRCState . . . 1498
8.1527 wcdmaLongMsgDecodingParams Struct Reference . . . 1498
8.1527.1 Detailed Description . . . 1498
8.1527.2 Field Documentation . . . 1499
8.1527.2.1 Date . . . 1499
8.1527.2.2 pIsUDHPresent . . . 1500
8.1527.2.3 pMessage . . . 1500
8.1527.2.4 pPartNum . . . 1500
8.1527.2.5 pReferenceNum . . . 1500
8.1527.2.6 pScAddr . . . 1500
8.1527.2.7 pScAddrLength . . . 1500
8.1527.2.8 pSenderAddr . . . 1500
8.1527.2.9 pSenderAddrLength . . . 1500
8.1527.2.10 pTextMsg . . . 1500
8.1527.2.11 pTextMsgLength . . . 1500
8.1527.2.12 pTotalNum . . . 1500
8.1527.2.13 Time . . . 1500
8.1528 wcdmaMsgDecodingParams Struct Reference . . . 1500
8.1528.1 Detailed Description . . . 1500
8.1528.2 Field Documentation . . . 1501
8.1528.2.1 Date . . . 1501
8.1528.2.2 pMessage . . . 1501
8.1528.2.3 pScAddr . . . 1501
8.1528.2.4 pScAddrLength . . . 1501
8.1528.2.5 pSenderAddr . . . 1501
8.1528.2.6 pSenderAddrLength . . . 1501
8.1528.2.7 pTextMsg . . . 1501
8.1528.2.8 pTextMsgLength . . . 1501
8.1528.2.9 Time . . . 1501
8.1529 wcdmaMsgEncodingParams Struct Reference . . . 1501
8.1529.1 Detailed Description . . . 1502

8.1529.2 Field Documentation . . . 1502
8.1529.2.1 alphabet . . . 1502
8.1529.2.2 messageSize . . . 1502
8.1529.2.3 pDestAddr . . . 1502
8.1529.2.4 pPDUMessage . . . 1502
8.1529.2.5 pTextMsg . . . 1502
8.1530 WCDMARSSIThresh Struct Reference . . . 1502
8.1530.1 Detailed Description . . . 1503
8.1530.2 Field Documentation . . . 1503
8.1530.2.1 pWCDMARSSIThreshList . . . 1503
8.1530.2.2 WCDMARSSIThreshListLen . . . 1503
8.1531 WCDMASysInfo Struct Reference . . . 1503
8.1531.1 Detailed Description . . . 1503
8.1531.2 Field Documentation . . . 1506
8.1531.2.1 cellId . . . 1506
8.1531.2.2 cellIdValid . . . 1506
8.1531.2.3 hsCallStatus . . . 1506
8.1531.2.4 hsCallStatusValid . . . 1506
8.1531.2.5 hsInd . . . 1506
8.1531.2.6 hsIndValid . . . 1506
8.1531.2.7 lac . . . 1506
8.1531.2.8 lacValid . . . 1506
8.1531.2.9 MCC . . . 1506
8.1531.2.10 MNC . . . 1506
8.1531.2.11 networkIdValid . . . 1506
8.1531.2.12 psc . . . 1506
8.1531.2.13 pscValid . . . 1506
8.1531.2.14 regRejectInfoValid . . . 1506
8.1531.2.15 rejCause . . . 1506
8.1531.2.16 rejectSrvDomain . . . 1506
8.1531.2.17 sysInfoWCDMA . . . 1506
8.1532 wcdmaUARFCN Struct Reference . . . 1506
8.1532.1 Detailed Description . . . 1506
8.1532.2 Field Documentation . . . 1507
8.1532.2.1 status . . . 1507
8.1532.2.2 uarfcn . . . 1507
8.1533 wds_channelRate Struct Reference . . . 1507
8.1533.1 Detailed Description . . . 1507
8.1533.2 Field Documentation . . . 1507
8.1533.2.1 CurrChanRxRate . . . 1507

8.1533.2.2 CurrChanTxRate . . . 1507
8.1534 wds_ChannelRateTlv Struct Reference . . . 1507
8.1534.1 Detailed Description . . . 1507
8.1534.2 Field Documentation . . . 1508
8.1534.2.1 ChannelRate . . . 1508
8.1534.2.2 TlvPresent . . . 1508
8.1535 wds_ConnStatusTlv Struct Reference . . . 1508
8.1535.1 Detailed Description . . . 1508
8.1535.2 Field Documentation . . . 1508
8.1535.2.1 MDMConnStatus . . . 1508
8.1535.2.2 TlvPresent . . . 1508
8.1536 wds_currNetworkInfo Struct Reference . . . 1508
8.1536.1 Detailed Description . . . 1508
8.1536.2 Field Documentation . . . 1510
8.1536.2.1 NetworkType . . . 1510
8.1536.2.2 RATMask . . . 1510
8.1536.2.3 SOMask . . . 1510
8.1537 wds_DataBearTechTlv Struct Reference . . . 1510
8.1537.1 Detailed Description . . . 1510
8.1537.2 Field Documentation . . . 1511
8.1537.2.1 DataBearerTech . . . 1511
8.1537.2.2 TlvPresent . . . 1511
8.1538 wds_DataULongLongTlv Struct Reference . . . 1511
8.1538.1 Detailed Description . . . 1511
8.1538.2 Field Documentation . . . 1511
8.1538.2.1 TlvPresent . . . 1511
8.1538.2.2 ullData . . . 1512
8.1539 wds_DataULongTlv Struct Reference . . . 1512
8.1539.1 Detailed Description . . . 1512
8.1539.2 Field Documentation . . . 1512
8.1539.2.1 TlvPresent . . . 1512
8.1539.2.2 ulData . . . 1512
8.1540 wds_DHCPLeaseOptTlv Struct Reference . . . 1512
8.1540.1 Detailed Description . . . 1512
8.1540.2 Field Documentation . . . 1512
8.1540.2.1 numOpt . . . 1512
8.1540.2.2 optList . . . 1512
8.1540.2.3 optListData . . . 1512
8.1540.2.4 TlvPresent . . . 1513
8.1541 wds_DHCPLeaseStateTlv Struct Reference . . . 1513

8.1541.1 Detailed Description . . . 1513
8.1541.2 Field Documentation . . . 1513
8.1541.2.1 leaseState . . . 1513
8.1541.2.2 TlvPresent . . . 1513
8.1542 wds_DHCPOpt Struct Reference . . . 1513
8.1542.1 Detailed Description . . . 1513
8.1542.2 Field Documentation . . . 1514
8.1542.2.1 optCode . . . 1514
8.1542.2.2 optValLen . . . 1514
8.1542.2.3 pOptVal . . . 1514
8.1543 wds_DHCPProfileIdTlv Struct Reference . . . 1514
8.1543.1 Detailed Description . . . 1514
8.1543.2 Field Documentation . . . 1514
8.1543.2.1 profileId . . . 1514
8.1543.2.2 profileType . . . 1514
8.1543.2.3 TlvPresent . . . 1514
8.1544 wds_DHCPv4HWConfig Struct Reference . . . 1514
8.1544.1 Detailed Description . . . 1515
8.1544.2 Field Documentation . . . 1515
8.1544.2.1 chaddr . . . 1515
8.1544.2.2 chaddrLen . . . 1515
8.1544.2.3 hwType . . . 1515
8.1545 wds_DHCPv4Option Struct Reference . . . 1515
8.1545.1 Detailed Description . . . 1515
8.1545.2 Field Documentation . . . 1515
8.1545.2.1 optCode . . . 1515
8.1545.2.2 optVal . . . 1516
8.1545.2.3 optValLen . . . 1516
8.1546 wds_DHCPv4OptionList Struct Reference . . . 1516
8.1546.1 Detailed Description . . . 1516
8.1546.2 Field Documentation . . . 1516
8.1546.2.1 numOpt . . . 1516
8.1546.2.2 pOptList . . . 1516
8.1547 wds_DHCPv4ProfileId Struct Reference . . . 1516
8.1547.1 Detailed Description . . . 1516
8.1547.2 Field Documentation . . . 1516
8.1547.2.1 profileId . . . 1517
8.1547.2.2 profileType . . . 1517
8.1548 wds_Domain Struct Reference . . . 1517
8.1548.1 Detailed Description . . . 1517

8.1548.2 Field Documentation . . . 1517
8.1548.2.1 domainLen . . . 1517
8.1548.2.2 domainName . . . 1517
8.1549 wds_DomainNameList Struct Reference . . . 1517
8.1549.1 Detailed Description . . . 1517
8.1549.2 Field Documentation . . . 1518
8.1549.2.1 domain . . . 1518
8.1549.2.2 numInstances . . . 1518
8.1550 wds_DormStatTlv Struct Reference . . . 1518
8.1550.1 Detailed Description . . . 1518
8.1550.2 Field Documentation . . . 1518
8.1550.2.1 DormancyStat . . . 1518
8.1550.2.2 TlvPresent . . . 1518
8.1551 wds_GPRSQoS Struct Reference . . . 1518
8.1551.1 Detailed Description . . . 1518
8.1551.2 Field Documentation . . . 1519
8.1551.2.1 delayClass . . . 1519
8.1551.2.2 meanThroughputClass . . . 1519
8.1551.2.3 peakThroughputClass . . . 1519
8.1551.2.4 precedenceClass . . . 1519
8.1551.2.5 reliabilityClass . . . 1519
8.1552 wds_IPv4AdTlv Struct Reference . . . 1519
8.1552.1 Detailed Description . . . 1519
8.1552.2 Field Documentation . . . 1519
8.1552.2.1 IPv4Addr . . . 1519
8.1552.2.2 TlvPresent . . . 1519
8.1553 wds_IPV6AddressInfo Struct Reference . . . 1520
8.1553.1 Detailed Description . . . 1520
8.1553.2 Field Documentation . . . 1520
8.1553.2.1 IPAddressV6 . . . 1520
8.1553.2.2 IPV6PrefixLen . . . 1520
8.1554 wds_IPV6GWAddressInfo Struct Reference . . . 1520
8.1554.1 Detailed Description . . . 1520
8.1554.2 Field Documentation . . . 1521
8.1554.2.1 gwAddressV6 . . . 1521
8.1554.2.2 gwV6PrefixLen . . . 1521
8.1555 wds_LastMdmCallEndRsnTlv Struct Reference . . . 1521
8.1555.1 Detailed Description . . . 1521
8.1555.2 Field Documentation . . . 1521
8.1555.2.1 CallEndReason . . . 1521

8.1555.2.2 TlvPresent . . . 1521
8.1556 wds_PCSCFFQDNAddress Struct Reference . . . 1521
8.1556.1 Detailed Description . . . 1521
8.1556.2 Field Documentation . . . 1522
8.1556.2.1 fqdnAddr . . . 1522
8.1556.2.2 fqdnLen . . . 1522
8.1557 wds_PCSCFFQDNAddressList Struct Reference . . . 1522
8.1557.1 Detailed Description . . . 1522
8.1557.2 Field Documentation . . . 1522
8.1557.2.1 numInstances . . . 1522
8.1557.2.2 pcsfFQDNAddress . . . 1522
8.1558 wds_PCSCFIPv4ServerAddressList Struct Reference . . . 1522
8.1558.1 Detailed Description . . . 1522
8.1558.2 Field Documentation . . . 1523
8.1558.2.1 numInstances . . . 1523
8.1558.2.2 pscsfIPv4Addr . . . 1523
8.1559 wds_ProfileIdentifier Struct Reference . . . 1523
8.1559.1 Detailed Description . . . 1523
8.1559.2 Field Documentation . . . 1523
8.1559.2.1 profileIndex . . . 1523
8.1559.2.2 profileType . . . 1523
8.1560 wds_profileInfo Union Reference . . . 1523
8.1560.1 Detailed Description . . . 1523
8.1560.2 Field Documentation . . . 1524
8.1560.2.1 SlqsProfile3GPP . . . 1524
8.1560.2.2 SlqsProfile3GPP2 . . . 1524
8.1561 wds_RXBytesOKTlv Struct Reference . . . 1524
8.1561.1 Detailed Description . . . 1524
8.1561.2 Field Documentation . . . 1524
8.1561.2.1 RxByteOKCnt . . . 1524
8.1561.2.2 TlvPresent . . . 1524
8.1562 wds_transferStatInd Struct Reference . . . 1524
8.1562.1 Detailed Description . . . 1524
8.1562.2 Field Documentation . . . 1525
8.1562.2.1 StatsMask . . . 1525
8.1562.2.2 StatsPeriod . . . 1525
8.1563 wds_TrStatInd Struct Reference . . . 1525
8.1563.1 Detailed Description . . . 1525
8.1563.2 Field Documentation . . . 1525
8.1563.2.1 statsMask . . . 1525

8.1563.2.2 statsPeriod . . . 1525
8.1564 wds_TXBytesOKTlv Struct Reference . . . 1526
8.1564.1 Detailed Description . . . 1526
8.1564.2 Field Documentation . . . 1526
8.1564.2.1 TlvPresent . . . 1526
8.1564.2.2 TxByteOKCnt . . . 1526
8.1565 wds_UMTSMinQoS Struct Reference . . . 1526
8.1565.1 Detailed Description . . . 1526
8.1565.2 Field Documentation . . . 1528
8.1565.2.1 deliveryErrSDU . . . 1528
8.1565.2.2 grntDownlinkBitrate . . . 1528
8.1565.2.3 grntUplinkBitrate . . . 1528
8.1565.2.4 maxDownlinkBitrate . . . 1528
8.1565.2.5 maxSDUSize . . . 1528
8.1565.2.6 maxUplinkBitrate . . . 1528
8.1565.2.7 qosDeliveryOrder . . . 1528
8.1565.2.8 resBerRatio . . . 1528
8.1565.2.9 sduErrorRatio . . . 1528
8.1565.2.10 trafficClass . . . 1528
8.1565.2.11 trafficPriority . . . 1528
8.1565.2.12 transferDelay . . . 1528
8.1566 WdsByteTotals Struct Reference . . . 1528
8.1566.1 Detailed Description . . . 1529
8.1566.2 Field Documentation . . . 1529
8.1566.2.1 ByteTotalsElmntsV4 . . . 1529
8.1566.2.2 ByteTotalsElmntsV6 . . . 1529
8.1566.2.3 pV4sessionId . . . 1529
8.1566.2.4 pV6sessionId . . . 1529
8.1567 WdsByteTotalsElmnts Struct Reference . . . 1529
8.1567.1 Detailed Description . . . 1529
8.1567.2 Field Documentation . . . 1530
8.1567.2.1 pRXTotalBytes . . . 1530
8.1567.2.2 pTXTotalBytes . . . 1530
8.1568 WdsClientLeaseChange Struct Reference . . . 1530
8.1568.1 Detailed Description . . . 1530
8.1568.2 Field Documentation . . . 1530
8.1568.2.1 pEnableNotification . . . 1530
8.1569 WdsConnectionRate Struct Reference . . . 1530
8.1569.1 Detailed Description . . . 1530
8.1569.2 Field Documentation . . . 1531

8.1569.2.1 ConnRateElmntsV4 . . . 1531
8.1569.2.2 ConnRateElmntsV6 . . . 1531
8.1569.2.3 pV4sessionId . . . 1531
8.1569.2.4 pV6sessionId . . . 1531
8.1570 WdsConnectionRateElmnts Struct Reference . . . 1531
8.1570.1 Detailed Description . . . 1531
8.1570.2 Field Documentation . . . 1532
8.1570.2.1 pCurrentChannelRXRate . . . 1532
8.1570.2.2 pCurrentChannelTXRate . . . 1532
8.1570.2.3 pMaxChannelRXRate . . . 1532
8.1570.2.4 pMaxChannelTXRate . . . 1532
8.1571 WdsDHCPv4ClientLeaseInd Struct Reference . . . 1532
8.1571.1 Detailed Description . . . 1532
8.1571.2 Field Documentation . . . 1532
8.1571.2.1 pIPv4Addr . . . 1532
8.1571.2.2 pLeaseState . . . 1532
8.1571.2.3 pOptList . . . 1533
8.1571.2.4 pProfileId . . . 1533
8.1572 WdsDHCPv4Config Struct Reference . . . 1533
8.1572.1 Detailed Description . . . 1533
8.1572.2 Field Documentation . . . 1533
8.1572.2.1 pHwConfig . . . 1533
8.1572.2.2 pProfileId . . . 1533
8.1572.2.3 pRequestOptionList . . . 1533
8.1573 WdsDhcpv4HwConfig Struct Reference . . . 1533
8.1573.1 Detailed Description . . . 1533
8.1573.2 Field Documentation . . . 1534
8.1573.2.1 chaddr . . . 1534
8.1573.2.2 chaddrLen . . . 1534
8.1573.2.3 hwType . . . 1534
8.1574 WdsDHCPv4HWConfig Struct Reference . . . 1534
8.1574.1 Detailed Description . . . 1534
8.1574.2 Field Documentation . . . 1534
8.1574.2.1 chaddr . . . 1534
8.1574.2.2 chaddrLen . . . 1534
8.1574.2.3 hwType . . . 1534
8.1575 WdsDHCPv4Option Struct Reference . . . 1534
8.1575.1 Detailed Description . . . 1535
8.1575.2 Field Documentation . . . 1535
8.1575.2.1 optCode . . . 1535

8.1575.2.2 optVal . . . 1535
8.1575.2.3 optValLen . . . 1535
8.1576 WdsDhcpv4Option Struct Reference . . . 1535
8.1576.1 Detailed Description . . . 1535
8.1576.2 Field Documentation . . . 1535
8.1576.2.1 optCode . . . 1535
8.1576.2.2 optVal . . . 1535
8.1576.2.3 optValLen . . . 1536
8.1577 WdsDHCPv4OptionList Struct Reference . . . 1536
8.1577.1 Detailed Description . . . 1536
8.1577.2 Field Documentation . . . 1536
8.1577.2.1 numOpt . . . 1536
8.1577.2.2 pOptList . . . 1536
8.1578 WdsDhcpv4OptionList Struct Reference . . . 1536
8.1578.1 Detailed Description . . . 1536
8.1578.2 Field Documentation . . . 1536
8.1578.2.1 numOpt . . . 1537
8.1578.2.2 pOptList . . . 1537
8.1579 WdsDhcpv4ProfileId Struct Reference . . . 1537
8.1579.1 Detailed Description . . . 1537
8.1579.2 Field Documentation . . . 1537
8.1579.2.1 profileId . . . 1537
8.1579.2.2 profileType . . . 1537
8.1580 WdsDHCPv4ProfileId Struct Reference . . . 1537
8.1580.1 Detailed Description . . . 1537
8.1580.2 Field Documentation . . . 1537
8.1580.2.1 profileId . . . 1538
8.1580.2.2 profileType . . . 1538
8.1581 WDSGetLoopbackData Struct Reference . . . 1538
8.1581.1 Detailed Description . . . 1538
8.1581.2 Field Documentation . . . 1538
8.1581.2.1 ByteLoopbackMode . . . 1538
8.1581.2.2 ByteLoopbackMultiplier . . . 1538
8.1582 WdsIpAddressInfoReq Struct Reference . . . 1538
8.1582.1 Detailed Description . . . 1539
8.1582.2 Field Documentation . . . 1539
8.1582.2.1 ip . . . 1539
8.1582.2.2 pv4sessionId . . . 1539
8.1582.2.3 pv6sessionId . . . 1539
8.1583 WdsPktStatisticsElmnts Struct Reference . . . 1539

8.1583.1 Detailed Description . . . 1539
8.1583.2 Field Documentation . . . 1540
8.1583.2.1 pRXDroppedCount . . . 1540
8.1583.2.2 pRXOkBytesCount . . . 1540
8.1583.2.3 pRXOKBytesLastCall . . . 1540
8.1583.2.4 pRXPacketErrors . . . 1540
8.1583.2.5 pRXPacketOverflows . . . 1540
8.1583.2.6 pRXPacketSuccesses . . . 1541
8.1583.2.7 pTXDroppedCount . . . 1541
8.1583.2.8 pTXOkBytesCount . . . 1541
8.1583.2.9 pTXOKBytesLastCall . . . 1541
8.1583.2.10 pTXPacketErrors . . . 1541
8.1583.2.11 pTXPacketOverflows . . . 1541
8.1583.2.12 pTXPacketSuccesses . . . 1541
8.1584 WdsPktStatisticsReq Struct Reference . . . 1541
8.1584.1 Detailed Description . . . 1541
8.1584.2 Field Documentation . . . 1541
8.1584.2.1 pStatMask . . . 1541
8.1585 WdsPktStatisticsResp Struct Reference . . . 1541
8.1585.1 Detailed Description . . . 1541
8.1585.2 Field Documentation . . . 1542
8.1585.2.1 PktStatElmntsV4 . . . 1542
8.1585.2.2 PktStatElmntsV6 . . . 1542
8.1585.2.3 pV4sessionId . . . 1542
8.1585.2.4 pV6sessionId . . . 1542
8.1586 WdsProfileParam Union Reference . . . 1542
8.1586.1 Detailed Description . . . 1542
8.1586.2 Field Documentation . . . 1542
8.1586.2.1 SlqsProfile3GPP . . . 1543
8.1586.2.2 SlqsProfile3GPP2 . . . 1543
8.1587 WdsRunTimeSettings Struct Reference . . . 1543
8.1587.1 Detailed Description . . . 1543
8.1587.2 Field Documentation . . . 1543
8.1587.2.1 rts . . . 1543
8.1587.2.2 v4sessionId . . . 1543
8.1587.2.3 v6sessionId . . . 1543
8.1588 WdsSetEventReportReq Struct Reference . . . 1543
8.1588.1 Detailed Description . . . 1544
8.1588.2 Field Documentation . . . 1545
8.1588.2.1 pCurrChannelRateInd . . . 1545

8.1588.2.2 pCurrDataBearerTechInd . . . 1545
8.1588.2.3 pCurrPrefDataSysInd . . . 1545
8.1588.2.4 pDataBearerTechInd . . . 1545
8.1588.2.5 pDataCallStatusChangeInd . . . 1545
8.1588.2.6 pDataSystemStatusChangeInd . . . 1545
8.1588.2.7 pDormancyStatusInd . . . 1545
8.1588.2.8 pEVDOPageMonPerChangeInd . . . 1545
8.1588.2.9 pMIPStatusInd . . . 1545
8.1588.2.10 pTransferStatInd . . . 1545
8.1589 WDSSetLoopbackData Struct Reference . . . 1545
8.1589.1 Detailed Description . . . 1545
8.1589.2 Field Documentation . . . 1546
8.1589.2.1 pLoopbackMode . . . 1546
8.1589.2.2 pLoopbackMultiplier . . . 1546
8.1590 WDSSWICurrentChannelRates Struct Reference . . . 1546
8.1590.1 Detailed Description . . . 1546
8.1590.2 Field Documentation . . . 1546
8.1590.2.1 current_channel_rx_rate . . . 1546
8.1590.2.2 current_channel_tx_rate . . . 1546
8.1590.2.3 max_channel_rx_rate . . . 1546
8.1590.2.4 max_channel_tx_rate . . . 1546

**9 File Documentation** **1547**

9.1 apdoxypages.c File Reference . . . 1547
9.1.1 Detailed Description . . . 1547
9.2 common.h File Reference . . . 1547
9.2.1 Macro Definition Documentation . . . 1549
9.2.1.1 DEAULT_LOC_TIMEOUT_IN_SEC . . . 1549
9.2.1.2 MINREQBKLEN . . . 1549
9.2.1.3 MSGID_AND_LEN . . . 1549
9.2.1.4 MSGID_DONT_CARE . . . 1549
9.2.1.5 SDK_VALIDATE_INPUT_PACK_PARAM . . . 1549
9.2.1.6 SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID . . . 1549
9.2.1.7 SDK_VALIDATE_INPUT_UNPACK_PARAM . . . 1549
9.2.1.8 SDU_HDR_LEN . . . 1550
9.2.1.9 UNUSEDPARAM . . . 1550
9.2.2 Typedef Documentation . . . 1550
9.2.2.1 logger . . . 1550
9.2.3 Enumeration Type Documentation . . . 1550
9.2.3.1 eLOG_LEVEL . . . 1550

9.2.3.2 eQMI_SVC . . . 1550
9.2.3.3 eTimeout . . . 1551
9.2.3.4 msgtype . . . 1551
9.2.4 Function Documentation . . . 1551
9.2.4.1 fill_pack_ctx . . . 1551
9.2.4.2 fill_sdu_hdr . . . 1551
9.2.4.3 get_version . . . 1551
9.2.4.4 helper_get_error_code . . . 1551
9.2.4.5 helper_get_error_reason . . . 1551
9.2.4.6 helper_get_req_str . . . 1551
9.2.4.7 helper_get_resp_ctx . . . 1552
9.2.4.8 helper_get_xid . . . 1552
9.2.4.9 helper_set_log_func . . . 1552
9.2.4.10 helper_set_log_lvl . . . 1552
9.2.4.11 liteqmi_GetVersion . . . 1552
9.2.4.12 liteqmi_helper_decode7bitAsciiEncString . . . 1552
9.2.4.13 liteqmi_log . . . 1552
9.2.4.14 unpack_result_code_only . . . 1552
9.2.5 Variable Documentation . . . 1553
9.2.5.1 glog . . . 1553
9.2.5.2 gloglvl . . . 1553
9.3 dms.h File Reference . . . 1553
9.3.1 Macro Definition Documentation . . . 1560
9.3.1.1 ACT_CODE_MAX_SIZE . . . 1560
9.3.1.2 CK_MAX_SIZE . . . 1560
9.3.1.3 DMS_IMGDETAILS_LEN . . . 1560
9.3.1.4 DMS_MAX_CUST_ID_LEN . . . 1560
9.3.1.5 DMS_MAX_CUST_VALUE_LEN . . . 1560
9.3.1.6 DMS_MAX_FWUPDATE_LOG_STR_SZ . . . 1560
9.3.1.7 DMS_MAX_FWUPDATE_REF_STR_SZ . . . 1560
9.3.1.8 DMS_MAX_RADIO_IFCS_SIZE . . . 1560
9.3.1.9 DMS_PM_FACTORY . . . 1560
9.3.1.10 DMS_PM_LOW . . . 1560
9.3.1.11 DMS_PM_OFFLINE . . . 1560
9.3.1.12 DMS_PM_ONLINE . . . 1560
9.3.1.13 DMS_PM_PERSISTENT_LOW . . . 1560
9.3.1.14 DMS_PM_RESET . . . 1560
9.3.1.15 DMS_PM_SHUT_DOWN . . . 1560
9.3.1.16 DMS_SET_REG_IND_DISABLE . . . 1560
9.3.1.17 DMS_SET_REG_IND_ENABLE . . . 1560

9.3.1.18 DMS_SET_REG_IND_NO_CHANGE . . . 1560
9.3.1.19 DMS_SET_REPORT_DISABLE . . . 1560
9.3.1.20 DMS_SET_REPORT_ENABLE . . . 1561
9.3.1.21 DMS_SLQSFWINFO_APPVERSION_SZ . . . 1561
9.3.1.22 DMS_SLQSFWINFO_BOOTVERSION_SZ . . . 1561
9.3.1.23 DMS_SLQSFWINFO_CARRIER_SZ . . . 1561
9.3.1.24 DMS_SLQSFWINFO_CUR_CARR_NAME . . . 1561
9.3.1.25 DMS_SLQSFWINFO_CUR_CARR_REV . . . 1561
9.3.1.26 DMS_SLQSFWINFO_MODELID_SZ . . . 1561
9.3.1.27 DMS_SLQSFWINFO_PACKAGEID_SZ . . . 1561
9.3.1.28 DMS_SLQSFWINFO_PRIVERSION_SZ . . . 1561
9.3.1.29 DMS_SLQSFWINFO_SKU_SZ . . . 1561
9.3.1.30 DMS_SWI_SET_IND_DISABLE . . . 1561
9.3.1.31 DMS_SWI_SET_IND_ENABLE . . . 1561
9.3.1.32 DMS_UINT8_MAX_STRING_SZ . . . 1561
9.3.1.33 DMS_VALID_FSN_LEN . . . 1561
9.3.1.34 ERI_DATA_MAX_SIZE . . . 1561
9.3.1.35 MAX_BUILD_ID_LEN . . . 1561
9.3.1.36 MEID_MAX_SIZE . . . 1561
9.3.1.37 SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1561
9.3.1.38 SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1561
9.3.1.39 SPC_SIZE . . . 1561
9.3.1.40 UNIQUE_ID_LEN . . . 1561
9.3.2 Function Documentation . . . 1561
9.3.2.1 pack_dms_ActivateAutomatic . . . 1561
9.3.2.2 pack_dms_GetActivationState . . . 1562
9.3.2.3 pack_dms_GetBandCapability . . . 1562
9.3.2.4 pack_dms_GetCrashAction . . . 1562
9.3.2.5 pack_dms_GetCustFeature . . . 1563
9.3.2.6 pack_dms_GetCustFeaturesV2 . . . 1563
9.3.2.7 pack_dms_GetDeviceCap . . . 1564
9.3.2.8 pack_dms_GetDeviceCapabilities . . . 1564
9.3.2.9 pack_dms_GetDeviceHardwareRev . . . 1564
9.3.2.10 pack_dms_GetDeviceMfr . . . 1565
9.3.2.11 pack_dms_GetDeviceSerialNumbers . . . 1565
9.3.2.12 pack_dms_GetFirmwareInfo . . . 1565
9.3.2.13 pack_dms_GetFirmwareRevision . . . 1566
9.3.2.14 pack_dms_GetFirmwareRevisions . . . 1566
9.3.2.15 pack_dms_GetFSN . . . 1566
9.3.2.16 pack_dms_GetHardwareRevision . . . 1567

9.3.2.17 pack_dms_GetIMSI . . . 1567
9.3.2.18 pack_dms_GetManufacturer . . . 1567
9.3.2.19 pack_dms_GetModelID . . . 1568
9.3.2.20 pack_dms_GetNetworkTime . . . 1568
9.3.2.21 pack_dms_GetOfflineReason . . . 1568
9.3.2.22 pack_dms_GetPower . . . 1569
9.3.2.23 pack_dms_GetPRLVersion . . . 1569
9.3.2.24 pack_dms_GetSerialNumbers . . . 1569
9.3.2.25 pack_dms_GetUSBComp . . . 1570
9.3.2.26 pack_dms_GetVoiceNumber . . . 1570
9.3.2.27 pack_dms_ResetToFactoryDefaults . . . 1571
9.3.2.28 pack_dms_SetActivationStatusCallback . . . 1571
9.3.2.29 pack_dms_SetCrashAction . . . 1571
9.3.2.30 pack_dms_SetCustFeature . . . 1572
9.3.2.31 pack_dms_SetCustFeaturesV2 . . . 1572
9.3.2.32 pack_dms_SetEventReport . . . 1572
9.3.2.33 pack_dms_SetFirmwarePreference . . . 1573
9.3.2.34 pack_dms_SetIndicationRegister . . . 1573
9.3.2.35 pack_dms_SetPower . . . 1574
9.3.2.36 pack_dms_SetUSBComp . . . 1574
9.3.2.37 pack_dms_SLQSDmsSwiGetPCInfo . . . 1574
9.3.2.38 pack_dms_SLQSDmsSwiGetResetInfo . . . 1575
9.3.2.39 pack_dms_SLQSDmsSwiIndicationRegister . . . 1575
9.3.2.40 pack_dms_SLQSGetBandCapability . . . 1575
9.3.2.41 pack_dms_SLQSGetERIFile . . . 1576
9.3.2.42 pack_dms_SLQSGetPowerSaveModeConfig . . . 1576
9.3.2.43 pack_dms_SLQSSetPowerSaveModeConfig . . . 1577
9.3.2.44 pack_dms_SLQSSwiClearDyingGaspStatistics . . . 1577
9.3.2.45 pack_dms_SLQSSwiGetCrashInfo . . . 1577
9.3.2.46 pack_dms_SLQSSwiGetDyingGaspCfg . . . 1578
9.3.2.47 pack_dms_SLQSSwiGetDyingGaspStatistics . . . 1578
9.3.2.48 pack_dms_SLQSSwiGetFirmwareCurr . . . 1578
9.3.2.49 pack_dms_SLQSSwiGetFwUpdateStatus . . . 1579
9.3.2.50 pack_dms_SLQSSwiGetHostDevInfo . . . 1579
9.3.2.51 pack_dms_SLQSSwiGetOSInfo . . . 1579
9.3.2.52 pack_dms_SLQSSwiGetSerialNoExt . . . 1580
9.3.2.53 pack_dms_SLQSSwiSetDyingGaspCfg . . . 1580
9.3.2.54 pack_dms_SLQSSwiSetHostDevInfo . . . 1580
9.3.2.55 pack_dms_SLQSSwiSetOSInfo . . . 1581
9.3.2.56 pack_dms_SLQSUIMGetState . . . 1581

9.3.2.57 pack_dms_SwiSetEventReport . . . 1581
9.3.2.58 pack_dms_UIMChangePIN . . . 1582
9.3.2.59 pack_dms_UIMGetControlKeyStatus . . . 1582
9.3.2.60 pack_dms_UIMGetICCID . . . 1583
9.3.2.61 pack_dms_UIMGetPINStatus . . . 1583
9.3.2.62 pack_dms_UIMSetControlKeyProtection . . . 1583
9.3.2.63 pack_dms_UIMSetPINProtection . . . 1584
9.3.2.64 pack_dms_UIMUnblockControlKey . . . 1584
9.3.2.65 pack_dms_UIMUnblockPIN . . . 1584
9.3.2.66 pack_dms_UIMVerifyPIN . . . 1585
9.3.2.67 pack_dms_ValidateSPC . . . 1585
9.3.2.68 unpack_dms_ActivateAutomatic . . . 1586
9.3.2.69 unpack_dms_GetActivationState . . . 1586
9.3.2.70 unpack_dms_GetBandCapability . . . 1586
9.3.2.71 unpack_dms_GetCrashAction . . . 1587
9.3.2.72 unpack_dms_GetCustFeature . . . 1587
9.3.2.73 unpack_dms_GetCustFeaturesV2 . . . 1587
9.3.2.74 unpack_dms_GetDeviceCap . . . 1588
9.3.2.75 unpack_dms_GetDeviceCapabilities . . . 1588
9.3.2.76 unpack_dms_GetDeviceHardwareRev . . . 1588
9.3.2.77 unpack_dms_GetDeviceMfr . . . 1589
9.3.2.78 unpack_dms_GetDeviceSerialNumbers . . . 1589
9.3.2.79 unpack_dms_GetFirmwareInfo . . . 1589
9.3.2.80 unpack_dms_GetFirmwareRevision . . . 1590
9.3.2.81 unpack_dms_GetFirmwareRevisions . . . 1590
9.3.2.82 unpack_dms_GetFSN . . . 1590
9.3.2.83 unpack_dms_GetHardwareRevision . . . 1591
9.3.2.84 unpack_dms_GetIMSI . . . 1591
9.3.2.85 unpack_dms_GetManufacturer . . . 1592
9.3.2.86 unpack_dms_GetModelID . . . 1592
9.3.2.87 unpack_dms_GetNetworkTime . . . 1592
9.3.2.88 unpack_dms_GetOfflineReason . . . 1593
9.3.2.89 unpack_dms_GetPower . . . 1593
9.3.2.90 unpack_dms_GetPRLVersion . . . 1593
9.3.2.91 unpack_dms_GetSerialNumbers . . . 1594
9.3.2.92 unpack_dms_GetUSBComp . . . 1594
9.3.2.93 unpack_dms_GetVoiceNumber . . . 1594
9.3.2.94 unpack_dms_PSMCfgChange_ind . . . 1595
9.3.2.95 unpack_dms_ResetToFactoryDefaults . . . 1595
9.3.2.96 unpack_dms_SetActivationStatusCallback . . . 1595

9.3.2.97 unpack_dms_SetCrashAction . . . 1596
9.3.2.98 unpack_dms_SetCustFeature . . . 1596
9.3.2.99 unpack_dms_SetCustFeaturesV2 . . . 1596
9.3.2.100 unpack_dms_SetEventReport . . . 1597
9.3.2.101 unpack_dms_SetEventReport_ind . . . 1597
9.3.2.102 unpack_dms_SetFirmwarePreference . . . 1598
9.3.2.103 unpack_dms_SetIndicationRegister . . . 1598
9.3.2.104 unpack_dms_SetPower . . . 1598
9.3.2.105 unpack_dms_SetUSBComp . . . 1599
9.3.2.106 unpack_dms_SLQSDmsSwiGetPCInfo . . . 1599
9.3.2.107 unpack_dms_SLQSDmsSwiGetResetInfo . . . 1599
9.3.2.108 unpack_dms_SLQSDmsSwiGetResetInfo_Ind . . . 1600
9.3.2.109 unpack_dms_SLQSDmsSwiIndicationRegister . . . 1600
9.3.2.110 unpack_dms_SLQSGetBandCapability . . . 1600
9.3.2.111 unpack_dms_SLQSGetERIFile . . . 1601
9.3.2.112 unpack_dms_SLQSGetPowerSaveModeConfig . . . 1601
9.3.2.113 unpack_dms_SLQSSetPowerSaveModeConfig . . . 1602
9.3.2.114 unpack_dms_SLQSSwiClearDyingGaspStatistics . . . 1602
9.3.2.115 unpack_dms_SLQSSwiGetCrashInfo . . . 1602
9.3.2.116 unpack_dms_SLQSSwiGetDyingGaspCfg . . . 1603
9.3.2.117 unpack_dms_SLQSSwiGetDyingGaspStatistics . . . 1603
9.3.2.118 unpack_dms_SLQSSwiGetFirmwareCurr . . . 1603
9.3.2.119 unpack_dms_SLQSSwiGetFwUpdateStatus . . . 1604
9.3.2.120 unpack_dms_SLQSSwiGetHostDevInfo . . . 1604
9.3.2.121 unpack_dms_SLQSSwiGetOSInfo . . . 1604
9.3.2.122 unpack_dms_SLQSSwiGetSerialNoExt . . . 1605
9.3.2.123 unpack_dms_SLQSSwiSetDyingGaspCfg . . . 1605
9.3.2.124 unpack_dms_SLQSSwiSetHostDevInfo . . . 1605
9.3.2.125 unpack_dms_SLQSSwiSetOSInfo . . . 1606
9.3.2.126 unpack_dms_SLQSUIMGetState . . . 1606
9.3.2.127 unpack_dms_SwiEventReportCallBack_ind . . . 1606
9.3.2.128 unpack_dms_SwiSetEventReport . . . 1607
9.3.2.129 unpack_dms_UIMChangePIN . . . 1607
9.3.2.130 unpack_dms_UIMGetControlKeyStatus . . . 1608
9.3.2.131 unpack_dms_UIMGetICCID . . . 1608
9.3.2.132 unpack_dms_UIMGetPINStatus . . . 1608
9.3.2.133 unpack_dms_UIMSetControlKeyProtection . . . 1609
9.3.2.134 unpack_dms_UIMSetPINProtection . . . 1609
9.3.2.135 unpack_dms_UIMUnblockControlKey . . . 1609
9.3.2.136 unpack_dms_UIMUnblockPIN . . . 1610

9.3.2.137 unpack_dms_UIMVerifyPIN . . . 1610
9.3.2.138 unpack_dms_ValidateSPC . . . 1610
9.4 fms.h File Reference . . . 1611
9.4.1 Macro Definition Documentation . . . 1612
9.4.1.1 FMS_FW_PRI_BUILD_MATCH_LEN . . . 1612
9.4.1.2 FMS_GOBI_LISTENTRIES_MAX . . . 1612
9.4.1.3 FMS_GOBI_MBN_BUILD_ID_STR_LEN . . . 1612
9.4.1.4 FMS_GOBI_MBN_IMG_ID_STR_LEN . . . 1612
9.4.1.5 FMS_IMAGE_ID_BUILD_ID_LEN . . . 1612
9.4.1.6 FMS_IMAGE_ID_IMG_ID_LEN . . . 1612
9.4.1.7 FMS_IMAGE_ID_MAX_ENTRIES . . . 1612
9.4.1.8 FMS_IMAGE_ID_PRI_IMGTYPE . . . 1612
9.4.1.9 FMS_MAX_IMAGE_ID_ELEMENT . . . 1612
9.4.1.10 FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE . . . 1612
9.4.2 Function Documentation . . . 1612
9.4.2.1 GetValidFwPriCombinations . . . 1612
9.4.2.2 pack_fms_GetImagesPreference . . . 1612
9.4.2.3 pack_fms_GetStoredImages . . . 1613
9.4.2.4 pack_fms_SetImagesPreference . . . 1613
9.4.2.5 unpack_fms_GetImagesPreference . . . 1613
9.4.2.6 unpack_fms_GetStoredImages . . . 1613
9.4.2.7 unpack_fms_SetImagesPreference . . . 1614
9.5 loc.h File Reference . . . 1614
9.5.1 Macro Definition Documentation . . . 1617
9.5.1.1 LOC_UINT8_MAX_STRING_SZ . . . 1617
9.5.1.2 LOCEVENTMASKBATCHFULLNOTIFICATION . . . 1617
9.5.1.3 LOCEVENTMASKENGINESTATE . . . 1617
9.5.1.4 LOCEVENTMASKFIXSESSIONSTATE . . . 1617
9.5.1.5 LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION . . . 1617
9.5.1.6 LOCEVENTMASKGEOFENCEBREACHNOTIFICATION . . . 1617
9.5.1.7 LOCEVENTMASKGEOFENCEGENALERT . . . 1618
9.5.1.8 LOCEVENTMASKGNSSMEASUREMENTREPORT . . . 1618
9.5.1.9 LOCEVENTMASKGNSSSVINFO . . . 1618
9.5.1.10 LOCEVENTMASKINJECTPOSITIONREQ . . . 1618
9.5.1.11 LOCEVENTMASKINJECTPREDICTEDORBITSREQ . . . 1618
9.5.1.12 LOCEVENTMASKINJECTTIMEREQ . . . 1618
9.5.1.13 LOCEVENTMASKINJECTWIFIAPDATAREQ . . . 1618
9.5.1.14 LOCEVENTMASKINVALIDVALUE . . . 1618
9.5.1.15 LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT . . . 1618
9.5.1.16 LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ . . . 1618

9.5.1.17 LOCEVENTMASKMOTIONDATACONTROL . . . 1618
9.5.1.18 LOCEVENTMASKNIGEOFENCENOTIFICATION . . . 1619
9.5.1.19 LOCEVENTMASKNINOTIFYVERIFYREQ . . . 1619
9.5.1.20 LOCEVENTMASKNMEA . . . 1619
9.5.1.21 LOCEVENTMASKPEDOMETERCONTROL . . . 1619
9.5.1.22 LOCEVENTMASKPOSITIONREPORT . . . 1619
9.5.1.23 LOCEVENTMASKSENSORSTREAMINGREADYSTATUS . . . 1619
9.5.1.24 LOCEVENTMASKSETSPISTREAMINGREPORT . . . 1619
9.5.1.25 LOCEVENTMASKTIMESYNCREQ . . . 1619
9.5.1.26 LOCEVENTMASKVEHICLEDATAREADYSTATUS . . . 1619
9.5.1.27 LOCEVENTMASKWIFIREQ . . . 1619
9.5.1.28 MAX_LOC_NMEA_STR_LEN . . . 1619
9.5.1.29 MAX_SENSOR_DATA_LEN . . . 1619
9.5.1.30 MAX_TEMP_DATA_LEN . . . 1620
9.5.2 Enumeration Type Documentation . . . 1620
9.5.2.1 anonymous enum . . . 1620
9.5.3 Function Documentation . . . 1620
9.5.3.1 pack_loc_DeleteAssistData . . . 1620
9.5.3.2 pack_loc_EventRegister . . . 1620
9.5.3.3 pack_loc_SetExtPowerState . . . 1621
9.5.3.4 pack_loc_SetOperationMode . . . 1621
9.5.3.5 pack_loc_SLQSLOCGetBestAvailPos . . . 1621
9.5.3.6 pack_loc_SLQSLOCGetServer . . . 1622
9.5.3.7 pack_loc_SLQSLOCInjectPosition . . . 1622
9.5.3.8 pack_loc_SLQSLOCInjectSensorData . . . 1623
9.5.3.9 pack_loc_SLQSLOCInjectUTCTime . . . 1623
9.5.3.10 pack_loc_SLQSLOCSetCradleMountConfig . . . 1623
9.5.3.11 pack_loc_SLQSLOCSetServer . . . 1624
9.5.3.12 pack_loc_Start . . . 1624
9.5.3.13 pack_loc_Stop . . . 1625
9.5.3.14 unpack_loc_BestAvailPos_Ind . . . 1625
9.5.3.15 unpack_loc_CradleMountCallback_Ind . . . 1625
9.5.3.16 unpack_loc_DeleteAssistData . . . 1626
9.5.3.17 unpack_loc_DeleteAssistData_Ind . . . 1626
9.5.3.18 unpack_loc_EngineState_Ind . . . 1626
9.5.3.19 unpack_loc_EventNMEA_Ind . . . 1627
9.5.3.20 unpack_loc_EventRegister . . . 1627
9.5.3.21 unpack_loc_EventTimeSyncCallback_Ind . . . 1627
9.5.3.22 unpack_loc_GetServer_Ind . . . 1628
9.5.3.23 unpack_loc_GnssSvInfo_Ind . . . 1628

9.5.3.24 unpack_loc_InjectPositionCallback_Ind . . . 1628
9.5.3.25 unpack_loc_InjectSensorDataCallback_Ind . . . 1629
9.5.3.26 unpack_loc_InjectTimeSyncDataCallback_Ind . . . 1629
9.5.3.27 unpack_loc_InjectUTCTimeCallback_Ind . . . 1629
9.5.3.28 unpack_loc_PositionRpt_Ind . . . 1630
9.5.3.29 unpack_loc_SensorStreamingCallback_Ind . . . 1630
9.5.3.30 unpack_loc_SetExtPowerConfig_Ind . . . 1630
9.5.3.31 unpack_loc_SetExtPowerState . . . 1631
9.5.3.32 unpack_loc_SetOperationMode . . . 1631
9.5.3.33 unpack_loc_SetOperationMode_Ind . . . 1631
9.5.3.34 unpack_loc_SetServer_Ind . . . 1632
9.5.3.35 unpack_loc_SLQSLOCGetBestAvailPos . . . 1632
9.5.3.36 unpack_loc_SLQSLOCGetServer . . . 1632
9.5.3.37 unpack_loc_SLQSLOCInjectPosition . . . 1633
9.5.3.38 unpack_loc_SLQSLOCInjectSensorData . . . 1633
9.5.3.39 unpack_loc_SLQSLOCInjectUTCTime . . . 1633
9.5.3.40 unpack_loc_SLQSLOCSetCradleMountConfig . . . 1634
9.5.3.41 unpack_loc_SLQSLOCSetServer . . . 1634
9.5.3.42 unpack_loc_Start . . . 1634
9.5.3.43 unpack_loc_Stop . . . 1635
9.6 nas.h File Reference . . . 1635
9.6.1 Macro Definition Documentation . . . 1644
9.6.1.1 NAS_IMSI_M_S1_LENGTH . . . 1644
9.6.1.2 NAS_IMSI_M_S2_LENGTH . . . 1644
9.6.1.3 NAS_MAX_DESCRIPTION_LENGTH . . . 1644
9.6.1.4 NAS_MAX_NUM_NETWORKS . . . 1644
9.6.1.5 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1644
9.6.1.6 NAS_NAM_NAME_LENGTH . . . 1644
9.6.1.7 NAS_OTA_MESSAGE_MAX_BUF_SIZE . . . 1644
9.6.1.8 NAS_PLMN_LENGTH . . . 1644
9.6.1.9 NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST . . . 1644
9.6.2 Enumeration Type Documentation . . . 1644
9.6.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB . . . 1644
9.6.2.2 LITEQMI_NAS_LTE_CPHY_SCELL_STATE . . . 1644
9.6.2.3 NAS_LTE_CPHY_CA_BW_NRB_LITE . . . 1644
9.6.2.4 NAS_LTE_CPHY_SCELL_STATE_LITE . . . 1645
9.6.3 Function Documentation . . . 1645
9.6.3.1 pack_nas_GetACCOLC . . . 1645
9.6.3.2 pack_nas_GetANAAAAuthenticationStatus . . . 1645
9.6.3.3 pack_nas_GetCDMANetworkParameters . . . 1645

9.6.3.4 pack_nas_GetHomeNetwork . . . 1646
9.6.3.5 pack_nas_GetHomeNetwork3GPP2 . . . 1646
9.6.3.6 pack_nas_GetNetworkPreference . . . 1646
9.6.3.7 pack_nas_GetRFInfo . . . 1647
9.6.3.8 pack_nas_GetServingNetwork . . . 1647
9.6.3.9 pack_nas_GetServingNetworkCapabilities . . . 1647
9.6.3.10 pack_nas_GetSignalStrengths . . . 1648
9.6.3.11 pack_nas_InitiateDomainAttach . . . 1648
9.6.3.12 pack_nas_PerformNetworkScan . . . 1648
9.6.3.13 pack_nas_SetACCOLC . . . 1649
9.6.3.14 pack_nas_SetCDMANetworkParameters . . . 1649
9.6.3.15 pack_nas_SetLURejectCallback . . . 1649
9.6.3.16 pack_nas_SetNetworkPreference . . . 1650
9.6.3.17 pack_nas_SetRFInfoCallback . . . 1650
9.6.3.18 pack_nas_SLQSConfigSigInfo . . . 1650
9.6.3.19 pack_nas_SLQSGetErrorRate . . . 1651
9.6.3.20 pack_nas_SlqsGetLTECphyCAInfo . . . 1651
9.6.3.21 pack_nas_SLQSGetNetworkTime . . . 1651
9.6.3.22 pack_nas_SLQSGetOperatorNameData . . . 1652
9.6.3.23 pack_nas_SLQSGetPLMNName . . . 1652
9.6.3.24 pack_nas_SLQSGetServingSystem . . . 1652
9.6.3.25 pack_nas_SLQSGetSignalStrength . . . 1653
9.6.3.26 pack_nas_SLQSGetSysInfo . . . 1653
9.6.3.27 pack_nas_SLQSGetSysSelectionPref . . . 1653
9.6.3.28 pack_nas_SLQSInitiateNetworkRegistration . . . 1654
9.6.3.29 pack_nas_SLQSNasConfigSigInfo2 . . . 1654
9.6.3.30 pack_nas_SLQSNasGet3GPP2Subscription . . . 1654
9.6.3.31 pack_nas_SLQSNasGetCellLocationInfo . . . 1655
9.6.3.32 pack_nas_SLQSNASGeteDRXParams . . . 1655
9.6.3.33 pack_nas_SLQSNasGetHDRColorCode . . . 1655
9.6.3.34 pack_nas_SLQSNasGetRFInfo . . . 1656
9.6.3.35 pack_nas_SLQSNasGetSigInfo . . . 1656
9.6.3.36 pack_nas_SLQSNasGetTxRxInfo . . . 1656
9.6.3.37 pack_nas_SLQSNasIndicationRegisterExt . . . 1657
9.6.3.38 pack_nas_SLQSNASSeteDRXParams . . . 1657
9.6.3.39 pack_nas_SLQSNASSwiGetChannelLock . . . 1658
9.6.3.40 pack_nas_SLQSNasSwiIndicationRegister . . . 1658
9.6.3.41 pack_nas_SLQSNasSwiModemStatus . . . 1658
9.6.3.42 pack_nas_SLQSNASSwiSetChannelLock . . . 1659
9.6.3.43 pack_nas_SLQSSetBandPreference . . . 1659

9.6.3.44 pack_nas_SLQSSetSignalStrengthsCallback . . . 1659
9.6.3.45 pack_nas_SLQSSetSysSelectionPref . . . 1660
9.6.3.46 pack_nas_SLQSSetSysSelectionPrefExt . . . 1660
9.6.3.47 pack_nas_SLQSSwiGetHDRPersonality . . . 1660
9.6.3.48 pack_nas_SLQSSwiGetHDRProtSubtype . . . 1661
9.6.3.49 pack_nas_SLQSSwiGetHRPDStats . . . 1661
9.6.3.50 pack_nas_SLQSSwiGetLteCQI . . . 1661
9.6.3.51 pack_nas_SLQSSwiGetLteSccRxInfo . . . 1662
9.6.3.52 pack_nas_SLQSSwiNetworkDebug . . . 1662
9.6.3.53 pack_nas_SLQSSwiPSDetach . . . 1662
9.6.3.54 unpack_nas_GetACCOLC . . . 1663
9.6.3.55 unpack_nas_GetANAAAAuthenticationStatus . . . 1663
9.6.3.56 unpack_nas_GetCDMANetworkParameters . . . 1664
9.6.3.57 unpack_nas_GetHomeNetwork . . . 1664
9.6.3.58 unpack_nas_GetHomeNetwork3GPP2 . . . 1664
9.6.3.59 unpack_nas_GetNetworkPreference . . . 1665
9.6.3.60 unpack_nas_GetRFInfo . . . 1665
9.6.3.61 unpack_nas_GetServingNetwork . . . 1665
9.6.3.62 unpack_nas_GetServingNetworkCapabilities . . . 1666
9.6.3.63 unpack_nas_GetSignalStrengths . . . 1666
9.6.3.64 unpack_nas_InitiateDomainAttach . . . 1667
9.6.3.65 unpack_nas_PerformNetworkScan . . . 1667
9.6.3.66 unpack_nas_SetACCOLC . . . 1667
9.6.3.67 unpack_nas_SetCDMANetworkParameters . . . 1668
9.6.3.68 unpack_nas_SetDataCapabilitiesCallback_ind . . . 1668
9.6.3.69 unpack_nas_SetEventReportInd . . . 1668
9.6.3.70 unpack_nas_SetLURejectCallback . . . 1669
9.6.3.71 unpack_nas_SetNasLTECphyCaIndCallback_ind . . . 1669
9.6.3.72 unpack_nas_SetNetworkPreference . . . 1669
9.6.3.73 unpack_nas_SetRFInfoCallback . . . 1670
9.6.3.74 unpack_nas_SetRoamingIndicatorCallback_ind . . . 1670
9.6.3.75 unpack_nas_SetServingSystemCallback_ind . . . 1670
9.6.3.76 unpack_nas_SLQSConfigSigInfo . . . 1671
9.6.3.77 unpack_nas_SLQSGetErrorRate . . . 1671
9.6.3.78 unpack_nas_SlqsGetLTECphyCAInfo . . . 1671
9.6.3.79 unpack_nas_SLQSGetNetworkTime . . . 1672
9.6.3.80 unpack_nas_SLQSGetOperatorNameData . . . 1672
9.6.3.81 unpack_nas_SLQSGetPLMNName . . . 1672
9.6.3.82 unpack_nas_SLQSGetServingSystem . . . 1673
9.6.3.83 unpack_nas_SLQSGetSignalStrength . . . 1673

9.6.3.84 unpack_nas_SLQSGetSysInfo . . . 1673
9.6.3.85 unpack_nas_SLQSGetSysSelectionPref . . . 1674
9.6.3.86 unpack_nas_SLQSInitiateNetworkRegistration . . . 1674
9.6.3.87 unpack_nas_SLQSNasConfigSigInfo2 . . . 1674
9.6.3.88 unpack_nas_SLQSNasGet3GPP2Subscription . . . 1675
9.6.3.89 unpack_nas_SLQSNasGetCellLocationInfo . . . 1675
9.6.3.90 unpack_nas_SLQSNASGeteDRXParams . . . 1675
9.6.3.91 unpack_nas_SLQSNasGetHDRColorCode . . . 1676
9.6.3.92 unpack_nas_SLQSNasGetRFInfo . . . 1676
9.6.3.93 unpack_nas_SLQSNasGetSigInfo . . . 1677
9.6.3.94 unpack_nas_SLQSNasGetTxRxInfo . . . 1677
9.6.3.95 unpack_nas_SLQSNasIndicationRegisterExt . . . 1677
9.6.3.96 unpack_nas_SLQSNasNetworkTimeCallBack_ind . . . 1678
9.6.3.97 unpack_nas_SLQSNASSeteDRXParams . . . 1678
9.6.3.98 unpack_nas_SLQSNasSigInfoCallback_ind . . . 1678
9.6.3.99 unpack_nas_SLQSNASSwiGetChannelLock . . . 1679
9.6.3.100 unpack_nas_SLQSNasSwiIndicationRegister . . . 1679
9.6.3.101 unpack_nas_SLQSNasSwiModemStatus . . . 1679
9.6.3.102 unpack_nas_SLQSNasSwiOTAMessageCallback_ind . . . 1680
9.6.3.103 unpack_nas_SLQSNASSwiSetChannelLock . . . 1680
9.6.3.104 unpack_nas_SLQSNasSysInfoCallback_ind . . . 1680
9.6.3.105 unpack_nas_SLQSNasTimerCallback_ind . . . 1681
9.6.3.106 unpack_nas_SLQSSetBandPreference . . . 1681
9.6.3.107 unpack_nas_SLQSSetSignalStrengthsCallback . . . 1681
9.6.3.108 unpack_nas_SLQSSetSysSelectionPref . . . 1682
9.6.3.109 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind . . . 1682
9.6.3.110 unpack_nas_SLQSSetSysSelectionPrefExt . . . 1682
9.6.3.111 unpack_nas_SLQSSwiGetHDRPersonality . . . 1683
9.6.3.112 unpack_nas_SLQSSwiGetHDRProtSubtype . . . 1683
9.6.3.113 unpack_nas_SLQSSwiGetHRPDStats . . . 1683
9.6.3.114 unpack_nas_SLQSSwiGetLteCQI . . . 1684
9.6.3.115 unpack_nas_SLQSSwiGetLteSccRxInfo . . . 1684
9.6.3.116 unpack_nas_SLQSSwiHDRPersonalityCallback_Ind . . . 1684
9.6.3.117 unpack_nas_SLQSSwiNetworkDebug . . . 1685
9.6.3.118 unpack_nas_SLQSSwiPSDetach . . . 1685
9.6.3.119 unpack_nas_SLQSSwiRandIndicatorCallback_Ind . . . 1685
9.6.3.120 unpack_valid_nas_GetCDMANetworkParameters . . . 1686
9.6.3.121 unpack_valid_nas_SLQSGetServingSystem . . . 1686
9.6.3.122 unpack_valid_nas_SLQSGetSignalStrength . . . 1687
9.6.3.123 unpack_valid_nas_SLQSNasGetSigInfo . . . 1688

9.7 pds.h File Reference . . . 1688
9.7.1 Function Documentation . . . 1690
9.7.1.1 pack_pds_ForceXTRADownload . . . 1690
9.7.1.2 pack_pds_GetPDSDefaults . . . 1691
9.7.1.3 pack_pds_GetPDSState . . . 1691
9.7.1.4 pack_pds_GetPortAutomaticTracking . . . 1691
9.7.1.5 pack_pds_GetServiceAutomaticTracking . . . 1692
9.7.1.6 pack_pds_GetXTRAAutomaticDownload . . . 1692
9.7.1.7 pack_pds_GetXTRANetwork . . . 1692
9.7.1.8 pack_pds_GetXTRAValidity . . . 1693
9.7.1.9 pack_pds_PDSInjectTimeReference . . . 1693
9.7.1.10 pack_pds_ResetPDSData . . . 1693
9.7.1.11 pack_pds_SetEventReportCallback . . . 1694
9.7.1.12 pack_pds_SetPDSDefaults . . . 1694
9.7.1.13 pack_pds_SetPDSState . . . 1694
9.7.1.14 pack_pds_SetPortAutomaticTracking . . . 1695
9.7.1.15 pack_pds_SetServiceAutomaticTracking . . . 1695
9.7.1.16 pack_pds_SetXTRAAutomaticDownload . . . 1695
9.7.1.17 pack_pds_SetXTRANetwork . . . 1696
9.7.1.18 pack_pds_SLQSGetAGPSConfig . . . 1696
9.7.1.19 pack_pds_SLQSGetGPSStateInfo . . . 1697
9.7.1.20 pack_pds_SLQSPDSDeterminePosition . . . 1697
9.7.1.21 pack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1697
9.7.1.22 pack_pds_SLQSPDSInjectPositionData . . . 1698
9.7.1.23 pack_pds_SLQSSetAGPSConfig . . . 1698
9.7.1.24 pack_pds_SLQSSetPositionMethodState . . . 1698
9.7.1.25 pack_pds_StartPDSTrackingSessionExt . . . 1699
9.7.1.26 pack_pds_StopPDSTrackingSession . . . 1699
9.7.1.27 unpack_pds_ForceXTRADownload . . . 1699
9.7.1.28 unpack_pds_GetPDSDefaults . . . 1700
9.7.1.29 unpack_pds_GetPDSState . . . 1700
9.7.1.30 unpack_pds_GetPortAutomaticTracking . . . 1700
9.7.1.31 unpack_pds_GetServiceAutomaticTracking . . . 1701
9.7.1.32 unpack_pds_GetXTRAAutomaticDownload . . . 1701
9.7.1.33 unpack_pds_GetXTRANetwork . . . 1701
9.7.1.34 unpack_pds_GetXTRAValidity . . . 1702
9.7.1.35 unpack_pds_PDSInjectTimeReference . . . 1702
9.7.1.36 unpack_pds_ResetPDSData . . . 1702
9.7.1.37 unpack_pds_SetEventReport_Ind . . . 1703
9.7.1.38 unpack_pds_SetEventReportCallback . . . 1703

9.7.1.39 unpack_pds_SetPDSDefaults . . . 1703
9.7.1.40 unpack_pds_SetPDSState . . . 1704
9.7.1.41 unpack_pds_SetPdsState_Ind . . . 1704
9.7.1.42 unpack_pds_SetPortAutomaticTracking . . . 1704
9.7.1.43 unpack_pds_SetServiceAutomaticTracking . . . 1705
9.7.1.44 unpack_pds_SetXTRAAutomaticDownload . . . 1705
9.7.1.45 unpack_pds_SetXTRANetwork . . . 1705
9.7.1.46 unpack_pds_SLQSGetAGPSConfig . . . 1706
9.7.1.47 unpack_pds_SLQSGetGPSStateInfo . . . 1706
9.7.1.48 unpack_pds_SLQSPDSDeterminePosition . . . 1706
9.7.1.49 unpack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1707
9.7.1.50 unpack_pds_SLQSPDSInjectPositionData . . . 1707
9.7.1.51 unpack_pds_SLQSSetAGPSConfig . . . 1707
9.7.1.52 unpack_pds_SLQSSetPositionMethodState . . . 1708
9.7.1.53 unpack_pds_StartPDSTrackingSessionExt . . . 1708
9.7.1.54 unpack_pds_StopPDSTrackingSession . . . 1708
9.8 qaCbkCatEventReportInd.h File Reference . . . 1709
9.8.1 Macro Definition Documentation . . . 1710
9.8.1.1 QMI_CAN_COMMON_EVENT_TLV_NUMBER . . . 1710
9.8.1.2 QMI_MAX_CAT_EVENT_DATA_LENGTH . . . 1710
9.8.2 Enumeration Type Documentation . . . 1710
9.8.2.1 eQMI_CAT_EVENT_REPORT_IND_TLV . . . 1710
9.8.2.2 eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH . . . 1710
9.8.3 Function Documentation . . . 1710
9.8.3.1 UpkQmiCbkCatEventReportInd . . . 1710
9.9 qaCbkSwiOmaDmEventReportInd.h File Reference . . . 1710
9.9.1 Macro Definition Documentation . . . 1711
9.9.1.1 QMI_SWIOMA_DM_CONFIG . . . 1711
9.9.1.2 QMI_SWIOMA_DM_FOTA . . . 1711
9.9.1.3 QMI_SWIOMA_DM_NOT . . . 1711
9.9.2 Enumeration Type Documentation . . . 1711
9.9.2.1 eQMI_SWIOMA_DM_EVENT_REPORT_IND . . . 1711
9.9.3 Function Documentation . . . 1711
9.9.3.1 UpkQmiCbkSwiOmaDmEventReportInd . . . 1711
9.9.3.2 UpkQmiCbkSwiOmaDmEventReportIndExt . . . 1711
9.10 qaGobiApiAudio.h File Reference . . . 1711
9.10.1 Detailed Description . . . 1712
9.10.2 Function Documentation . . . 1712
9.10.2.1 SLQSGetAudioPathConfig . . . 1712
9.10.2.2 SLQSGetAudioProfile . . . 1713

9.10.2.3 SLQSGetAudioVolTLBConfig . . . 1713
9.10.2.4 SLQSSetAudioPathConfig . . . 1714
9.10.2.5 SLQSSetAudioProfile . . . 1714
9.10.2.6 SLQSSetAudioVolTLBConfig . . . 1714
9.11 qaGobiApiCat.h File Reference . . . 1715
9.11.1 Detailed Description . . . 1715
9.11.2 Function Documentation . . . 1715
9.11.2.1 CATSendEnvelopeCommand . . . 1715
9.11.2.2 CATSendTerminalResponse . . . 1716
9.12 qaGobiApiCbk.h File Reference . . . 1716
9.12.1 Detailed Description . . . 1725
9.12.2 Macro Definition Documentation . . . 1725
9.12.2.1 CBK_DISABLE_EVENT . . . 1725
9.12.2.2 CBK_ENABLE_EVENT . . . 1725
9.12.2.3 CBK_NOCHANGE . . . 1725
9.12.2.4 DEREGISTER_EVENT . . . 1725
9.12.2.5 DEREGISTER_SRV . . . 1725
9.12.2.6 DHCP_MAX_NUM_OPTIONS . . . 1725
9.12.2.7 DHCP_OPTION_DATA_BUF_SIZE . . . 1725
9.12.2.8 EVENT_MASK_CARD . . . 1725
9.12.2.9 EVENT_MASK_DEREGISTER_ALL . . . 1725
9.12.2.10 EVENT_MASK_PHY_SLOT_STATUS . . . 1725
9.12.2.11 FIRST_INSTANCE . . . 1725
9.12.2.12 INVALID_INSTACNE . . . 1725
9.12.2.13 IPV4 . . . 1725
9.12.2.14 IPV4V6 . . . 1725
9.12.2.15 IPV6 . . . 1725
9.12.2.16 LOC_EVENT_MASK_ENG_STATE . . . 1725
9.12.2.17 LOC_EVENT_MASK_GNSS_SV_INFO . . . 1725
9.12.2.18 LOC_EVENT_MASK_INJECT_TIME . . . 1725
9.12.2.19 LOC_EVENT_MASK_NMEA . . . 1725
9.12.2.20 LOC_EVENT_MASK_SENSOR_STREAM . . . 1726
9.12.2.21 LOC_EVENT_MASK_TIME_SYNC . . . 1726
9.12.2.22 LOC_EVENT_POSITION_REPORT . . . 1726
9.12.2.23 MAX_MITIGATION_DEV_ID_LEN . . . 1726
9.12.2.24 MAX_NMEA_STR_LEN . . . 1726
9.12.2.25 MAX_NO_OF_APPLICATIONS . . . 1726
9.12.2.26 MAX_NO_OF_CALLS . . . 1726
9.12.2.27 MAX_NO_OF_FILES . . . 1726
9.12.2.28 MAX_NO_OF_SLOTS . . . 1726

9.12.2.29 MAX_NO_OF_UUSINFO . . . 1726
9.12.2.30 MAX_PATH_LENGTH . . . 1726
9.12.2.31 MAX_RADIO_INTERFACE_LIST . . . 1726
9.12.2.32 MAX_URL_ADDR_STR_LEN . . . 1726
9.12.2.33 MAXUSSDLENGTH . . . 1726
9.12.2.34 NAS_SRV . . . 1726
9.12.2.35 NUM_OF_SET . . . 1726
9.12.2.36 PDS_SRV . . . 1726
9.12.2.37 QMI_ETWS_MAX_PAYLOAD_LENGTH . . . 1726
9.12.2.38 QMI_MAX_VOICE_NUMBER_LENGTH . . . 1726
9.12.2.39 QMI_WMS_MAX_PAYLOAD_LENGTH . . . 1726
9.12.2.40 REGISTER_EVENT . . . 1726
9.12.2.41 REGISTER_SRV . . . 1726
9.12.2.42 SECOND_INSTANCE . . . 1726
9.12.2.43 SIGSTRENGTH_THRESHOLD_ARR_SZ . . . 1726
9.12.2.44 THIRD_INSTANCE . . . 1726
9.12.2.45 USSD_DCS_8BIT . . . 1726
9.12.2.46 USSD_DCS_ASCII . . . 1726
9.12.2.47 USSD_DCS_UCS2 . . . 1726
9.12.2.48 VOICE_SRV . . . 1727
9.12.2.49 WDS_SRV . . . 1727
9.12.3 Typedef Documentation . . . 1727
9.12.3.1 accelAcceptReady . . . 1727
9.12.3.2 accelTempAcceptReady . . . 1727
9.12.3.3 eDevState . . . 1728
9.12.3.4 eSMSEventType . . . 1728
9.12.3.5 gpsTime . . . 1728
9.12.3.6 gyroAcceptReady . . . 1728
9.12.3.7 gyroTempAcceptReady . . . 1729
9.12.3.8 LteNasReleaseInfo . . . 1729
9.12.3.9 modemTempNotification . . . 1729
9.12.3.10 packetSrvStatus . . . 1730
9.12.3.11 precisionDilution . . . 1731
9.12.3.12 ResetInfoNotification . . . 1731
9.12.3.13 sensorDataUsage . . . 1732
9.12.3.14 sessionInformation . . . 1733
9.12.3.15 sessionInformationExt . . . 1733
9.12.3.16 SMSAsyncRawSend . . . 1733
9.12.3.17 SMSCAddressInfo . . . 1734
9.12.3.18 SMSEtwsMessageInfo . . . 1734

9.12.3.19 SMSEtwsPlmnInfo . . . 1734
9.12.3.20 SMSEventInfo . . . 1735
9.12.3.21 SMSMessageModeInfo . . . 1735
9.12.3.22 SMSMTMessageInfo . . . 1735
9.12.3.23 SMSOnIMSInfo . . . 1736
9.12.3.24 SMSTransferRouteMTMessageInfo . . . 1736
9.12.3.25 svUsedforFix . . . 1736
9.12.3.26 SwiAvmsEventInd . . . 1737
9.12.3.27 SwiOTAMsg . . . 1737
9.12.3.28 tFNActivationStatus . . . 1737
9.12.3.29 tFNAllCallStatus . . . 1738
9.12.3.30 tFNASwiLTECphyCaInfo . . . 1738
9.12.3.31 tFNASwiOTAMsg . . . 1738
9.12.3.32 tFNAsyncRawSend . . . 1738
9.12.3.33 tFNBandPreference . . . 1739
9.12.3.34 tFNBestAvailPos . . . 1741
9.12.3.35 tFNCATEvent . . . 1741
9.12.3.36 tFNCbkUimSlotStatusChangeInd . . . 1741
9.12.3.37 tFNDataCapabilities . . . 1742
9.12.3.38 tFNDataSysStatus . . . 1742
9.12.3.39 tFNDelAssistData . . . 1742
9.12.3.40 tFNDeviceStateChange . . . 1742
9.12.3.41 tFNDHCPv4ClientLeaseStatus . . . 1743
9.12.3.42 tFNDTMFEvent . . . 1743
9.12.3.43 tFNDUNCallInfo . . . 1743
9.12.3.44 tFNEventNmea . . . 1743
9.12.3.45 tFNEventPosition . . . 1744
9.12.3.46 tFNFwDldCompletion . . . 1744
9.12.3.47 tFNGetServer . . . 1744
9.12.3.48 tFNGnssSvInfo . . . 1744
9.12.3.49 tFNHDRPersonaity . . . 1745
9.12.3.50 tFNImsaPdpStatus . . . 1745
9.12.3.51 tFNImsaRatStatus . . . 1745
9.12.3.52 tFNImsaRegStatus . . . 1745
9.12.3.53 tFNImsaSvcStatus . . . 1745
9.12.3.54 tFNImsRegMgrConfig . . . 1746
9.12.3.55 tFNImsSIPConfig . . . 1746
9.12.3.56 tFNImsSMSConfig . . . 1746
9.12.3.57 tFNImsUserConfig . . . 1746
9.12.3.58 tFNImsVoIPConfig . . . 1746

9.12.3.59 tFNInfoRec . . . 1747
9.12.3.60 tFNInjectPosition . . . 1747
9.12.3.61 tFNInjectSensorData . . . 1747
9.12.3.62 tFNInjectTimeStatus . . . 1747
9.12.3.63 tFNInjectUTCTime . . . 1747
9.12.3.64 tFNLUReject . . . 1748
9.12.3.65 tFNMemoryFull . . . 1750
9.12.3.66 tFNMessageWaiting . . . 1750
9.12.3.67 tFNMitiLvlRpt . . . 1750
9.12.3.68 tFNMobileIPStatus . . . 1750
9.12.3.69 tFNModemTempInfo . . . 1750
9.12.3.70 tFNNasTimer . . . 1751
9.12.3.71 tFNNet . . . 1751
9.12.3.72 tFNNetworkReject . . . 1751
9.12.3.73 tFNNetworkTime . . . 1751
9.12.3.74 tFNNewGPS . . . 1752
9.12.3.75 tFNNewNMEA . . . 1752
9.12.3.76 tFNNewRMTransferStatistics . . . 1753
9.12.3.77 tFNNewSMS . . . 1753
9.12.3.78 tFNOMADMState . . . 1753
9.12.3.79 tFNOpMode . . . 1754
9.12.3.80 tFNOTASPStatus . . . 1755
9.12.3.81 tFNPacketSrvState . . . 1755
9.12.3.82 tFNPDSState . . . 1755
9.12.3.83 tFNPower . . . 1756
9.12.3.84 tFNPrivacyChange . . . 1756
9.12.3.85 tFNPSMCfgParamChange . . . 1756
9.12.3.86 tFNQosNWStatus . . . 1757
9.12.3.87 tFNQosPriEvent . . . 1757
9.12.3.88 tFNQosStatus . . . 1757
9.12.3.89 tFNRankIndicator . . . 1758
9.12.3.90 tFNResetInfo . . . 1758
9.12.3.91 tFNRFInfo . . . 1758
9.12.3.92 tFNRoamingIndicator . . . 1759
9.12.3.93 tFNSDKTerminated . . . 1759
9.12.3.94 tFNSensorStreaming . . . 1759
9.12.3.95 tFNServingSystem . . . 1760
9.12.3.96 tFNSetCradleMount . . . 1760
9.12.3.97 tFNSetEngineState . . . 1760
9.12.3.98 tFNSetEventTimeSync . . . 1760

9.12.3.99 tFNSetExtPowerConfig . . . 1760
9.12.3.100 tFNSetServer . . . 1761
9.12.3.101 tFNSigInfo . . . 1761
9.12.3.102 tFNSignalStrength . . . 1761
9.12.3.103 tFNSLQSOMADMAlert . . . 1761
9.12.3.104 tFNSLQSQOSEvent . . . 1762
9.12.3.105 tFNSLQSSessionState . . . 1762
9.12.3.106 tFNSLQSSignalStrengths . . . 1762
9.12.3.107 tFNSLQSWDSEvent . . . 1762
9.12.3.108 tFNSMSEvents . . . 1762
9.12.3.109 tFNSUPSInfo . . . 1763
9.12.3.110 tFNSUPSNotification . . . 1763
9.12.3.111 tFNSwiAvmEventReport . . . 1763
9.12.3.112 tFNSysInfo . . . 1763
9.12.3.113 tFNSysSelectionPref . . . 1763
9.12.3.114 tFNtransLayerInfo . . . 1764
9.12.3.115 tFNtransNWRegInfo . . . 1764
9.12.3.116 tFNUIMRefresh . . . 1764
9.12.3.117 tFNUIMStatusChangeInfo . . . 1764
9.12.3.118 tFNUSSDNotification . . . 1764
9.12.3.119 tFNUSSDNoWaitIndication . . . 1765
9.12.3.120 tFNUSSDRelease . . . 1765
9.12.3.121 ttransLayerNotification . . . 1765
9.12.3.122 ttransNWRegInfoNotification . . . 1765
9.12.4 Enumeration Type Documentation . . . 1766
9.12.4.1 device_state_enum . . . 1766
9.12.4.2 eQaQMIService . . . 1766
9.12.4.3 SMSEventType . . . 1766
9.12.4.4 TYPE_OF_SERVICE_DOMAIN . . . 1767
9.12.5 Function Documentation . . . 1767
9.12.5.1 iSetCATEventCallback . . . 1767
9.12.5.2 iSetSignalStrengthCallback . . . 1767
9.12.5.3 iSLQSSetDUNCallInfoCallback . . . 1767
9.12.5.4 iSLQSSetSignalStrengthsCallback . . . 1767
9.12.5.5 iSLQSSetWdsFirstInstEventCallback . . . 1767
9.12.5.6 iSLQSSetWdsSecondInstEventCallback . . . 1767
9.12.5.7 iSLQSSetWdsThirdInstEventCallback . . . 1767
9.12.5.8 iSLQSSetWdsXferStatsFirstInstCallback . . . 1767
9.12.5.9 iSLQSSetWdsXferStatsSecondInstCallback . . . 1767
9.12.5.10 SetActivationStatusCallback . . . 1767

9.12.5.11 SetCATEventCallback . . . 1768
9.12.5.12 SetCfgParamChangeCallback . . . 1769
9.12.5.13 SetDataCapabilitiesCallback . . . 1769
9.12.5.14 SetDeviceStateChangeCbk . . . 1769
9.12.5.15 SetFwDldCompletionCbk . . . 1770
9.12.5.16 SetGPSCallback . . . 1770
9.12.5.17 SetLocBestAvailPosCallback . . . 1770
9.12.5.18 SetLocCradleMountCallback . . . 1771
9.12.5.19 SetLocDeleteAssistDataCallback . . . 1771
9.12.5.20 SetLocEngineStateCallback . . . 1771
9.12.5.21 SetLocEventMaskNMEACallback . . . 1771
9.12.5.22 SetLocEventPositionCallback . . . 1772
9.12.5.23 SetLocEventTimeSyncCallback . . . 1772
9.12.5.24 SetLocGetServerCallback . . . 1772
9.12.5.25 SetLocGnssSvInfoCallback . . . 1773
9.12.5.26 SetLocInjectSensorDataCallback . . . 1773
9.12.5.27 SetLocInjectTimeCallback . . . 1773
9.12.5.28 SetLocOpModeCallback . . . 1774
9.12.5.29 SetLocSensorStreamingCallback . . . 1774
9.12.5.30 SetLocSetExtPowerConfigCallback . . . 1774
9.12.5.31 SetLocSetServerCallback . . . 1774
9.12.5.32 SetLURejectCallback . . . 1775
9.12.5.33 SetMobileIPStatusCallback . . . 1775
9.12.5.34 SetNasLTECphyCaIndCallback . . . 1775
9.12.5.35 SetNetChangeCbk . . . 1776
9.12.5.36 SetNewSMSCallback . . . 1777
9.12.5.37 SetNMEACallback . . . 1777
9.12.5.38 SetOMADMStateCallback . . . 1777
9.12.5.39 SetPDSStateCallback . . . 1778
9.12.5.40 SetPowerCallback . . . 1778
9.12.5.41 SetRankIndicatorCallback . . . 1778
9.12.5.42 SetRFInfoCallback . . . 1778
9.12.5.43 SetRMTransferStatisticsCallback . . . 1779
9.12.5.44 SetRoamingIndicatorCallback . . . 1779
9.12.5.45 SetSignalStrengthCallback . . . 1780
9.12.5.46 SetSLQSOMADMAlertCallback . . . 1780
9.12.5.47 SetSLQSOMADMAlertCallbackExt . . . 1781
9.12.5.48 SetUimSlotStatusChangeCallback . . . 1781
9.12.5.49 SetUSSDNotificationCallback . . . 1781
9.12.5.50 SetUSSDNoWaitIndicationCallback . . . 1782

9.12.5.51 SetUSSDReleaseCallback . . . 1782
9.12.5.52 SLQSNasNetworkRejectCallback . . . 1783
9.12.5.53 SLQSNasNetworkTimeCallBack . . . 1783
9.12.5.54 SLQSNasSigInfo2CallBack . . . 1783
9.12.5.55 SLQSNasSigInfoCallBack . . . 1784
9.12.5.56 SLQSNasSwiOTAMessageCallback . . . 1784
9.12.5.57 SLQSNasSysInfoCallBack . . . 1785
9.12.5.58 SLQSNasTimerCallback . . . 1785
9.12.5.59 SLQSSetBandPreferenceCbk . . . 1785
9.12.5.60 SLQSSetDataSystemStatusCallback . . . 1786
9.12.5.61 SLQSSetDHCPv4ClientLeaseStatusCallback . . . 1786
9.12.5.62 SLQSSetDUNCallInfoCallback . . . 1786
9.12.5.63 SLQSSetIMSAPdpStatusCallback . . . 1787
9.12.5.64 SLQSSetIMSARatStatusCallback . . . 1787
9.12.5.65 SLQSSetIMSARegStatusCallback . . . 1788
9.12.5.66 SLQSSetIMSASvcStatusCallback . . . 1788
9.12.5.67 SLQSSetIMSSMSConfigCallback . . . 1789
9.12.5.68 SLQSSetIMSUserConfigCallback . . . 1789
9.12.5.69 SLQSSetIMSVoIPConfigCallback . . . 1789
9.12.5.70 SLQSSetLocInjectPositionCallback . . . 1790
9.12.5.71 SLQSSetLocInjectUTCTimeCallback . . . 1790
9.12.5.72 SLQSSetModemTempCallback . . . 1790
9.12.5.73 SLQSSetPacketSrvStatusCallback . . . 1791
9.12.5.74 SLQSSetQosEventCallback . . . 1791
9.12.5.75 SLQSSetQosNWStatusCallback . . . 1792
9.12.5.76 SLQSSetQosPriEventCallback . . . 1792
9.12.5.77 SLQSSetQosStatusCallback . . . 1792
9.12.5.78 SLQSSetRegMgrConfigCallback . . . 1793
9.12.5.79 SLQSSetSDKTerminatedCallback . . . 1793
9.12.5.80 SLQSSetServingSystemCallback . . . 1794
9.12.5.81 SLQSSetSessionStateCallback . . . 1794
9.12.5.82 SLQSSetSignalStrengthsCallback . . . 1795
9.12.5.83 SLQSSetSIPConfigCallback . . . 1795
9.12.5.84 SLQSSetSMSEventCallback . . . 1796
9.12.5.85 SLQSSetSwiGetResetInfoCallback . . . 1796
9.12.5.86 SLQSSetSwiHDRPersCallback . . . 1796
9.12.5.87 SLQSSetSysSelectionPrefCallBack . . . 1796
9.12.5.88 SLQSSetTransLayerInfoCallback . . . 1797
9.12.5.89 SLQSSetTransNWRegInfoCallback . . . 1797
9.12.5.90 SLQSSetWdsEventCallback . . . 1798

9.12.5.91 SLQSSetWdsTransferStatisticCallback . . . 1798
9.12.5.92 SLQSSwiAVMSEventReportCallback . . . 1799
9.12.5.93 SLQSTmdMitigationLvlRptCallback . . . 1799
9.12.5.94 SLQSUIMSetRefreshCallBack . . . 1799
9.12.5.95 SLQSUIMSetStatusChangeCallBack . . . 1800
9.12.5.96 SLQSVoiceInfoRecCallback . . . 1800
9.12.5.97 SLQSVoiceSetAllCallStatusCallBack . . . 1801
9.12.5.98 SLQSVoiceSetDTMFEventCallBack . . . 1801
9.12.5.99 SLQSVoiceSetOTASPStatusCallBack . . . 1802
9.12.5.100 SLQSVoiceSetPrivacyChangeCallBack . . . 1802
9.12.5.101 SLQSVoiceSetSUPSCallBack . . . 1802
9.12.5.102 SLQSVoiceSetSUPSNotificationCallback . . . 1803
9.12.5.103 SLQSWmsAsyncRawSendCallBack . . . 1803
9.12.5.104 SLQSWmsMemoryFullCallBack . . . 1804
9.12.5.105 SLQSWmsMessageWaitingCallBack . . . 1804
9.13 qaGobiApiDcs.h File Reference . . . 1804
9.13.1 Detailed Description . . . 1805
9.13.2 Macro Definition Documentation . . . 1805
9.13.2.1 LEN . . . 1805
9.13.2.2 PORTNAM_LEN . . . 1805
9.13.3 Function Documentation . . . 1806
9.13.3.1 GetQmapIsSupported . . . 1806
9.13.3.2 QCWWAN2kConnect . . . 1806
9.13.3.3 QCWWAN2kEnumerateDevices . . . 1806
9.13.3.4 QCWWAN2kGetConnectedDeviceID . . . 1807
9.13.3.5 QCWWANConnect . . . 1807
9.13.3.6 QCWWANDisconnect . . . 1808
9.13.3.7 QCWWANEnumerateDevices . . . 1808
9.13.3.8 SetQmapIPTable . . . 1809
9.13.3.9 SetQmapIPv6Table . . . 1809
9.13.3.10 SetSDKImagePath . . . 1809
9.13.3.11 SLQSGetDeviceMode . . . 1810
9.13.3.12 SLQSGetNetStatistic . . . 1810
9.13.3.13 SLQSGetUsbPortNames . . . 1811
9.13.3.14 SLQSKillSDKProcess . . . 1811
9.13.3.15 SLQSSetLoggingMask . . . 1811
9.13.3.16 SLQSStart . . . 1812
9.13.3.17 SLQSStart_AVAgent . . . 1812
9.13.3.18 SLQSStartSrv . . . 1813
9.14 qaGobiApiDms.h File Reference . . . 1813

9.14.1 Detailed Description . . . 1816
9.14.2 Macro Definition Documentation . . . 1816
9.14.2.1 MAX_CUST_ID_LEN . . . 1816
9.14.2.2 MAX_CUST_VALUE_LEN . . . 1816
9.14.2.3 MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1816
9.14.2.4 MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1816
9.14.2.5 MAX_FSN_LENGTH . . . 1816
9.14.3 Typedef Documentation . . . 1816
9.14.3.1 custFeaturesInfo . . . 1816
9.14.3.2 custFeaturesSetting . . . 1818
9.14.3.3 dmsCurrentPRLInfo . . . 1819
9.14.3.4 ERIFileparams . . . 1819
9.14.3.5 serialNumbersInfo . . . 1820
9.14.3.6 SLQSSwiGetHostDevInfoParams . . . 1820
9.14.3.7 SLQSSwiGetOSInfoParams . . . 1821
9.14.3.8 SLQSSwiGetSerialNoExtParams . . . 1822
9.14.3.9 SLQSSwiSetHostDevInfoParams . . . 1822
9.14.3.10 SLQSSwiSetOSInfoParams . . . 1822
9.14.4 Function Documentation . . . 1823
9.14.4.1 ActivateAutomatic . . . 1823
9.14.4.2 GetActivationState . . . 1823
9.14.4.3 GetDeviceCapabilities . . . 1824
9.14.4.4 GetFirmwareRevision . . . 1825
9.14.4.5 GetFirmwareRevisions . . . 1826
9.14.4.6 GetHardwareRevision . . . 1826
9.14.4.7 GetIMSI . . . 1827
9.14.4.8 GetManufacturer . . . 1827
9.14.4.9 GetModelID . . . 1828
9.14.4.10 GetNetworkTime . . . 1828
9.14.4.11 GetOfflineReason . . . 1829
9.14.4.12 GetPower . . . 1829
9.14.4.13 GetPRLVersion . . . 1830
9.14.4.14 GetSerialNumbers . . . 1830
9.14.4.15 GetVoiceNumber . . . 1831
9.14.4.16 ResetToFactoryDefaults . . . 1832
9.14.4.17 SetPower . . . 1832
9.14.4.18 SLQSDmsSwiGetPCInfo . . . 1833
9.14.4.19 SLQSDmsSwiGetResetInfo . . . 1833
9.14.4.20 SLQSDmsSwiIndicationRegister . . . 1833
9.14.4.21 SLQSGetBandCapabilities . . . 1833

9.14.4.22 SLQSGetBandCapability . . . 1834
9.14.4.23 SLQSGetCurrentPRLInfo . . . 1835
9.14.4.24 SLQSGetCustFeatures . . . 1835
9.14.4.25 SLQSGetCustFeaturesV2 . . . 1836
9.14.4.26 SLQSGetERIFile . . . 1836
9.14.4.27 SLQSGetPowerSaveModeConfig . . . 1837
9.14.4.28 SLQSGetSerialNumbers . . . 1837
9.14.4.29 SLQSSetCustFeatures . . . 1837
9.14.4.30 SLQSSetCustFeaturesV2 . . . 1838
9.14.4.31 SLQSSetPowerSaveModeConfig . . . 1838
9.14.4.32 SLQSSwiClearDyingGaspStatistics . . . 1839
9.14.4.33 SLQSSwiGetCrashAction . . . 1839
9.14.4.34 SLQSSwiGetCrashInfo . . . 1839
9.14.4.35 SLQSSwiGetDyingGaspCfg . . . 1840
9.14.4.36 SLQSSwiGetDyingGaspStatistics . . . 1840
9.14.4.37 SLQSSwiGetFSN . . . 1840
9.14.4.38 SLQSSwiGetFwUpdateStatus . . . 1841
9.14.4.39 SLQSSwiGetHostDevInfo . . . 1841
9.14.4.40 SLQSSwiGetOSInfo . . . 1842
9.14.4.41 SLQSSwiGetSerialNoExt . . . 1842
9.14.4.42 SLQSSwiGetUSBComp . . . 1843
9.14.4.43 SLQSSwiSetCrashAction . . . 1843
9.14.4.44 SLQSSwiSetDyingGaspCfg . . . 1844
9.14.4.45 SLQSSwiSetHostDevInfo . . . 1844
9.14.4.46 SLQSSwiSetOSInfo . . . 1844
9.14.4.47 SLQSSwiSetUSBComp . . . 1845
9.14.4.48 SLQSUIMGetState . . . 1845
9.14.4.49 UIMChangePIN . . . 1846
9.14.4.50 UIMGetControlKeyStatus . . . 1846
9.14.4.51 UIMGetICCID . . . 1847
9.14.4.52 UIMGetPINStatus . . . 1848
9.14.4.53 UIMSetControlKeyProtection . . . 1849
9.14.4.54 UIMSetPINProtection . . . 1849
9.14.4.55 UIMUnblockControlKey . . . 1850
9.14.4.56 UIMUnblockPIN . . . 1851
9.14.4.57 UIMVerifyPIN . . . 1852
9.14.4.58 ValidateSPC . . . 1853
9.15 qaGobiApiFms.h File Reference . . . 1853
9.15.1 Detailed Description . . . 1856
9.15.2 Macro Definition Documentation . . . 1857

9.15.2.1 BUILD_ID_LEN . . . 1857
9.15.2.2 BUILD_ID_MAX_LEN . . . 1857
9.15.2.3 DEVICE_OFFLINE . . . 1857
9.15.2.4 DEVICE_RESET . . . 1857
9.15.2.5 DEVICE_SHUTDOWN . . . 1857
9.15.2.6 FIRMWARE_UPDATE_FAIL . . . 1857
9.15.2.7 FIRMWARE_UPDATE_SUCCESS . . . 1857
9.15.2.8 FIRMWARE_UPGRADE_SUCCESS . . . 1857
9.15.2.9 G3K_FIRMWARE_DOWNLOAD . . . 1857
9.15.2.10 GOBI_LISTENTRIES_MAX . . . 1857
9.15.2.11 GOBI_MBN_BUILD_ID_STR_LEN . . . 1857
9.15.2.12 GOBI_MBN_IMG_ID_STR_LEN . . . 1857
9.15.2.13 GOBI_SET_IMG_PREF_RSPLEN . . . 1857
9.15.2.14 IMG_ID_LEN . . . 1857
9.15.2.15 IMGDETAILS_LEN . . . 1857
9.15.2.16 MAX_IMAGE_IDE_ELEMENTS . . . 1857
9.15.2.17 PRI_UPDATE_FAIL . . . 1857
9.15.2.18 SLQSFWINFO_APPVERSION_SZ . . . 1857
9.15.2.19 SLQSFWINFO_BOOTVERSION_SZ . . . 1857
9.15.2.20 SLQSFWINFO_CARRIER_SZ . . . 1857
9.15.2.21 SLQSFWINFO_CUR_CARR_NAME . . . 1857
9.15.2.22 SLQSFWINFO_CUR_CARR_REV . . . 1857
9.15.2.23 SLQSFWINFO_MODELID_SZ . . . 1857
9.15.2.24 SLQSFWINFO_PACKAGEID_SZ . . . 1857
9.15.2.25 SLQSFWINFO_PRIVERSION_SZ . . . 1857
9.15.2.26 SLQSFWINFO_SKU_SZ . . . 1857
9.15.2.27 SPKG_FIRMWARE_DOWNLOAD . . . 1857
9.15.2.28 UNIQUE_ID_LEN . . . 1858
9.15.3 Enumeration Type Documentation . . . 1858
9.15.3.1 eGobiDeviceSeries . . . 1858
9.15.3.2 eGobiImageCarrier . . . 1858
9.15.3.3 eGobiImageGPS . . . 1859
9.15.3.4 eGobiImageRegion . . . 1859
9.15.3.5 eGobiImageTech . . . 1860
9.15.4 Function Documentation . . . 1860
9.15.4.1 DeleteStoredImage . . . 1860
9.15.4.2 DownloadToSlot . . . 1860
9.15.4.3 eGetDeviceSeries . . . 1861
9.15.4.4 GetImagesPreference . . . 1861
9.15.4.5 GetImageStore . . . 1862

9.15.4.6 GetStoredImages . . . 1862
9.15.4.7 SetImagePath . . . 1863
9.15.4.8 SetImagesPreference . . . 1863
9.15.4.9 SLQSDownloadFirmwareToSlot . . . 1864
9.15.4.10 SLQSGetBootVersionNumber . . . 1865
9.15.4.11 SLQSGetFirmwareInfo . . . 1865
9.15.4.12 SLQSGetImageInfo . . . 1866
9.15.4.13 SLQSGetImageInfo_9x15 . . . 1866
9.15.4.14 SLQSGetImageInfoMC77xx . . . 1867
9.15.4.15 SLQSGetImageInfoMC83xx . . . 1867
9.15.4.16 SLQSGetValidFwPriCombinations . . . 1868
9.15.4.17 SLQSIsSpkgFormatRequired . . . 1868
9.15.4.18 SLQSSetCrashStateCheckIgnore . . . 1869
9.15.4.19 SLQSSetSIMBasedImageSwitching . . . 1869
9.15.4.20 SLQSSetSpkgFormatRequired . . . 1869
9.15.4.21 SLQSSwiGetAllCarrierImages . . . 1870
9.15.4.22 SLQSSwiGetFirmwareCurr . . . 1870
9.15.4.23 SLQSUpgradeFirmware9x07 . . . 1871
9.15.4.24 SLQSUpgradeFirmware9x15 . . . 1871
9.15.4.25 upgrade_9x07_9x50_fw . . . 1872
9.15.4.26 upgrade_mc77xx_fw . . . 1873
9.15.4.27 UpgradeFirmware2k . . . 1873
9.16 qaGobiApiIms.h File Reference . . . 1873
9.16.1 Detailed Description . . . 1874
9.16.2 Function Documentation . . . 1874
9.16.2.1 SLQSGetIMSSMSConfig . . . 1874
9.16.2.2 SLQSGetIMSUserConfig . . . 1875
9.16.2.3 SLQSGetIMSVoIPConfig . . . 1875
9.16.2.4 SLQSGetRegMgrConfig . . . 1876
9.16.2.5 SLQSGetSIPConfig . . . 1876
9.16.2.6 SLQSImsConfigIndicationRegister . . . 1877
9.16.2.7 SLQSSetIMSSMSConfig . . . 1877
9.16.2.8 SLQSSetIMSUserConfig . . . 1878
9.16.2.9 SLQSSetIMSVoIPConfig . . . 1878
9.16.2.10 SLQSSetRegMgrConfig . . . 1879
9.16.2.11 SLQSSetSIPConfig . . . 1879
9.17 qaGobiApiImsa.h File Reference . . . 1880
9.17.1 Detailed Description . . . 1880
9.17.2 Function Documentation . . . 1880
9.17.2.1 SLQSGetIMSARegStatus . . . 1880

9.17.2.2 SLQSGetIMSAServiceStatus . . . 1881
9.17.2.3 SLQSGetIMSASupportedFields . . . 1881
9.17.2.4 SLQSGetIMSASupportedMsg . . . 1882
9.17.2.5 SLQSRegisterIMSAIndication . . . 1882
9.18 qaGobiApiLoc.h File Reference . . . 1883
9.18.1 Detailed Description . . . 1884
9.18.2 Macro Definition Documentation . . . 1884
9.18.2.1 MAX_SENSOR_DATA_LEN . . . 1884
9.18.2.2 MAX_TEMP_DATA_LEN . . . 1884
9.18.3 Function Documentation . . . 1884
9.18.3.1 SLQSLOCDelAssData . . . 1884
9.18.3.2 SLQSLOCEventRegister . . . 1885
9.18.3.3 SLQSLOCGetBestAvailPos . . . 1885
9.18.3.4 SLQSLOCGetServer . . . 1885
9.18.3.5 SLQSLOCInjectPosition . . . 1886
9.18.3.6 SLQSLOCInjectSensorData . . . 1886
9.18.3.7 SLQSLOCInjectUTCTime . . . 1887
9.18.3.8 SLQSLOCSetCradleMountConfig . . . 1887
9.18.3.9 SLQSLOCSetExtPowerState . . . 1888
9.18.3.10 SLQSLOCSetOpMode . . . 1888
9.18.3.11 SLQSLOCSetServer . . . 1889
9.18.3.12 SLQSLOCStart . . . 1889
9.18.3.13 SLQSLOCStop . . . 1889
9.18.3.14 SwiLocGetAutoStart . . . 1890
9.18.3.15 SwiLocSetAutoStart . . . 1890
9.19 qaGobiApiNas.h File Reference . . . 1890
9.19.1 Detailed Description . . . 1896
9.19.2 Macro Definition Documentation . . . 1896
9.19.2.1 IMSI_M_S1_LENGTH . . . 1896
9.19.2.2 IMSI_M_S2_LENGTH . . . 1896
9.19.2.3 MAX_DATA_SRV_CAPABILITIES . . . 1896
9.19.2.4 MAX_DESCRIPTION_LENGTH . . . 1896
9.19.2.5 MAX_PILOT_SETS . . . 1896
9.19.2.6 MAX_SERV_SYSTEM_RADIO_INTERFACES . . . 1896
9.19.2.7 NAM_NAME_LENGTH . . . 1896
9.19.2.8 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1896
9.19.2.9 NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE . . . 1896
9.19.2.10 NAS_SIG_INFO_MIN_dB_FLOAT_VALUE . . . 1896
9.19.2.11 NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE . . . 1896
9.19.2.12 PLMN_LENGTH . . . 1896

9.19.2.13 SLQS_SS_INFO_LIST_MAX_ELEMENTS . . . 1896
9.19.2.14 SLQS_SYSTEM_ID_SIZE . . . 1897
9.19.2.15 UATISIZE . . . 1897
9.19.3 Typedef Documentation . . . 1897
9.19.3.1 SlqsNas3GppNetworkRAT . . . 1897
9.19.3.2 slqsNetworkScanInfo . . . 1897
9.19.3.3 sysSelectPrefInfo . . . 1898
9.19.3.4 sysSelectPrefParams . . . 1902
9.19.4 Enumeration Type Documentation . . . 1908
9.19.4.1 _NAMS_RADIO_IF_TECHNOLOGY_ . . . 1908
9.19.4.2 eSYS_SRV_DOMAIN . . . 1908
9.19.4.3 NAS_LTE_CPHY_CA_BW_NRB . . . 1909
9.19.4.4 NAS_LTE_CPHY_SCELL_STATE . . . 1909
9.19.5 Function Documentation . . . 1909
9.19.5.1 GetACCOLC . . . 1909
9.19.5.2 GetANAAAAuthenticationStatus . . . 1909
9.19.5.3 GetCDMANetworkParameters . . . 1910
9.19.5.4 GetHomeNetwork . . . 1912
9.19.5.5 GetHomeNetwork3GPP2 . . . 1912
9.19.5.6 GetNetworkPreference . . . 1914
9.19.5.7 GetRFInfo . . . 1915
9.19.5.8 GetServingNetwork . . . 1915
9.19.5.9 GetServingNetworkCapabilities . . . 1916
9.19.5.10 GetSignalStrengths . . . 1917
9.19.5.11 InitiateDomainAttach . . . 1918
9.19.5.12 InitiateNetworkRegistration . . . 1918
9.19.5.13 PerformNetworkScan . . . 1919
9.19.5.14 SetACCOLC . . . 1919
9.19.5.15 SetCDMANetworkParameters . . . 1920
9.19.5.16 SetNetworkPreference . . . 1921
9.19.5.17 SLQSConfigSigInfo . . . 1922
9.19.5.18 SLQSGetErrorRate . . . 1923
9.19.5.19 SLQSGetNetworkTime . . . 1923
9.19.5.20 SLQSGetOperatorNameData . . . 1923
9.19.5.21 SLQSGetPLMNName . . . 1924
9.19.5.22 SLQSGetServingSystem . . . 1924
9.19.5.23 SLQSGetSignalStrength . . . 1925
9.19.5.24 SLQSGetSysSelectionPref . . . 1925
9.19.5.25 SLQSInitiateNetworkRegistration . . . 1926
9.19.5.26 SLQSNasConfigSigInfo2 . . . 1926

9.19.5.27 SLQSNasGet3GPP2Subscription . . . 1927
9.19.5.28 SLQSNasGetCellLocationInfo . . . 1927
9.19.5.29 SLQSNASGeteDRXParams . . . 1928
9.19.5.30 SLQSNasGetHDRColorCode . . . 1928
9.19.5.31 SLQSNASGetLTECPHYCaInfo . . . 1928
9.19.5.32 SLQSNasGetRFInfo . . . 1929
9.19.5.33 SLQSNasGetSigInfo . . . 1929
9.19.5.34 SLQSNasGetSysInfo . . . 1930
9.19.5.35 SLQSNasGetTxRxInfo . . . 1930
9.19.5.36 SLQSNasIndicationRegister . . . 1931
9.19.5.37 SLQSNasIndicationRegisterExt . . . 1932
9.19.5.38 SLQSNasIndicationRegisterLTECphyCa . . . 1932
9.19.5.39 SLQSNASSeteDRXParams . . . 1932
9.19.5.40 SLQSNASSwiGetChannelLock . . . 1933
9.19.5.41 SLQSNasSwiIndicationRegister . . . 1933
9.19.5.42 SLQSNasSwiModemStatus . . . 1934
9.19.5.43 SLQSNASSwiSetChannelLock . . . 1934
9.19.5.44 SLQSPerformNetworkScan . . . 1934
9.19.5.45 SLQSSetBandPreference . . . 1935
9.19.5.46 SLQSSetSysSelectionPref . . . 1937
9.19.5.47 SLQSSetSysSelectionPrefExt . . . 1937
9.19.5.48 SLQSSwiGetHDRPersonality . . . 1938
9.19.5.49 SLQSSwiGetHDRProtSubtype . . . 1938
9.19.5.50 SLQSSwiGetHRPDStats . . . 1938
9.19.5.51 SLQSSwiGetLteCQI . . . 1939
9.19.5.52 SLQSSwiGetLteSccRxInfo . . . 1939
9.19.5.53 SLQSSwiNetworkDebug . . . 1940
9.19.5.54 SLQSSwiPSDetach . . . 1940
9.20 qaGobiApiOmadm.h File Reference . . . 1940
9.20.1 Detailed Description . . . 1941
9.20.2 Function Documentation . . . 1941
9.20.2.1 OMADMCancelSession . . . 1941
9.20.2.2 OMADMGetPendingNIA . . . 1941
9.20.2.3 OMADMGetSessionInfo . . . 1942
9.20.2.4 OMADMStartSession . . . 1943
9.21 qaGobiApiPds.h File Reference . . . 1943
9.21.1 Detailed Description . . . 1944
9.21.2 Macro Definition Documentation . . . 1945
9.21.2.1 DEFAULTBYTEVALUE . . . 1945
9.21.2.2 DEFAULTLONGVALUE . . . 1945

9.21.2.3 DEFAULTWORDVALUE . . . 1945
9.21.3 Enumeration Type Documentation . . . 1945
9.21.3.1 anonymous enum . . . 1945
9.21.4 Function Documentation . . . 1945
9.21.4.1 ForceXTRADownload . . . 1945
9.21.4.2 GetPDSDefaults . . . 1945
9.21.4.3 GetPDSState . . . 1946
9.21.4.4 GetPortAutomaticTracking . . . 1947
9.21.4.5 GetServiceAutomaticTracking . . . 1947
9.21.4.6 GetXTRAAutomaticDownload . . . 1947
9.21.4.7 GetXTRANetwork . . . 1948
9.21.4.8 GetXTRAValidity . . . 1948
9.21.4.9 PDSInjectTimeReference . . . 1949
9.21.4.10 ResetPDSData . . . 1949
9.21.4.11 SetPDSDefaults . . . 1950
9.21.4.12 SetPDSState . . . 1951
9.21.4.13 SetPortAutomaticTracking . . . 1952
9.21.4.14 SetServiceAutomaticTracking . . . 1952
9.21.4.15 SetXTRAAutomaticDownload . . . 1952
9.21.4.16 SetXTRANetwork . . . 1953
9.21.4.17 SLQSGetAGPSConfig . . . 1953
9.21.4.18 SLQSGetGPSStateInfo . . . 1954
9.21.4.19 SLQSPDSDeterminePosition . . . 1955
9.21.4.20 SLQSPDSInjectAbsoluteTimeReference . . . 1955
9.21.4.21 SLQSPDSInjectPositionData . . . 1955
9.21.4.22 SLQSSetAGPSConfig . . . 1956
9.21.4.23 SLQSSetPositionMethodState . . . 1957
9.21.4.24 StartPDSTrackingSessionExt . . . 1957
9.21.4.25 StopPDSTrackingSession . . . 1958
9.22 qaGobiApiQos.h File Reference . . . 1958
9.22.1 Detailed Description . . . 1959
9.22.2 Macro Definition Documentation . . . 1959
9.22.2.1 MAX_QOS_FILTER_TLV . . . 1959
9.22.2.2 MAX_QOS_SPEC_PER_APN . . . 1959
9.22.3 Function Documentation . . . 1959
9.22.3.1 SetQosMuxID . . . 1959
9.22.3.2 SLQSQosGetFlowStatus . . . 1960
9.22.3.3 SLQSQosGetGranted . . . 1960
9.22.3.4 SLQSQosGetNetworkStatus . . . 1960
9.22.3.5 SLQSQosGetNWProf . . . 1961

9.22.3.6 SLQSQosModify . . . 1961
9.22.3.7 SLQSQosRel . . . 1962
9.22.3.8 SLQSQosReq . . . 1962
9.22.3.9 SLQSQosReset . . . 1963
9.22.3.10 SLQSQosResume . . . 1963
9.22.3.11 SLQSQosSuspend . . . 1964
9.22.3.12 SLQSQosSwiReadApnExtraParams . . . 1964
9.22.3.13 SLQSQosSwiReadDataStats . . . 1965
9.23 qaGobiApiRms.h File Reference . . . 1965
9.23.1 Detailed Description . . . 1965
9.23.2 Function Documentation . . . 1965
9.23.2.1 GetSMSWake . . . 1965
9.23.2.2 SetSMSWake . . . 1966
9.24 qaGobiApiSar.h File Reference . . . 1966
9.24.1 Detailed Description . . . 1967
9.24.2 Enumeration Type Documentation . . . 1967
9.24.2.1 eQMISARRFState . . . 1967
9.24.3 Function Documentation . . . 1968
9.24.3.1 SLQSGetRfSarState . . . 1968
9.24.3.2 SLQSSetRfSarState . . . 1968
9.25 qaGobiApiSms.h File Reference . . . 1969
9.25.1 Detailed Description . . . 1971
9.25.2 Macro Definition Documentation . . . 1971
9.25.2.1 ABSOLUTE_VALIDITY . . . 1971
9.25.2.2 CONFIG_LEN . . . 1971
9.25.2.3 MAX_SMS_ROUTES . . . 1971
9.25.2.4 NUM_OF_SET . . . 1971
9.25.2.5 TIME_DATE_BUF . . . 1971
9.25.2.6 TIME_STAMP_BUF . . . 1971
9.25.3 Typedef Documentation . . . 1971
9.25.3.1 getIndicationRegResp . . . 1971
9.25.3.2 getTransLayerInfoResp . . . 1972
9.25.3.3 getTransNWRegInfoResp . . . 1972
9.25.3.4 qaQmi3GPP2BroadcastCfgInfo . . . 1973
9.25.3.5 qaQmi3GPPBroadcastCfgInfo . . . 1973
9.25.3.6 setIndicationRegReq . . . 1973
9.25.3.7 transLayerInfo . . . 1974
9.25.4 Function Documentation . . . 1974
9.25.4.1 GetSMSCAddress . . . 1974
9.25.4.2 SaveSMS . . . 1975

9.25.4.3 SendSMS . . . 1976
9.25.4.4 SetSMSCAddress . . . 1977
9.25.4.5 SLQSCDMADecodeMTTextMsg . . . 1977
9.25.4.6 SLQSCDMADecodeMTTextMsgExt . . . 1977
9.25.4.7 SLQSCDMAEncodeMOTextMsg . . . 1978
9.25.4.8 SLQSDeleteSMS . . . 1978
9.25.4.9 SLQSGetIndicationRegister . . . 1979
9.25.4.10 SLQSGetMessageWaiting . . . 1980
9.25.4.11 SLQSGetSMS . . . 1980
9.25.4.12 SLQSGetSmsBroadcastConfig . . . 1981
9.25.4.13 SLQSGetSMSList . . . 1982
9.25.4.14 SLQSGetTransLayerInfo . . . 1983
9.25.4.15 SLQSGetTransNWRegInfo . . . 1983
9.25.4.16 SLQSModifySMSStatus . . . 1983
9.25.4.17 SLQSSendAsyncSMS . . . 1984
9.25.4.18 SLQSSendLongSMS . . . 1985
9.25.4.19 SLQSSendSMS . . . 1985
9.25.4.20 SLQSSetIndicationRegister . . . 1986
9.25.4.21 SLQSSetSmsBroadcastActivation . . . 1986
9.25.4.22 SLQSSetSmsBroadcastConfig . . . 1987
9.25.4.23 SLQSSetSmsStorage . . . 1987
9.25.4.24 SLQSSmsGetMaxStorageSize . . . 1988
9.25.4.25 SLQSSmsGetMessageProtocol . . . 1988
9.25.4.26 SLQSSmsSetRoutes . . . 1989
9.25.4.27 SLQSSwiGetSMSStorage . . . 1989
9.25.4.28 SLQSWCDMADecodeLongTextMsg . . . 1990
9.25.4.29 SLQSWCDMADecodeMTTextMsg . . . 1990
9.25.4.30 SLQSWCDMAEncodeMOTextMsg . . . 1991
9.26 qaGobiApiSwi.h File Reference . . . 1991
9.26.1 Detailed Description . . . 1991
9.26.2 Function Documentation . . . 1991
9.26.2.1 SLQSGetPidof . . . 1991
9.26.2.2 SLQSGetSdkVersion . . . 1992
9.26.2.3 SLQSSendRawQMI . . . 1992
9.27 qaGobiApiSwiAudio.h File Reference . . . 1992
9.27.1 Detailed Description . . . 1993
9.27.2 Macro Definition Documentation . . . 1993
9.27.2.1 MAX_LEN_IFACE_TABLE . . . 1993
9.27.3 Function Documentation . . . 1993
9.27.3.1 SLQSGetM2MAudioProfile . . . 1993

9.27.3.2 SLQSGetM2MAudioVolume . . . 1994
9.27.3.3 SLQSGetM2MAVMute . . . 1994
9.27.3.4 SLQSGetM2MSpkrGain . . . 1995
9.27.3.5 SLQSSetM2MAudioAVCFG . . . 1995
9.27.3.6 SLQSSetM2MAudioLPBK . . . 1995
9.27.3.7 SLQSSetM2MAudioNVDef . . . 1996
9.27.3.8 SLQSSetM2MAudioProfile . . . 1996
9.27.3.9 SLQSSetM2MAudioVolume . . . 1996
9.27.3.10 SLQSSetM2MAVMute . . . 1997
9.27.3.11 SLQSSetM2MSpkrGain . . . 1997
9.28 qaGobiApiSwiAvms.h File Reference . . . 1998
9.28.1 Detailed Description . . . 2000
9.28.2 Macro Definition Documentation . . . 2000
9.28.2.1 MAX_AVMS_RETRY_TIMER_NUMBER . . . 2000
9.28.2.2 MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH . . . 2000
9.28.2.3 MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH . . . 2000
9.28.2.4 MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH . . . 2000
9.28.2.5 MAX_SWI_AVMS_SET_SETTING_APN_LENGTH . . . 2000
9.28.2.6 MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS . . . 2000
9.28.2.7 MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH . . . 2000
9.28.2.8 MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH . . . 2000
9.28.2.9 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH . . . 2000
9.28.2.10 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH . . . 2000
9.28.2.11 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH . . . 2000
9.28.2.12 QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH . . . 2000
9.28.3 Typedef Documentation . . . 2000
9.28.3.1 QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp . . . 2000
9.28.3.2 QmiSwiAvmsEventReportConfigResp . . . 2001
9.28.3.3 QmiSwiAvmsEventReportConnectionRequestResp . . . 2001
9.28.3.4 QmiSwiAvmsEventReportDataSessionStatusResp . . . 2002
9.28.3.5 QmiSwiAvmsEventReportHTTPStatusResp . . . 2002
9.28.3.6 QmiSwiAvmsEventReportNotificationResp . . . 2003
9.28.3.7 QmiSwiAvmsEventReportPackageIDResp . . . 2004
9.28.3.8 QmiSwiAvmsEventReportRegStatusResp . . . 2004
9.28.3.9 QmiSwiAvmsEventReportSessionTypeResp . . . 2005
9.28.3.10 QmiSwiAvmsEventReportWAMSParamChangeResp . . . 2005
9.28.3.11 QmiSwiAvmsSetSettingsAPNInfo . . . 2005
9.28.3.12 QmiSwiAvmsSetSettingsConnectionRetryTimers . . . 2005

9.28.3.13 QmiSwiAvmsSetSettingsPeriodInfo . . . 2005
9.28.3.14 SLQSAVMSSetSettingsReq . . . 2005
9.28.3.15 SLQSAVMSSettings . . . 2006
9.28.3.16 SLQSAVMSSettingsAPNInfo . . . 2006
9.28.3.17 SLQSAVMSSettingsConnectionRetryTimers . . . 2006
9.28.3.18 SLQSAVMSSettingsPeriodsInfo . . . 2007
9.28.4 Function Documentation . . . 2007
9.28.4.1 SLQSAVMSGetSessionInfo . . . 2007
9.28.4.2 SLQSAVMSGetSettings . . . 2007
9.28.4.3 SLQSAVMSSelection . . . 2007
9.28.4.4 SLQSAVMSSetEventReport . . . 2008
9.28.4.5 SLQSAVMSSetSettings . . . 2008
9.28.4.6 SLQSAVMSSetSettingsIgnoreAutoReboot . . . 2009
9.28.4.7 SLQSAVMSStartSession . . . 2009
9.28.4.8 SLQSAVMSStopSession . . . 2010
9.29 qaGobiApiSwiDms.h File Reference . . . 2010
9.29.1 Detailed Description . . . 2011
9.29.2 Function Documentation . . . 2011
9.29.2.1 SLQSSwiDmsGetMTU . . . 2011
9.29.2.2 SLQSSwiDmsGetUsbNetNum . . . 2011
9.29.2.3 SLQSSwiDmsSetMTU . . . 2012
9.29.2.4 SLQSSwiDmsSetUsbNetNum . . . 2012
9.30 qaGobiApiSwiOmadms.h File Reference . . . 2012
9.30.1 Detailed Description . . . 2013
9.30.2 Typedef Documentation . . . 2013
9.30.2.1 SLQSOMADMSessionInfo . . . 2013
9.30.2.2 SLQSOMADMSettings . . . 2015
9.30.2.3 SLQSOMADMSettingsReqParams . . . 2016
9.30.2.4 SLQSOMADMSettingsReqParams3 . . . 2017
9.30.3 Function Documentation . . . 2018
9.30.3.1 SLQSOMADMCancelSession . . . 2018
9.30.3.2 SLQSOMADMGetSessionInfo . . . 2018
9.30.3.3 SLQSOMADMGetSettings . . . 2019
9.30.3.4 SLQSOMADMGetSettings2 . . . 2019
9.30.3.5 SLQSOMADMSendSelection . . . 2020
9.30.3.6 SLQSOMADMSendSelection2 . . . 2020
9.30.3.7 SLQSOMADMSetSettings . . . 2021
9.30.3.8 SLQSOMADMSetSettings2 . . . 2022
9.30.3.9 SLQSOMADMSetSettings3 . . . 2022
9.30.3.10 SLQSOMADMStartSession . . . 2022

9.30.3.11 SLQSOMADMStartSession2 . . . 2023
9.31 qaGobiApiTableBandClasses.h File Reference . . . 2023
9.31.1 Detailed Description . . . 2023
9.31.2 Band Classes (Value - Description) . . . 2023
9.31.2.1 LTE Bands . . . 2025
9.32 qaGobiApiTableCallControlReturnReasons.h File Reference . . . 2026
9.32.1 Detailed Description . . . 2026
9.32.2 S1 . . . 2026
9.33 qaGobiApiTableCallEndReasons.h File Reference . . . 2027
9.33.1 Detailed Description . . . 2027
9.33.2 Call end reason codes (Code - Reason) . . . 2027
9.33.2.1 Technology-agnostic call end reasons . . . 2027
9.33.2.2 EVDO CDMA 1xEV-DO . . . 2028
9.33.2.3 WCDMA/GSM call end reasons . . . 2028
9.33.2.4 EVDO CDMA 1xEV-DO . . . 2031
9.33.2.5 call end reason type . . . 2031
9.33.2.6 Mobile IP call end reasons (Type=1) . . . 2031
9.33.2.7 Internal call end reasons (Type=2) . . . 2033
9.33.2.8 Call Manager defined call end reasons (Type=3) . . . 2035
9.33.2.9 3GPP specification defined call end reasons (Type=6) . . . 2040
9.33.2.10 PPP call end reasons (Type=7) . . . 2042
9.33.2.11 EHRPD call end reasons (Type=8) . . . 2042
9.33.2.12 IPV6 call end reasons (Type=9) . . . 2043
9.34 qaGobiApiTableCarrierCodes.h File Reference . . . 2043
9.34.1 Detailed Description . . . 2044
9.34.2 Carrier Codes (Number - Carrier) . . . 2044
9.35 qaGobiApiTableCodingScheme.h File Reference . . . 2045
9.35.1 Detailed Description . . . 2045
9.35.2 S1 . . . 2046
9.35.2.1 Use of bits 3..0\n\n . . . 2046
9.35.3 Coding Group Bits 7..4(0001) . . . 2046
9.35.3.1 use of bits 3..0 . . . 2046
9.35.4 Coding Group Bits 7..4(0010) . . . 2046
9.35.4.1 use of bits 3..0 . . . 2046
9.35.5 Coding Group Bits 7..4(0011) . . . 2047
9.35.5.1 use of bits 3..0 . . . 2047
9.35.6 Coding Group Bits 7..4(01xx) . . . 2047
9.35.6.1 use of bits 3..0 . . . 2047
9.35.7 Coding Group Bits 7..4(1001) . . . 2047
9.35.7.1 Reserved coding groups . . . 2047

9.35.8 Coding Group Bits 7..4(1010..1101) . . . 2048
9.35.8.1 Reserved coding groups . . . 2048
9.35.9 Coding Group Bits 7..4(1110) . . . 2048
9.35.9.1 Defined by the WAP Forum . . . 2048
9.35.10 Coding Group Bits 7..4 (1111) . . . 2048
9.35.10.1 Data coding / message handling . . . 2048
9.35.11 Macro Definition Documentation . . . 2048
9.35.11.1 __GOBI_API_CODING_SCHEME_H__ . . . 2048
9.36 qaGobiApiTableGpsCapabilityCodes.h File Reference . . . 2048
9.36.1 Detailed Description . . . 2048
9.36.2 GPS capability (Value - Capability) . . . 2048
9.37 qaGobiApiTablePowerModes.h File Reference . . . 2049
9.37.1 Detailed Description . . . 2049
9.37.2 Power Modes (Value - Description) . . . 2049
9.38 qaGobiApiTableRadioInterfaces.h File Reference . . . 2049
9.38.1 Detailed Description . . . 2049
9.38.2 Radio interface . . . 2049
9.38.2.1 Technology (Value - Radio Interface Technology) . . . 2049
9.39 qaGobiApiTableRegionCodes.h File Reference . . . 2050
9.39.1 Detailed Description . . . 2050
9.39.2 Region Codes (Code - Region) . . . 2050
9.40 qaGobiApiTableServiceOptions.h File Reference . . . 2050
9.40.1 Detailed Description . . . 2050
9.40.2 Service Option codes (Code - Reason) . . . 2051
9.40.2.1 Description . . . 2051
9.41 qaGobiApiTableSupServiceInfoClasses.h File Reference . . . 2053
9.41.1 Detailed Description . . . 2053
9.41.2 Supplementary Service Information Classes (Value - Service Class) . . . 2053
9.42 qaGobiApiTableSwiAudio.h File Reference . . . 2053
9.42.1 Detailed Description . . . 2053
9.42.2 ACDB Device (Device ID - description) . . . 2053
9.42.3 Physical Interface (Device ID - description - Interface parameters) . . . 2054
9.43 qaGobiApiTableSwiOMADMSessionStatus.h File Reference . . . 2054
9.43.1 Detailed Description . . . 2054
9.43.2 OMA DM Session Status (Session Status - Meaning - Usage) . . . 2054
9.44 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference . . . 2055
9.44.1 Detailed Description . . . 2055
9.44.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage) . . . 2055
9.45 qaGobiApiTableVoiceCallEndReasons.h File Reference . . . 2056
9.45.1 Detailed Description . . . 2056

9.45.2 Voice Call and supplementary services end reason codes (Code - Reason) . . . . . 2056
9.45.2.1 General . . . . . 2056
9.45.2.2 service Errors . . . . . 2058
9.45.2.3 control cause values . . . . . 2059
9.45.2.4 reject causes . . . . . 2061
9.45.2.5 reject causes . . . . . 2061
9.45.2.6 reject causes . . . . . 2061
9.45.2.7 stratum reject causes . . . . . 2062
9.45.2.8 reject causes . . . . . 2062
9.45.2.9 IP end reasons . . . . . 2062
9.46 qaGobiApiTmd.h File Reference . . . . . 2063
9.46.1 Detailed Description . . . . . 2063
9.46.2 Macro Definition Documentation . . . . . 2063
9.46.2.1 MAX_MITIGATION_DEV_ID_LEN . . . . . 2063
9.46.2.2 MAX_MITIGATION_DEV_LIST_LEN . . . . . 2063
9.46.3 Function Documentation . . . . . 2063
9.46.3.1 SLQSTmdDeRegNotMitigationLvl . . . . . 2063
9.46.3.2 SLQSTmdGetMitigationDevList . . . . . 2064
9.46.3.3 SLQSTmdGetMitigationLvl . . . . . 2064
9.46.3.4 SLQSTmdRegNotMitigationLvl . . . . . 2064
9.47 qaGobiApiUim.h File Reference . . . . . 2065
9.47.1 Detailed Description . . . . . 2067
9.47.2 Macro Definition Documentation . . . . . 2067
9.47.2.1 MAX_ACTIVE_PERS_FEATURES . . . . . 2067
9.47.2.2 MAX_CONTENT_LENGTH . . . . . 2067
9.47.2.3 MAX_DESCRIPTION_LENGTH . . . . . 2067
9.47.2.4 MAX_ICCID_LENGTH . . . . . 2067
9.47.2.5 MAX_NO_OF_APPLICATIONS . . . . . 2067
9.47.2.6 MAX_NO_OF_SLOTS . . . . . 2067
9.47.2.7 MAX_PATH_LENGTH . . . . . 2067
9.47.2.8 MAX_PUK_LENGTH . . . . . 2067
9.47.2.9 MAX_SLOTS_STATUS . . . . . 2067
9.47.3 Function Documentation . . . . . 2067
9.47.3.1 SLQSUIMAuthenticate . . . . . 2067
9.47.3.2 SLQSUIMChangePin . . . . . 2068
9.47.3.3 SLQSUIMDepersonalization . . . . . 2068
9.47.3.4 SLQSUIMEventRegister . . . . . 2069
9.47.3.5 SLQSUIMGetCardStatus . . . . . 2069
9.47.3.6 SLQSUIMGetConfiguration . . . . . 2070
9.47.3.7 SLQSUIMGetFileAttributes . . . . . 2070

9.47.3.8 SLQSUIMGetSlotsStatus . . . 2071
9.47.3.9 SLQSUIMPowerDown . . . 2071
9.47.3.10 SLQSUIMPowerUp . . . 2072
9.47.3.11 SLQSUIMReadTransparent . . . 2072
9.47.3.12 SLQSUIMRefreshComplete . . . 2073
9.47.3.13 SLQSUIMRefreshGetLastEvent . . . 2073
9.47.3.14 SLQSUIMRefreshOK . . . 2074
9.47.3.15 SLQSUIMRefreshRegister . . . 2074
9.47.3.16 SLQSUIMReset . . . 2075
9.47.3.17 SLQSUIMSetPinProtection . . . 2075
9.47.3.18 SLQSUIMSwitchSlot . . . 2076
9.47.3.19 SLQSUIMUnblockPin . . . 2076
9.47.3.20 SLQSUIMVerifyPin . . . 2077
9.48 qaGobiApiVoice.h File Reference . . . 2077
9.48.1 Detailed Description . . . 2080
9.48.2 Macro Definition Documentation . . . 2080
9.48.2.1 MAX_CALL_NO_LEN . . . 2080
9.48.2.2 MAX_DESCRIPTION_LENGTH . . . 2080
9.48.2.3 MAX_NO_OF_CALLS . . . 2080
9.48.2.4 MAXUSSDLENGTH . . . 2081
9.48.2.5 PASSWORD_LENGTH . . . 2081
9.48.3 Enumeration Type Documentation . . . 2081
9.48.3.1 serviceClassInformation . . . 2081
9.48.4 Function Documentation . . . 2081
9.48.4.1 AnswerUSSD . . . 2081
9.48.4.2 CancelUSSD . . . 2081
9.48.4.3 OriginateUSSD . . . 2082
9.48.4.4 SLQSOriginateUSSD . . . 2082
9.48.4.5 SLQSVoiceALSSelectLine . . . 2083
9.48.4.6 SLQSVoiceALSSetLineSwitching . . . 2083
9.48.4.7 SLQSVoiceAnswerCall . . . 2084
9.48.4.8 SLQSVoiceBindSubscription . . . 2084
9.48.4.9 SLQSVoiceBurstDTMF . . . 2085
9.48.4.10 SLQSVoiceDialCall . . . 2085
9.48.4.11 SLQSVoiceEndCall . . . 2086
9.48.4.12 SLQSVoiceGetAllCallInfo . . . 2086
9.48.4.13 SLQSVoiceGetCallBarring . . . 2087
9.48.4.14 SLQSVoiceGetCallForwardingStatus . . . 2087
9.48.4.15 SLQSVoiceGetCallInfo . . . 2088
9.48.4.16 SLQSVoiceGetCallWaiting . . . 2089

9.48.4.17 SLQSVoiceGetCLIP . . . 2089
9.48.4.18 SLQSVoiceGetCLIR . . . 2090
9.48.4.19 SLQSVoiceGetCNAP . . . 2090
9.48.4.20 SLQSVoiceGetCOLP . . . 2091
9.48.4.21 SLQSVoiceGetCOLR . . . 2091
9.48.4.22 SLQSVoiceGetConfig . . . 2092
9.48.4.23 SLQSVoiceIndicationRegister . . . 2092
9.48.4.24 SLQSVoiceManageCalls . . . 2093
9.48.4.25 SLQSVoiceOrigUSSDNoWait . . . 2093
9.48.4.26 SLQSVoiceSendFlash . . . 2094
9.48.4.27 SLQSVoiceSetCallBarringPassword . . . 2094
9.48.4.28 SLQSVoiceSetConfig . . . 2095
9.48.4.29 SLQSVoiceSetPreferredPrivacy . . . 2096
9.48.4.30 SLQSVoiceSetSUPSService . . . 2096
9.48.4.31 SLQSVoiceStartContDTMF . . . 2097
9.48.4.32 SLQSVoiceStopContDTMF . . . 2097
9.49 qaGobiApiWds.h File Reference . . . 2098
9.49.1 Detailed Description . . . 2101
9.49.2 Macro Definition Documentation . . . 2101
9.49.2.1 IPV6_ADDRESS_ARRAY_SIZE . . . 2101
9.49.3 Typedef Documentation . . . 2101
9.49.3.1 GetProfileSettingIn . . . 2101
9.49.3.2 GetProfileSettingOut . . . 2102
9.49.3.3 QmiProfileInfo . . . 2102
9.49.3.4 QmiWDSDataBearers . . . 2102
9.49.3.5 QmiWDSDataBearerTechnology . . . 2103
9.49.3.6 slqs3GPPConfigItem . . . 2104
9.49.4 Enumeration Type Documentation . . . 2105
9.49.4.1 qmiDataBearerMasks . . . 2105
9.49.5 Function Documentation . . . 2105
9.49.5.1 GetAutoconnect . . . 2105
9.49.5.2 GetByteTotals . . . 2105
9.49.5.3 GetConnectionRate . . . 2106
9.49.5.4 GetDataBearerTechnology . . . 2107
9.49.5.5 GetDefaultProfile . . . 2108
9.49.5.6 GetDefaultProfileLTE . . . 2109
9.49.5.7 GetDefaultProfileNum . . . 2111
9.49.5.8 GetDormancyState . . . 2111
9.49.5.9 GetIPAddressLTE . . . 2112
9.49.5.10 GetLastMobileIPError . . . 2112

9.49.5.11 GetMobileIP . . . 2113
9.49.5.12 GetMobileIPProfile . . . 2113
9.49.5.13 GetPacketStatistics . . . 2115
9.49.5.14 GetPacketStatus . . . 2115
9.49.5.15 GetSessionDuration . . . 2116
9.49.5.16 GetSessionState . . . 2117
9.49.5.17 iGetByteTotals . . . 2117
9.49.5.18 iGetConnectionRate . . . 2117
9.49.5.19 iGetPacketStatistics . . . 2117
9.49.5.20 iSLQSMISetIPFamilyPreference . . . 2117
9.49.5.21 RMSetTransferStatistics . . . 2117
9.49.5.22 SetActiveMobileIPProfile . . . 2118
9.49.5.23 SetAutoconnect . . . 2118
9.49.5.24 SetDefaultProfile . . . 2119
9.49.5.25 SetDefaultProfileLTE . . . 2120
9.49.5.26 SetDefaultProfileLTEV2 . . . 2121
9.49.5.27 SetDefaultProfileNum . . . 2123
9.49.5.28 SetMobileIP . . . 2123
9.49.5.29 SetMobileIPParameters . . . 2124
9.49.5.30 SetMobileIPProfile . . . 2125
9.49.5.31 SetMuxID . . . 2126
9.49.5.32 SLQSAutoConnect . . . 2126
9.49.5.33 SLQSCreateProfile . . . 2127
9.49.5.34 SLQSDeleteProfile . . . 2127
9.49.5.35 SLQSGet3GPPConfigItem . . . 2128
9.49.5.36 SLQSGetByteTotals . . . 2128
9.49.5.37 SLQSGetConnectionRate . . . 2129
9.49.5.38 SLQSGetCurrDataSystemStat . . . 2129
9.49.5.39 SLQSGetCurrentChannelRate . . . 2130
9.49.5.40 SLQSGetDataBearerTechnology . . . 2130
9.49.5.41 SLQSGetDataBearerTechnologyExt . . . 2131
9.49.5.42 SLQSGetDUNCallInfo . . . 2131
9.49.5.43 SLQSGetPacketStatistics . . . 2132
9.49.5.44 SLQSGetProfile . . . 2132
9.49.5.45 SLQSGetProfileSettings . . . 2134
9.49.5.46 SLQSGetRuntimeSettings . . . 2134
9.49.5.47 SLQSGetSessionState . . . 2135
9.49.5.48 SLQSModifyProfile . . . 2135
9.49.5.49 SLQSResetPacketStatics . . . 2136
9.49.5.50 SLQSSet3GPPConfigItem . . . 2136

9.49.5.51 SLQSSetProfile . . . 2136
9.49.5.52 SLQSSGetDHCPv4ClientConfig . . . 2138
9.49.5.53 SLQSSGetLoopback . . . 2138
9.49.5.54 SLQSSSetDHCPv4ClientConfig . . . 2139
9.49.5.55 SLQSSSetLoopback . . . 2139
9.49.5.56 SLQSStartStopDataSession . . . 2139
9.49.5.57 SLQSWdsGoActive . . . 2140
9.49.5.58 SLQSWdsGoDormant . . . 2140
9.49.5.59 SLQSWdsSetEventReport . . . 2141
9.49.5.60 SLQSWdsSwiPDPRuntimeSettings . . . 2141
9.49.5.61 WDS_IsGobiDevice . . . 2142
9.50 qaNasGetRFBandInfo.h File Reference . . . 2142
9.50.1 Enumeration Type Documentation . . . 2142
9.50.1.1 eQMI_NAS_GET_RF_INFO_RESP . . . 2142
9.50.2 Function Documentation . . . 2142
9.50.2.1 PkQmiNasGetRFBandInfo . . . 2142
9.50.2.2 UpkQmiNasGetRFBandInfo . . . 2142
9.51 qaNasPerformNetworkScan.h File Reference . . . 2142
9.51.1 Macro Definition Documentation . . . 2143
9.51.1.1 FORBIDDEN_INDEX . . . 2143
9.51.1.2 INDEX_ZERO . . . 2143
9.51.1.3 MAX_DESCRIPTION_LENGTH . . . 2143
9.51.1.4 PREFERRED_INDEX . . . 2143
9.51.1.5 QMI_NAS_MAX_INSTANCES . . . 2143
9.51.1.6 QMI_NAS_NETSTATUS_MASK . . . 2143
9.51.1.7 ROAMING_INDEX . . . 2143
9.51.2 Enumeration Type Documentation . . . 2143
9.51.2.1 eQMI_NAS_PERFORM_NETWORK_SCAN_RESP . . . 2143
9.51.3 Function Documentation . . . 2143
9.51.3.1 PkQmiNasPerformNetworkScan . . . 2143
9.51.3.2 UpkQmiNasPerformNetworkScan . . . 2143
9.52 qmerrno.h File Reference . . . 2143
9.52.1 Enumeration Type Documentation . . . 2145
9.52.1.1 eQCWWANError . . . 2145
9.52.1.2 qm_wds_ds_profile_extended_err_codes . . . 2150
9.53 qos.h File Reference . . . 2150
9.53.1 Macro Definition Documentation . . . 2152
9.53.1.1 LITEQMI_MAX_QOS_FILTERS . . . 2152
9.53.1.2 LITEQMI_MAX_QOS_FLOW_PER_APN_STATS . . . 2152
9.53.1.3 LITEQMI_MAX_QOS_FLOWS . . . 2152

9.53.2 Function Documentation . . . 2152
9.53.2.1 pack_qos_BindDataPort . . . 2152
9.53.2.2 pack_qos_SLQSQosGetNetworkStatus . . . 2152
9.53.2.3 pack_qos_SLQSQosSwiReadApnExtraParams . . . 2153
9.53.2.4 pack_qos_SLQSQosSwiReadDataStats . . . 2154
9.53.2.5 pack_qos_SLQSSetQosEventCallback . . . 2154
9.53.2.6 unpack_qos_BindDataPort . . . 2155
9.53.2.7 unpack_qos_SLQSQosGetNetworkStatus . . . 2155
9.53.2.8 unpack_qos_SLQSQosSwiReadApnExtraParams . . . 2156
9.53.2.9 unpack_qos_SLQSQosSwiReadDataStats . . . 2156
9.53.2.10 unpack_qos_SLQSSetQosEventCallback . . . 2157
9.53.2.11 unpack_qos_SLQSSetQosEventCallback_ind . . . 2157
9.53.2.12 unpack_qos_SLQSSetQosNWStatusCallback_ind . . . 2158
9.53.2.13 unpack_qos_SLQSSetQosPriEventCallback_ind . . . 2158
9.53.2.14 unpack_qos_SLQSSetQosStatusCallback_ind . . . 2159
9.54 rms.h File Reference . . . 2160
9.54.1 Macro Definition Documentation . . . 2160
9.54.1.1 __LITEQMI_RMS_H__ . . . 2160
9.54.2 Function Documentation . . . 2160
9.54.2.1 pack_rms_GetSMSWake . . . 2160
9.54.2.2 pack_rms_SetSMSWake . . . 2161
9.54.2.3 unpack_rms_GetSMSWake . . . 2161
9.54.2.4 unpack_rms_SetSMSWake . . . 2161
9.55 sar.h File Reference . . . 2162
9.55.1 Function Documentation . . . 2162
9.55.1.1 pack_sar_SLQSGetRfSarState . . . 2162
9.55.1.2 pack_sar_SLQSSetRfSarState . . . 2162
9.55.1.3 unpack_sar_SLQSGetRfSarState . . . 2163
9.55.1.4 unpack_sar_SLQSSetRfSarState . . . 2163
9.56 sms.h File Reference . . . 2163
9.56.1 Macro Definition Documentation . . . 2167
9.56.1.1 MAX_CDMA_ENC_MO_TXT_MSG_SIZE . . . 2167
9.56.1.2 MAX_MS_TRANSFER_ROUTE_MSG . . . 2167
9.56.1.3 MAX_MSC_ADDRESS_SIZE . . . 2167
9.56.1.4 MAX_MSE_TWS_MSG . . . 2167
9.56.1.5 MAX_SMS_LIST_SIZE . . . 2167
9.56.1.6 MAX_SMS_MESSAGE_SIZE . . . 2167
9.56.1.7 SMS_CONFIG_LEN . . . 2167
9.56.1.8 SMS_MAX_SMS_ROUTES . . . 2167
9.56.1.9 SMS_NUM_OF_SET . . . 2167

9.56.1.10 SMSC_TYPE_LEN . . . 2167
9.56.2 Enumeration Type Documentation . . . 2167
9.56.2.1 eqmiCbkSetStatus . . . 2167
9.56.3 Function Documentation . . . 2167
9.56.3.1 pack_sms_GetSMSCAddress . . . 2167
9.56.3.2 pack_sms_SaveSMS . . . 2168
9.56.3.3 pack_sms_SendSMS . . . 2168
9.56.3.4 pack_sms_SetNewSMSCallback . . . 2168
9.56.3.5 pack_sms_SetSMSCAddress . . . 2169
9.56.3.6 pack_sms_SLQSDeleteSMS . . . 2169
9.56.3.7 pack_sms_SLQSGetIndicationRegister . . . 2170
9.56.3.8 pack_sms_SLQSGetMessageWaiting . . . 2170
9.56.3.9 pack_sms_SLQSGetSMS . . . 2170
9.56.3.10 pack_sms_SLQSGetSmsBroadcastConfig . . . 2171
9.56.3.11 pack_sms_SLQSGetSMSList . . . 2171
9.56.3.12 pack_sms_SLQSGetTransLayerInfo . . . 2171
9.56.3.13 pack_sms_SLQSGetTransNWRegInfo . . . 2172
9.56.3.14 pack_sms_SLQSModifySMSStatus . . . 2172
9.56.3.15 pack_sms_SLQSSendAsyncSMS . . . 2172
9.56.3.16 pack_sms_SLQSSetIndicationRegister . . . 2173
9.56.3.17 pack_sms_SLQSSetSmsBroadcastActivation . . . 2173
9.56.3.18 pack_sms_SLQSSetSmsBroadcastConfig . . . 2173
9.56.3.19 pack_sms_SLQSSetSmsStorage . . . 2174
9.56.3.20 pack_sms_SLQSSmsGetMaxStorageSize . . . 2174
9.56.3.21 pack_sms_SLQSSmsGetMessageProtocol . . . 2175
9.56.3.22 pack_sms_SLQSSmsSetRoutes . . . 2175
9.56.3.23 pack_sms_SLQSSwiGetSMSStorage . . . 2175
9.56.3.24 unpack_sms_GetSMSCAddress . . . 2176
9.56.3.25 unpack_sms_SaveSMS . . . 2176
9.56.3.26 unpack_sms_SendSMS . . . 2176
9.56.3.27 unpack_sms_SetNewSMSCallback . . . 2177
9.56.3.28 unpack_sms_SetNewSMSCallback_ind . . . 2177
9.56.3.29 unpack_sms_SetSMSCAddress . . . 2177
9.56.3.30 unpack_sms_SLQSDeleteSMS . . . 2178
9.56.3.31 unpack_sms_SLQSGetIndicationRegister . . . 2178
9.56.3.32 unpack_sms_SLQSGetMessageWaiting . . . 2178
9.56.3.33 unpack_sms_SLQSGetSMS . . . 2179
9.56.3.34 unpack_sms_SLQSGetSmsBroadcastConfig . . . 2179
9.56.3.35 unpack_sms_SLQSGetSMSList . . . 2179
9.56.3.36 unpack_sms_SLQSGetTransLayerInfo . . . 2180

9.56.3.37 unpack_sms_SLQSGetTransNWRegInfo . . . 2180
9.56.3.38 unpack_sms_SLQSModifySMSStatus . . . 2180
9.56.3.39 unpack_sms_SLQSNWRegInfoCallback_ind . . . 2181
9.56.3.40 unpack_sms_SLQSSendAsyncSMS . . . 2181
9.56.3.41 unpack_sms_SLQSSetIndicationRegister . . . 2181
9.56.3.42 unpack_sms_SLQSSetSmsBroadcastActivation . . . 2182
9.56.3.43 unpack_sms_SLQSSetSmsBroadcastConfig . . . 2182
9.56.3.44 unpack_sms_SLQSSetSmsStorage . . . 2182
9.56.3.45 unpack_sms_SLQSSmsGetMaxStorageSize . . . 2183
9.56.3.46 unpack_sms_SLQSSmsGetMessageProtocol . . . 2183
9.56.3.47 unpack_sms_SLQSSmsSetRoutes . . . 2183
9.56.3.48 unpack_sms_SLQSSwiGetSMSStorage . . . 2184
9.56.3.49 unpack_sms_SLQSTransLayerInfoCallback_ind . . . 2184
9.56.3.50 unpack_sms_SLQSWmsAsyncRawSendCallBack_ind . . . 2184
9.56.3.51 unpack_sms_SLQSWmsMemoryFullCallBack_ind . . . 2185
9.56.3.52 unpack_sms_SLQSWmsMessageWaitingCallBack_ind . . . 2185
9.57 SwiDataTypes.h File Reference . . . 2185
9.57.1 Detailed Description . . . 2186
9.57.2 Macro Definition Documentation . . . 2186
9.57.2.1 QMI_NO_LTE_FW_SUPPORT . . . 2186
9.57.2.2 QMI_TLV_PLACEHOLDER . . . 2186
9.57.2.3 SWI_API . . . 2186
9.57.2.4 UNUSEDPARAM . . . 2186
9.57.3 Typedef Documentation . . . 2186
9.57.3.1 BOOL . . . 2186
9.57.3.2 BYTE . . . 2186
9.57.3.3 CHAR . . . 2186
9.57.3.4 FLOAT . . . 2186
9.57.3.5 INT32 . . . 2186
9.57.3.6 INT8 . . . 2186
9.57.3.7 LPCSTR . . . 2186
9.57.3.8 SHORT . . . 2187
9.57.3.9 ULONG . . . 2187
9.57.3.10 ULONGLONG . . . 2187
9.57.3.11 USHORT . . . 2187
9.57.3.12 WORD . . . 2187
9.58 swidms.h File Reference . . . 2187
9.58.1 Function Documentation . . . 2187
9.58.1.1 pack_swidms_SLQSSwiDmsGetMTU . . . 2187
9.58.1.2 pack_swidms_SLQSSwiDmsGetUsbNetNum . . . 2188

9.58.1.3 pack_swidms_SLQSSwiDmsSetMTU . . . 2188
9.58.1.4 pack_swidms_SLQSSwiDmsSetUsbNetNum . . . 2188
9.58.1.5 unpack_swidms_SLQSSwiDmsGetMTU . . . 2189
9.58.1.6 unpack_swidms_SLQSSwiDmsGetUsbNetNum . . . 2189
9.58.1.7 unpack_swidms_SLQSSwiDmsSetMTU . . . 2190
9.58.1.8 unpack_swidms_SLQSSwiDmsSetUsbNetNum . . . 2190
9.59 swiloc.h File Reference . . . 2190
9.59.1 Function Documentation . . . 2191
9.59.1.1 pack_swiloc_SwiLocGetAutoStart . . . 2191
9.59.1.2 pack_swiloc_SwiLocSetAutoStart . . . 2191
9.59.1.3 unpack_swiloc_SwiLocGetAutoStart . . . 2191
9.59.1.4 unpack_swiloc_SwiLocSetAutoStart . . . 2192
9.60 swioma.h File Reference . . . 2192
9.60.1 Macro Definition Documentation . . . 2193
9.60.1.1 LITEQMI_MAX_SWIOMA_STR_LEN . . . 2193
9.60.2 Function Documentation . . . 2193
9.60.2.1 pack_swioma_SLQSOMADMAlertCallback . . . 2193
9.60.2.2 pack_swioma_SLQSOMADMCancelSession . . . 2194
9.60.2.3 pack_swioma_SLQSOMADMGetSessionInfo . . . 2194
9.60.2.4 pack_swioma_SLQSOMADMGetSettings . . . 2195
9.60.2.5 pack_swioma_SLQSOMADMSendSelection . . . 2195
9.60.2.6 pack_swioma_SLQSOMADMSetSettings . . . 2196
9.60.2.7 pack_swioma_SLQSOMADMStartSession . . . 2197
9.60.2.8 unpack_swioma_SLQSOMADMAlertCallback . . . 2197
9.60.2.9 unpack_swioma_SLQSOMADMAlertCallback_ind . . . 2198
9.60.2.10 unpack_swioma_SLQSOMADMCancelSession . . . 2198
9.60.2.11 unpack_swioma_SLQSOMADMGetSessionInfo . . . 2199
9.60.2.12 unpack_swioma_SLQSOMADMGetSettings . . . 2199
9.60.2.13 unpack_swioma_SLQSOMADMSendSelection . . . 2200
9.60.2.14 unpack_swioma_SLQSOMADMSetSettings . . . 2200
9.60.2.15 unpack_swioma_SLQSOMADMStartSession . . . 2200
9.61 SWIWWANCMAPI.h File Reference . . . 2201
9.62 uim.h File Reference . . . 2201
9.62.1 Macro Definition Documentation . . . 2204
9.62.1.1 MAX_DESCRIPTION_LENGTH . . . 2204
9.62.1.2 MAX_ICCID_LENGTH . . . 2204
9.62.1.3 MAX_NO_OF_APPLICATIONS . . . 2204
9.62.1.4 MAX_NO_OF_SLOTS . . . 2204
9.62.1.5 MAX_SLOTS_STATUS . . . 2204
9.62.1.6 UIM_MAX_ACTIVE_PERS_FEATURES . . . 2204

9.62.1.7 UIM_MAX_CONTENT_LENGTH . . . 2204
9.62.1.8 UIM_MAX_DESCRIPTION_LENGTH . . . 2204
9.62.1.9 UIM_MAX_NO_OF_APPLICATIONS . . . 2204
9.62.1.10 UIM_MAX_NO_OF_SLOTS . . . 2204
9.62.1.11 UIM_UINT8_MAX_STRING_SZ . . . 2204
9.62.2 Function Documentation . . . 2204
9.62.2.1 pack_uim_ChangePin . . . 2204
9.62.2.2 pack_uim_GetCardStatus . . . 2204
9.62.2.3 pack_uim_ReadTransparent . . . 2205
9.62.2.4 pack_uim_SetPinProtection . . . 2205
9.62.2.5 pack_uim_SLQSUIMAuthenticate . . . 2206
9.62.2.6 pack_uim_SLQSUIMDepersonalization . . . 2206
9.62.2.7 pack_uim_SLQSUIMEventRegister . . . 2206
9.62.2.8 pack_uim_SLQSUIMGetConfiguration . . . 2207
9.62.2.9 pack_uim_SLQSUIMGetFileAttributes . . . 2207
9.62.2.10 pack_uim_SLQSUIMGetSlotsStatus . . . 2207
9.62.2.11 pack_uim_SLQSUIMPowerDown . . . 2208
9.62.2.12 pack_uim_SLQSUIMPowerUp . . . 2208
9.62.2.13 pack_uim_SLQSUIMRefreshComplete . . . 2209
9.62.2.14 pack_uim_SLQSUIMRefreshGetLastEvent . . . 2209
9.62.2.15 pack_uim_SLQSUIMRefreshOK . . . 2209
9.62.2.16 pack_uim_SLQSUIMRefreshRegister . . . 2210
9.62.2.17 pack_uim_SLQSUIMReset . . . 2210
9.62.2.18 pack_uim_SLQSUIMSwitchSlot . . . 2210
9.62.2.19 pack_uim_UnblockPin . . . 2211
9.62.2.20 pack_uim_VerifyPin . . . 2211
9.62.2.21 unpack_uim_ChangePin . . . 2212
9.62.2.22 unpack_uim_GetCardStatus . . . 2212
9.62.2.23 unpack_uim_ReadTransparent . . . 2212
9.62.2.24 unpack_uim_SetPinProtection . . . 2213
9.62.2.25 unpack_uim_SetUimSlotStatusChangeCallback_ind . . . 2213
9.62.2.26 unpack_uim_SLQSUIMAuthenticate . . . 2213
9.62.2.27 unpack_uim_SLQSUIMDepersonalization . . . 2214
9.62.2.28 unpack_uim_SLQSUIMEventRegister . . . 2214
9.62.2.29 unpack_uim_SLQSUIMGetConfiguration . . . 2214
9.62.2.30 unpack_uim_SLQSUIMGetFileAttributes . . . 2215
9.62.2.31 unpack_uim_SLQSUIMGetSlotsStatus . . . 2215
9.62.2.32 unpack_uim_SLQSUIMPowerDown . . . 2215
9.62.2.33 unpack_uim_SLQSUIMPowerUp . . . 2216
9.62.2.34 unpack_uim_SLQSUIMRefreshCallback_Ind . . . 2216

9.62.2.35 unpack_uim_SLQSUIMRefreshComplete . . . 2216
9.62.2.36 unpack_uim_SLQSUIMRefreshGetLastEvent . . . 2217
9.62.2.37 unpack_uim_SLQSUIMRefreshOK . . . 2217
9.62.2.38 unpack_uim_SLQSUIMRefreshRegister . . . 2217
9.62.2.39 unpack_uim_SLQSUIMReset . . . 2218
9.62.2.40 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind . . . 2218
9.62.2.41 unpack_uim_SLQSUIMSwitchSlot . . . 2218
9.62.2.42 unpack_uim_UnblockPin . . . 2219
9.62.2.43 unpack_uim_VerifyPin . . . 2219
9.63 voice.h File Reference . . . 2219
9.63.1 Detailed Description . . . 2224
9.63.2 S1 . . . 2224
9.63.3 Macro Definition Documentation . . . 2225
9.63.3.1 BARRING_PASSWORD_LENGTH . . . 2225
9.63.3.2 MAX_VOICE_CALL_NO_LEN . . . 2225
9.63.3.3 MAX_VOICE_DESCRIPTION_LENGTH . . . 2225
9.63.3.4 MAXVOICEUSSDLENGTH . . . 2225
9.63.3.5 VOICE_MAX_NO_OF_CALLS . . . 2225
9.63.4 Enumeration Type Documentation . . . 2225
9.63.4.1 liteServiceClassInformation . . . 2225
9.63.5 Function Documentation . . . 2225
9.63.5.1 pack_voice_AnswerUSSD . . . 2225
9.63.5.2 pack_voice_CancelUSSD . . . 2225
9.63.5.3 pack_voice_OriginateUSSD . . . 2226
9.63.5.4 pack_voice_SLQSOriginateUSSD . . . 2226
9.63.5.5 pack_voice_SLQSVoiceALSSelectLine . . . 2227
9.63.5.6 pack_voice_SLQSVoiceALSSetLineSwitching . . . 2227
9.63.5.7 pack_voice_SLQSVoiceAnswerCall . . . 2227
9.63.5.8 pack_voice_SLQSVoiceBindSubscription . . . 2228
9.63.5.9 pack_voice_SLQSVoiceBurstDTMF . . . 2228
9.63.5.10 pack_voice_SLQSVoiceDialCall . . . 2228
9.63.5.11 pack_voice_SLQSVoiceEndCall . . . 2229
9.63.5.12 pack_voice_SLQSVoiceGetAllCallInfo . . . 2229
9.63.5.13 pack_voice_SLQSVoiceGetCallBarring . . . 2230
9.63.5.14 pack_voice_SLQSVoiceGetCallForwardingStatus . . . 2230
9.63.5.15 pack_voice_SLQSVoiceGetCallInfo . . . 2230
9.63.5.16 pack_voice_SLQSVoiceGetCallWaiting . . . 2231
9.63.5.17 pack_voice_SLQSVoiceGetCLIP . . . 2231
9.63.5.18 pack_voice_SLQSVoiceGetCLIR . . . 2231
9.63.5.19 pack_voice_SLQSVoiceGetCNAP . . . 2232

9.63.5.20 pack_voice_SLQSVoiceGetCOLP . . . 2232
9.63.5.21 pack_voice_SLQSVoiceGetCOLR . . . 2232
9.63.5.22 pack_voice_SLQSVoiceGetConfig . . . 2233
9.63.5.23 pack_voice_SLQSVoiceIndicationRegister . . . 2233
9.63.5.24 pack_voice_SLQSVoiceManageCalls . . . 2233
9.63.5.25 pack_voice_SLQSVoiceOrigUSSDNoWait . . . 2234
9.63.5.26 pack_voice_SLQSVoiceSendFlash . . . 2234
9.63.5.27 pack_voice_SLQSVoiceSetCallBarringPassword . . . 2235
9.63.5.28 pack_voice_SLQSVoiceSetConfig . . . 2235
9.63.5.29 pack_voice_SLQSVoiceSetPreferredPrivacy . . . 2235
9.63.5.30 pack_voice_SLQSVoiceSetSUPSService . . . 2236
9.63.5.31 pack_voice_SLQSVoiceStartContDTMF . . . 2236
9.63.5.32 pack_voice_SLQSVoiceStopContDTMF . . . 2236
9.63.5.33 unpack_voice_allCallStatusCallback_ind . . . 2237
9.63.5.34 unpack_voice_AnswerUSSD . . . 2237
9.63.5.35 unpack_voice_CancelUSSD . . . 2237
9.63.5.36 unpack_voice_DTMFEventCallback_ind . . . 2238
9.63.5.37 unpack_voice_OriginateUSSD . . . 2238
9.63.5.38 unpack_voice_OTASPStatusCallback_ind . . . 2238
9.63.5.39 unpack_voice_SLQSOriginateUSSD . . . 2239
9.63.5.40 unpack_voice_SLQSVoiceALSSelectLine . . . 2239
9.63.5.41 unpack_voice_SLQSVoiceALSSetLineSwitching . . . 2239
9.63.5.42 unpack_voice_SLQSVoiceAnswerCall . . . 2240
9.63.5.43 unpack_voice_SLQSVoiceBindSubscription . . . 2240
9.63.5.44 unpack_voice_SLQSVoiceBurstDTMF . . . 2240
9.63.5.45 unpack_voice_SLQSVoiceDialCall . . . 2241
9.63.5.46 unpack_voice_SLQSVoiceEndCall . . . 2241
9.63.5.47 unpack_voice_SLQSVoiceGetAllCallInfo . . . 2242
9.63.5.48 unpack_voice_SLQSVoiceGetCallBarring . . . 2242
9.63.5.49 unpack_voice_SLQSVoiceGetCallForwardingStatus . . . 2242
9.63.5.50 unpack_voice_SLQSVoiceGetCallInfo . . . 2243
9.63.5.51 unpack_voice_SLQSVoiceGetCallWaiting . . . 2243
9.63.5.52 unpack_voice_SLQSVoiceGetCLIP . . . 2243
9.63.5.53 unpack_voice_SLQSVoiceGetCLIR . . . 2244
9.63.5.54 unpack_voice_SLQSVoiceGetCNAP . . . 2244
9.63.5.55 unpack_voice_SLQSVoiceGetCOLP . . . 2244
9.63.5.56 unpack_voice_SLQSVoiceGetCOLR . . . 2245
9.63.5.57 unpack_voice_SLQSVoiceGetConfig . . . 2245
9.63.5.58 unpack_voice_SLQSVoiceIndicationRegister . . . 2245
9.63.5.59 unpack_voice_SLQSVoiceManageCalls . . . 2246

9.63.5.60 unpack_voice_SLQSVoiceOrigUSSDNoWait . . . 2246
9.63.5.61 unpack_voice_SLQSVoiceSendFlash . . . 2246
9.63.5.62 unpack_voice_SLQSVoiceSetCallBarringPassword . . . 2247
9.63.5.63 unpack_voice_SLQSVoiceSetConfig . . . 2247
9.63.5.64 unpack_voice_SLQSVoiceSetPreferredPrivacy . . . 2247
9.63.5.65 unpack_voice_SLQSVoiceSetSUPSService . . . 2248
9.63.5.66 unpack_voice_SLQSVoiceStartContDTMF . . . 2248
9.63.5.67 unpack_voice_SLQSVoiceStopContDTMF . . . 2248
9.63.5.68 unpack_voice_SLQSVoiceSUPSCallback_ind . . . 2249
9.63.5.69 unpack_voice_SUPSNotificationCallback_ind . . . 2249
9.63.5.70 unpack_voice_USSDNotificationCallback_ind . . . 2249
9.63.5.71 unpack_voice_VoiceInfoRecCallback_ind . . . 2250
9.63.5.72 unpack_voice_voicePrivacyChangeCallback_ind . . . 2250
9.64 wds.h File Reference . . . 2250
9.64.1 Macro Definition Documentation . . . 2256
9.64.1.1 BYT_STAT_STAT_MASK . . . 2256
9.64.1.2 IPV6_ADDRESS_ARRAY_SIZE . . . 2256
9.64.1.3 MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE . . . 2256
9.64.1.4 PACK_WDS_IPV4 . . . 2257
9.64.1.5 PACK_WDS_IPV6 . . . 2257
9.64.1.6 WDS_DHCP_MAX_NUM_OPTIONS . . . 2257
9.64.1.7 WDS_DHCP_OPTION_DATA_BUF_SIZE . . . 2257
9.64.2 Typedef Documentation . . . 2257
9.64.2.1 UnpackQmiProfileInfo . . . 2257
9.64.3 Enumeration Type Documentation . . . 2257
9.64.3.1 liteQmiDataBearerMasks . . . 2257
9.64.4 Function Documentation . . . 2257
9.64.4.1 pack_wds_DHCPv4ClientLeaseChange . . . 2257
9.64.4.2 pack_wds_GetAutoconnect . . . 2258
9.64.4.3 pack_wds_GetByteTotals . . . 2258
9.64.4.4 pack_wds_GetConnectionRate . . . 2258
9.64.4.5 pack_wds_GetDataBearerTechnology . . . 2259
9.64.4.6 pack_wds_GetDefaultProfile . . . 2259
9.64.4.7 pack_wds_GetDefaultProfileNum . . . 2260
9.64.4.8 pack_wds_GetDormancyState . . . 2260
9.64.4.9 pack_wds_GetLastMobileIPError . . . 2261
9.64.4.10 pack_wds_GetMobileIP . . . 2261
9.64.4.11 pack_wds_GetMobileIPProfile . . . 2261
9.64.4.12 pack_wds_GetPacketStatistics . . . 2262
9.64.4.13 pack_wds_GetPacketStatus . . . 2262

9.64.4.14 pack_wds_GetSessionDuration . . . 2263
9.64.4.15 pack_wds_GetSessionState . . . 2263
9.64.4.16 pack_wds_RMSetTransferStatistics . . . 2264
9.64.4.17 pack_wds_SetAutoconnect . . . 2264
9.64.4.18 pack_wds_SetDefaultProfile . . . 2264
9.64.4.19 pack_wds_SetDefaultProfileNum . . . 2265
9.64.4.20 pack_wds_SetMobileIP . . . 2265
9.64.4.21 pack_wds_SetMobileIPParameters . . . 2266
9.64.4.22 pack_wds_SetMobileIPProfile . . . 2266
9.64.4.23 pack_wds_SetMuxID . . . 2266
9.64.4.24 pack_wds_SLQSCreateProfile . . . 2267
9.64.4.25 pack_wds_SLQSDeleteProfile . . . 2267
9.64.4.26 pack_wds_SLQSGet3GPPConfigItem . . . 2268
9.64.4.27 pack_wds_SLQSGetCurrDataSystemStat . . . 2268
9.64.4.28 pack_wds_SLQSGetCurrentChannelRate . . . 2269
9.64.4.29 pack_wds_SLQSGetDataBearerTechnology . . . 2269
9.64.4.30 pack_wds_SLQSGetDUNCallInfo . . . 2269
9.64.4.31 pack_wds_SLQSGetProfileSettings . . . 2270
9.64.4.32 pack_wds_SLQSGetRuntimeSettings . . . 2270
9.64.4.33 pack_wds_SLQSModifyProfile . . . 2271
9.64.4.34 pack_wds_SLQSResetPacketStatics . . . 2271
9.64.4.35 pack_wds_SLQSSet3GPPConfigItem . . . 2272
9.64.4.36 pack_wds_SLQSSetIPFamilyPreference . . . 2272
9.64.4.37 pack_wds_SLQSSetWdsEventCallback . . . 2272
9.64.4.38 pack_wds_SLQSSGetDHCPv4ClientConfig . . . 2273
9.64.4.39 pack_wds_SLQSSGetLoopback . . . 2273
9.64.4.40 pack_wds_SLQSSSetDHCPv4ClientConfig . . . 2274
9.64.4.41 pack_wds_SLQSSSetLoopback . . . 2274
9.64.4.42 pack_wds_SLQSStartDataSession . . . 2274
9.64.4.43 pack_wds_SLQSStopDataSession . . . 2275
9.64.4.44 pack_wds_SLQSWdsGoActive . . . 2275
9.64.4.45 pack_wds_SLQSWdsGoDormant . . . 2276
9.64.4.46 pack_wds_SLQSWdsSetEventReport . . . 2276
9.64.4.47 pack_wds_SLQSWdsSwiPDPRuntimeSettings . . . 2277
9.64.4.48 unpack_wds_DHCPv4ClientLease_ind . . . 2277
9.64.4.49 unpack_wds_DHCPv4ClientLeaseChange . . . 2277
9.64.4.50 unpack_wds_GetAutoconnect . . . 2278
9.64.4.51 unpack_wds_GetByteTotals . . . 2278
9.64.4.52 unpack_wds_GetConnectionRate . . . 2278
9.64.4.53 unpack_wds_GetDataBearerTechnology . . . 2279

9.64.4.54 unpack_wds_GetDefaultProfile . . . 2279
9.64.4.55 unpack_wds_GetDefaultProfileNum . . . 2279
9.64.4.56 unpack_wds_GetDormancyState . . . 2280
9.64.4.57 unpack_wds_GetLastMobileIPError . . . 2280
9.64.4.58 unpack_wds_GetMobileIP . . . 2280
9.64.4.59 unpack_wds_GetMobileIPProfile . . . 2281
9.64.4.60 unpack_wds_GetPacketStatistics . . . 2281
9.64.4.61 unpack_wds_GetPacketStatus . . . 2281
9.64.4.62 unpack_wds_GetSessionDuration . . . 2282
9.64.4.63 unpack_wds_GetSessionState . . . 2282
9.64.4.64 unpack_wds_RMSetTransferStatistics . . . 2282
9.64.4.65 unpack_wds_RMTransferStatistics_ind . . . 2283
9.64.4.66 unpack_wds_SetAutoconnect . . . 2283
9.64.4.67 unpack_wds_SetDefaultProfile . . . 2283
9.64.4.68 unpack_wds_SetDefaultProfileNum . . . 2284
9.64.4.69 unpack_wds_SetMobileIP . . . 2284
9.64.4.70 unpack_wds_SetMobileIPParameters . . . 2284
9.64.4.71 unpack_wds_SetMobileIPProfile . . . 2285
9.64.4.72 unpack_wds_SetMuxID . . . 2285
9.64.4.73 unpack_wds_SLQSCreateProfile . . . 2285
9.64.4.74 unpack_wds_SLQSDeleteProfile . . . 2286
9.64.4.75 unpack_wds_SLQSDUNCallInfoCallBack_ind . . . 2286
9.64.4.76 unpack_wds_SLQSGet3GPPConfigItem . . . 2286
9.64.4.77 unpack_wds_SLQSGetCurrDataSystemStat . . . 2287
9.64.4.78 unpack_wds_SLQSGetCurrentChannelRate . . . 2287
9.64.4.79 unpack_wds_SLQSGetDataBearerTechnology . . . 2287
9.64.4.80 unpack_wds_SLQSGetDUNCallInfo . . . 2288
9.64.4.81 unpack_wds_SLQSGetProfileSettings . . . 2288
9.64.4.82 unpack_wds_SLQSGetRuntimeSettings . . . 2288
9.64.4.83 unpack_wds_SLQSModifyProfile . . . 2289
9.64.4.84 unpack_wds_SLQSResetPacketStatics . . . 2289
9.64.4.85 unpack_wds_SLQSSet3GPPConfigItem . . . 2289
9.64.4.86 unpack_wds_SLQSSetIPFamilyPreference . . . 2290
9.64.4.87 unpack_wds_SLQSSetPacketSrvStatusCallback . . . 2290
9.64.4.88 unpack_wds_SLQSSetWdsEventCallback . . . 2290
9.64.4.89 unpack_wds_SLQSSetWdsEventCallback_ind . . . 2291
9.64.4.90 unpack_wds_SLQSSGetDHCPv4ClientConfig . . . 2291
9.64.4.91 unpack_wds_SLQSSGetLoopback . . . 2291
9.64.4.92 unpack_wds_SLQSSSetDHCPv4ClientConfig . . . 2292
9.64.4.93 unpack_wds_SLQSSSetLoopback . . . 2292

9.64.4.94 unpack_wds_SLQSStartDataSession . . . . . 2292
9.64.4.95 unpack_wds_SLQSStopDataSession . . . . . 2293
9.64.4.96 unpack_wds_SLQSWdsGoActive . . . . . 2293
9.64.4.97 unpack_wds_SLQSWdsGoDormant . . . . . 2293
9.64.4.98 unpack_wds_SLQSWdsSetEventReport . . . . . 2294
9.64.4.99 unpack_wds_SLQSWdsSwiPDPRuntimeSettings . . . . . 2294

**Index** **2295**

# 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:


Device Connectivity Service (DCS) . . . 39
Wireless Data Service (WDS) . . . 40
Device Management Service (DMS) . . . 41
Network Access Service (NAS) . . . 42
CallBacK registration (CBK) . . . 43
Short Message Service (SMS) . . . 44
Position Determination Service (PDS) . . . 45
Card Application Toolkit (CAT) . . . 46
Remote Management Service (RMS) . . . 47
Firmware Management Service (FMS) . . . 48
Open Mobile Alliance Service (OMA) . . . 49
Specific Absorption Rate (SAR) . . . 50
SWI Open Mobile Alliance Service (SWIOMA) . . . 51
Voice Service (VOICE) . . . 52
Non-service specific APIs (SWI) . . . 53
User Identity Module Service (UIM) . . . 54
Audio Service (AUDIO) . . . 55
Quality of Service (QOS) . . . 56
IMS Service (IMS) . . . 57
SWI Audio Service(SWIAUDIO) . . . 58
Location Service(LOC) . . . 59
Thermal Mitigation Device(TMD) . . . 60
AVMS (SWIAVMS) . . . 61
SWIDMS . . . 62

# Chapter 3

# Namespace Index

## 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Tables . . . . . 63

# Chapter 4

# Data Structure Index

## 4.1 Data Structures

Here are the data structures with brief descriptions:


_getIndicationRegResp . . . 65
_GetProfileSettingIn . . . 66
_GetProfileSettingOut . . . 66
_getResetInfoNotification . . . 67
_getTransLayerInfoResp . . . 68
_getTransNWRegInfoResp . . . 68
_MitigationDevInfo . . . 69
_modemTempNotification . . . 69
_packetSrvStatus . . . 70
_qaQmi3GPP2BroadcastCfgInfo . . . 72
_qaQmi3GPPBroadcastCfgInfo . . . 73
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp . . . 73
_QmiSwiAvmsEventReportConfigResp . . . 75
_QmiSwiAvmsEventReportConnectionRequestResp . . . 76
_QmiSwiAvmsEventReportDataSessionStatusResp . . . 77
_QmiSwiAvmsEventReportHTTPStatusResp . . . 77
_QmiSwiAvmsEventReportNotificationResp . . . 79
_QmiSwiAvmsEventReportPackageIDResp . . . 80
_QmiSwiAvmsEventReportRegStatusResp . . . 80
_QmiSwiAvmsEventReportSessionTypeResp . . . 80
_QmiSwiAvmsEventReportWAMSParamChangeResp . . . 81
_QmiSwiAvmsSetSettingsAPNInfo . . . 81
_QmiSwiAvmsSetSettingsConnectionRetryTimers . . . 82
_QmiSwiAvmsSetSettingsPeriodInfo . . . 82
_setIndicationRegReq . . . 82
_slqs3GPPConfigItem . . . 83
_SLQSAVMSSetSettingsReqPara . . . 85
_SLQSAVMSSettings . . . 87
_SLQSAVMSSettingsAPNInfo . . . 88
_SLQSAVMSSettingsConnectionRetryTimers . . . 88
_SLQSAVMSSettingsPeriodsInfo . . . 88
_SlqsNas3GppNetworkRAT_ . . . 89
_slqsNetworkScanInfo . . . 89
_SLQSOMADMSessionInfo . . . 91
_SLQSOMADMSettings . . . 93
_SLQSOMADMSettingsReqParams . . . 95
_SLQSOMADMSettingsReqParams3 . . . 96
_SLQSSwiGetHostDevInfoParams . . . 97

_SLQSSwiGetOSInfoParams . . . 98
_SLQSSwiGetSerialNoExtParams . . . 99
_SLQSSwiSetHostDevInfoParams . . . 99
_SLQSSwiSetOSInfoParams . . . 100
_SwiAvmsEventInd . . . 101
_sysSelectPrefInfo . . . 102
_sysSelectPrefParams . . . 107
_transLayerinfo . . . 113
_transLayerInfoNotification . . . 113
_transNWRegInfoNotification . . . 114
accelAcceptReady_s . . . 115
accelTempAcceptReady_s . . . 115
acqOrderPref . . . 116
ActPilotPNElement . . . 117
AddCDMASysInfo . . . 117
AddSysInfo . . . 118
airTimer . . . 118
allCallsAlphaIDInfo . . . 119
allCallsDiagInfo . . . 119
allCallsUUSInfo . . . 120
alphaIDInfo . . . 120
altitudeSrcInfo . . . 121
altSrcInfo_t . . . 122
appStats . . . 123
appStatus . . . 126
arrAlertingPattern . . . 129
arrAlertingType . . . 130
arrAlphaID . . . 131
arrCalledPartyNum . . . 131
arrCallEndReason . . . 132
arrCallInfo . . . 133
arrConnectPartyNum . . . 133
arrDiagInfo . . . 134
arrRedirPartyNum . . . 134
arrRemotePartyName . . . 135
arrRemotePartyNum . . . 135
arrSvcOption . . . 136
arrUUSInfo . . . 137
authenticateResult . . . 137
authenticationData . . . 138
BandCapabilityResp . . . 139
BdsSV . . . 142
BdsSVInfo . . . 142
BroadcastConfig . . . 143
burstDTMFInfo . . . 144
CallBarringSysInfo . . . 144
callBarStatus . . . 145
calledPartyInfo . . . 146
calledPartySubAdd . . . 148
callerIDInfo . . . 149
callFwdTypeAndPlan . . . 149
callFWExtInfo . . . 150
callFWInfo . . . 152
callInfo . . . 153
callingPartyInfo . . . 155
cardResult . . . 156
cardStatus . . . 157
CarrierImage_t . . . 158

CatAlPhaIdentifierTlv . . . 159
CatCommonEventTlv . . . 160
CatEndProactiveSessionTlv . . . 160
CATEventDataType . . . 161
CatEventIDDataTlv . . . 161
CatEventListTlv . . . 161
CatRefreshTlv . . . 162
ccSUPSType . . . 162
CDMABroadcastConfig . . . 163
CDMAChannel . . . 164
CDMAECIOThresh . . . 165
CDMAInfo . . . 165
cdmaMsgDecodingParams . . . 166
cdmaMsgDecodingParamsExt . . . 169
cdmaMsgEncodingParams . . . 172
CDMARSSIThresh . . . 174
CDMASSInfo . . . 174
cdmaSSInfo . . . 175
CDMASysInfo . . . 176
CDMASysInfoExt . . . 179
CellDb . . . 180
cellParams . . . 180
changeUIMPIN . . . 181
ChannelRate . . . 182
channelRate . . . 183
ciotAcqOrderPref . . . 183
CLIPResp . . . 184
CLIRResp . . . 185
ClkInfo . . . 186
CNAPResp . . . 187
COLPResp . . . 188
COLRResp . . . 189
CommInfo . . . 189
ConnectionStatus . . . 191
connectionStatus . . . 192
connectNumInfo . . . 193
CrashInfo . . . 194
CrashInfoParams . . . 195
crashInfoParams . . . 196
crashInformation . . . 196
CreateProfileIn . . . 197
CreateProfileOut . . . 198
CsgId . . . 199
CSGID . . . 199
CUGInfo . . . 200
curAMRConfig . . . 201
CurrDataSysStat . . . 202
currentCatEvent . . . 203
CurrentImgList . . . 203
currentPLMN . . . 204
CurrImageInfo . . . 205
CurrNetworkInfo . . . 206
currNetworkInfo . . . 208
custFeaturesInfo . . . 209
custFeaturesSetting . . . 212
custSettingInfo . . . 213
custSettingList . . . 214
dataBearers . . . 215

DataBearerTech . . . 215
DataBearerTechExt . . . 217
dataBearerTechnology . . . 218
dataRate . . . 219
dataSrvCapabilities . . . 220
DataStatusDetail . . . 221
DataULongLongTlv . . . 223
DataULongTlv . . . 223
DcsUsbPortNames . . . 223
delAssistDataStatus . . . 223
depersonalizationInformation . . . 224
detailSvcInfo . . . 225
DeviceConfigDetail . . . 227
DHCPOption . . . 228
DHCPOptionList . . . 228
diagInfo . . . 229
dirNum . . . 229
dms_ActivationStatusTlv . . . 230
dms_OperatingModeTlv . . . 231
dms_PSMActiveTimerIndTlv . . . 231
dms_PSMActiveTimerTlv . . . 232
dms_PSMDurationDueToOOSTlv . . . 232
dms_PSMDurationThresholdTlv . . . 233
dms_PSMEarlyWakeupTimeTlv . . . 233
dms_PSMEnableStateIndTlv . . . 234
dms_PSMEnableStateTlv . . . 234
dms_PSMPeriodicUpdateTimerIndTlv . . . 235
dms_PSMPeriodicUpdateTimerTlv . . . 235
dms_PSMRandomizationWindowTlv . . . 236
dms_TemperatureTlv . . . 236
dms_VoltageTlv . . . 237
dmsCurrentPRLInfo . . . 238
DMScustSettingInfo . . . 238
DMScustSettingList . . . 239
DMSgetCustomFeatureV2 . . . 240
DMSgetCustomInput . . . 241
dmsIndicationRegisterReq . . . 241
dmsPsmCfgParamChange . . . 242
dmsSwiGetPCInfo . . . 242
dmsSwiGetResetInfo . . . 244
Domain . . . 244
DomainNameList . . . 245
DRCParams . . . 245
DTMFInfo . . . 246
DTMFLengths . . . 247
DUNCallInfoInd . . . 248
dunchannelRate . . . 248
ecioListElement . . . 249
ECIOThresh . . . 250
ECTNum . . . 250
ehrpdMTUSizeTlv . . . 251
encryptedPIN1 . . . 252
eriDataparams . . . 252
ERIFileparams . . . 253
errorRateListElement . . . 253
eTWSPLMNInfoTlv . . . 254
extDispRecInfo . . . 255
FactorySequenceNumber . . . 255

fileAttributes . . . 256
fileInfo . . . 259
FirmwareUpdatStat . . . 260
FMSImageElement . . . 262
FMSImageIdElement . . . 262
FMSImageIDEntries . . . 263
FMSImageList . . . 264
FMSPrefImageList . . . 265
fwinfo_s . . . 265
GERANInfo . . . 266
geranInstInfo . . . 268
getAllCallInformation . . . 269
getAllCallRmtPtyName . . . 269
getAllCallRmtPtyNum . . . 270
GetAudioPathConfigReq . . . 270
GetAudioPathConfigResp . . . 271
GetAudioProfileReq . . . 273
GetAudioProfileResp . . . 273
GetAudioVolTLBConfigReq . . . 275
GetAudioVolTLBConfigResp . . . 275
getCallFWExtInfo . . . 276
getCallFWInfo . . . 276
getCustomFeatureV2 . . . 277
getCustomInput . . . 278
getDUNCallInfoReq . . . 278
getDUNCallInfoResp . . . 280
getDyingGaspCfg . . . 282
getDyingGaspStatistics . . . 283
GetErrRateResp . . . 284
GetHRPDStatsResp . . . 285
GetIMSSMSConfigParams . . . 285
GetIMSUserConfigParams . . . 286
GetIMSVoIPConfigResp . . . 287
GetInstIDResp . . . 289
GetM2MAudioProfileReq . . . 290
GetM2MAudioProfileResp . . . 290
GetM2MAudioVolumeReq . . . 291
GetM2MAudioVolumeResp . . . 292
GetM2MAVMuteReq . . . 292
GetM2MAVMuteResp . . . 293
GetM2MSpkrGainReq . . . 293
GetM2MSpkrGainResp . . . 294
getMsgWaitingInfo . . . 294
GetNetworkTimeResp . . . 295
GetRegMgrConfigParams . . . 295
GetSessionIDResp . . . 296
GetSIPConfigResp . . . 296
GnssData . . . 298
gnssSvInfoNotification . . . 299
GPRSQoS . . . 300
GPRSRequestedQoS . . . 301
GPSStateInfo . . . 302
gpsTime_s . . . 306
gsmCellInfo . . . 306
GSMRSSIThresh . . . 308
GSMSrvStatusInfo . . . 308
GSMSysInfo . . . 309
gyroAcceptReady_s . . . 312

gyroTempAcceptReady_s . . . 313
HDRECIOThresh . . . 313
HDRIOThresh . . . 314
HDRPersonalityInd . . . 314
HDRPersonalityResp . . . 315
HDRProtSubtypResp . . . 315
HDRRSSIThresh . . . 316
HDRSINRThresh . . . 317
HDRSINRThreshold . . . 317
HDRSSInfo . . . 318
hdrSSInfo . . . 319
HDRSysInfo . . . 320
homeSIDNID . . . 322
hotSwapStatus . . . 323
hrpdMTUSizeTlv . . . 323
image_info_t . . . 324
ImageElement . . . 325
ImageIdElement . . . 325
ImageIDEntries . . . 326
ImageList . . . 327
IMSAIndRegisterInfo . . . 328
imsaPdpStatusInfo . . . 329
imsaRatStatusInfo . . . 329
IMSARegistrationStatus . . . 330
imsaRegStatusInfo . . . 331
IMSAServiceStatus . . . 331
IMSASupportedFieldsResp . . . 334
IMSASupportedMsgInfo . . . 334
imsaSvcStatusInfo . . . 335
imsCfgIndRegisterInfo . . . 336
imsRegMgrConfigInfo . . . 337
imsSIPConfigInfo . . . 338
imsSMSConfigInfo . . . 339
imsUserConfigInfo . . . 340
imsVoIPConfigInfo . . . 340
IndFieldsList . . . 342
infoInterFreq . . . 343
IOThresh . . . 344
IPv4Addr . . . 345
IPv4Config . . . 345
IPv4Info . . . 346
IPv6Addr . . . 346
IPV6AddressInfo . . . 347
ipv6AddressInfo . . . 347
IPv6Config . . . 348
IPV6GWAddressInfo . . . 348
IPv6Info . . . 349
IPv6TrafCls . . . 349
LibPackGPRSRequestedQoS . . . 350
LibpackProfile3GPP . . . 351
LibpackProfile3GPP2 . . . 358
LibPackprofile_3GPP . . . 363
LibPackprofile_3GPP2 . . . 369
LibPackQosClassID . . . 374
LibPackTFTIDParams . . . 375
LibPackUMTSQoS . . . 377
LibPackUMTSReqQoSSigInd . . . 380
lineCtrlInfo . . . 380

loc_accelAcceptReady . . . 381
loc_accelTempAcceptReady . . . 382
loc_BdsSV . . . 382
loc_BdsSVInfo . . . 383
loc_CellDb . . . 383
loc_ClkInfo . . . 384
loc_GnssData . . . 385
loc_gpsTime . . . 387
loc_gyroAcceptReady . . . 387
loc_gyroTempAcceptReady . . . 388
loc_IPv4Config . . . 389
loc_IPv4Info . . . 389
loc_IPv6Config . . . 390
loc_IPv6Info . . . 390
loc_LocApplicationInfo . . . 391
loc_precisionDilution . . . 392
loc_satelliteInfo . . . 393
loc_sensorDataUsage . . . 395
loc_SV . . . 396
loc_SVInfo . . . 397
loc_svUsedforFix . . . 397
loc_urlAddr . . . 398
loc_URLAddrInfo . . . 398
LocApplicationInfo . . . 399
LocDelAssDataReq . . . 400
LOCEventRegisterReqResp . . . 401
LOCExtPowerStateReqResp . . . 403
LocGetServerReq . . . 403
LocInjectPositionReq . . . 404
LocInjectSensorDataReq . . . 408
LocSetCradleMountReq . . . 409
LocSetServerReq . . . 410
LOCStartReq . . . 411
LOCStopReq . . . 413
LteCQIParm . . . 413
lteEARFCN . . . 414
lteGsmCellInfo . . . 414
LTEInfo . . . 416
LTEInfoInterfreq . . . 418
LTEInfoIntrafreq . . . 418
LTEInfoNeighboringGSM . . . 420
LTEInfoNeighboringWCDMA . . . 421
LteNasReleaseInfo_s . . . 422
LTEOperationMode . . . 422
ltePCI . . . 423
lteRsrpinformation . . . 424
LTERSRPThresh . . . 424
LTERSRQThresh . . . 424
LTERSSIThresh . . . 425
LteSccRxInfoResp . . . 426
LTESigRptCfg . . . 426
LTESigRptConfig . . . 427
lteSnrinformation . . . 428
LTESNRThresh . . . 428
LTESNRThreshold . . . 429
LTESSInfo . . . 430
lteSSInfo . . . 430
LTESysInfo . . . 431

lteWcdmaCellInfo . . . 434
messageModeTlv . . . 435
messageWaitingInfoContent . . . 435
minBasedIMSI . . . 436
mitigationDevList . . . 437
MNRInfo . . . 437
ModifyProfileIn . . . 438
ModifyProfileOut . . . 439
msgWaitingInfo . . . 439
mtuSize3gppTlv . . . 440
MTUSizeReq . . . 440
namName . . . 441
nas_acqOrderPref . . . 441
nas_ActPilotPNElement . . . 442
nas_AddCDMASysInfo . . . 442
nas_AddSysInfo . . . 443
nas_CallBarringSysInfo . . . 444
nas_callBarStatus . . . 444
nas_CDMAChannel . . . 445
nas_CDMAECIOThresh . . . 446
nas_CDMAInfo . . . 446
nas_CDMARSSIThresh . . . 447
nas_CDMASysInfo . . . 448
nas_CDMASysInfoExt . . . 452
nas_cellParams . . . 452
nas_ciotAcqOrderPref . . . 453
nas_CommInfo . . . 454
nas_CSGID . . . 456
nas_currentPLMN . . . 457
nas_dataSrvCapabilities . . . 458
nas_DataStatusDetail . . . 458
nas_detailSvcInfo . . . 461
nas_DeviceConfigDetail . . . 462
nas_dirNum . . . 463
nas_DRCParams . . . 463
nas_ecioListElement . . . 464
nas_ECIOThresh . . . 465
nas_errorRateListElement . . . 465
nas_GERANInfo . . . 466
nas_geranInstInfo . . . 468
nas_gsmCellInfo . . . 469
nas_GSMRSSIThresh . . . 470
nas_GSMSrvStatusInfo . . . 470
nas_GSMSysInfo . . . 471
nas_HDRECIOThresh . . . 474
nas_HDRIOThresh . . . 474
nas_HDRPersonality_Ind_Data . . . 475
nas_HDRRSSIThresh . . . 476
nas_HDRSINRThresh . . . 476
nas_HDRSINRThreshold . . . 477
nas_HDRSysInfo . . . 478
nas_homeSIDNID . . . 480
nas_infoInterFreq . . . 480
nas_IOThresh . . . 481
nas_lteEARFCN . . . 482
nas_lteGsmCellInfo . . . 483
nas_LTEInfo . . . 484
nas_LTEInfoInterfreq . . . 486

nas_LTEInfoIntrafreq . . . 486
nas_LTEInfoNeighboringGSM . . . 488
nas_LTEInfoNeighboringWCDMA . . . 489
nas_LTEOperationMode . . . 490
nas_ltePCI . . . 490
nas_lteRsrpinformation . . . 491
nas_LTERSRPThresh . . . 491
nas_LTERSRQThresh . . . 492
nas_LTERSSIThresh . . . 492
nas_LTESigRptCfg . . . 493
nas_LTESigRptConfig . . . 494
nas_lteSnrinformation . . . 495
nas_LTESNRThresh . . . 495
nas_LTESNRThreshold . . . 496
nas_LTESysInfo . . . 497
nas_lteWcdmaCellInfo . . . 499
nas_minBasedIMSI . . . 500
nas_MNRInfo . . . 501
nas_namName . . . 502
nas_netSelectionPref . . . 502
nas_NetworkStat1x . . . 503
nas_NetworkStatEVDO . . . 505
nas_nmrCellInfo . . . 507
nas_nr5gBandPref . . . 508
nas_operatorNameString . . . 509
nas_OperatorPLMNData . . . 509
nas_operatorPLMNList . . . 510
nas_PhyCaAggPcellInfo . . . 511
nas_PhyCaAggScellDlBw . . . 512
nas_PhyCaAggScellIndex . . . 512
nas_PhyCaAggScellIndType . . . 513
nas_PhyCaAggScellInfo . . . 513
nas_PilotSetData . . . 516
nas_PilotSetParams . . . 517
nas_PLMNNetworkName . . . 517
nas_PLMNNetworkNameData . . . 518
nas_protocolSubtypeElement . . . 519
nas_qaQmi3Gpp2TimeZone . . . 521
nas_QmiNas3GppNetworkInfo . . . 522
nas_QmiNas3GppNetworkRAT . . . 523
nas_QmisNasPcsDigit . . . 524
nas_RankIndicatorTlv . . . 524
nas_RejectReasonTlv . . . 525
nas_RFBandInfoElements . . . 525
nas_RfBandInfoExtFormat . . . 526
nas_RfBandInfoExtFormatElements . . . 527
nas_RfBandInfoList . . . 527
nas_RfBandwidthInfo . . . 528
nas_RfBandwidthInfoElements . . . 529
nas_RfDedicatedBandInfo . . . 529
nas_RfDedicatedBandInfoElements . . . 530
nas_RFInfoTlv . . . 530
nas_roamIndList . . . 532
nas_RSRPThresh . . . 532
nas_rsrqInformation . . . 533
nas_RSRQThresh . . . 534
nas_RSSIThresh . . . 534
nas_rxInfo . . . 535

nas_RxSigInfo . . . 536
nas_rxSignalStrengthListElement . . . 537
nas_SccRxInfo . . . 538
nas_serviceProviderName . . . 539
nas_servSystem . . . 540
nas_sidNid . . . 541
nas_SignalStrengthTlv . . . 542
nas_SLQSSignalStrengthsIndReq . . . 542
nas_SLQSSignalStrengthsInformation . . . 544
nas_SLQSSignalStrengthsTlv . . . 545
nas_SrvStatusInfo . . . 545
nas_sysInfoCommon . . . 546
nas_TDSCDMAECIOThresh . . . 549
nas_TDSCDMARSCPThresh . . . 549
nas_TDSCDMARSSIThresh . . . 550
nas_TDSCDMASINRCONFThresh . . . 550
nas_TDSCDMASINRThresh . . . 551
nas_timeInfo . . . 551
nas_trueIMSI . . . 553
nas_txInfo . . . 554
nas_UMTSInfo . . . 554
nas_UMTSinstInfo . . . 556
nas_umtsLTENbrCell . . . 557
nas_UniversalTime . . . 558
nas_wcdmaCellInfo . . . 559
nas_WCDMAECIOThresh . . . 560
nas_WCDMAInfoLTENeighborCell . . . 560
nas_WCDMARSSIThresh . . . 561
nas_WCDMASysInfo . . . 562
nas_wcdmaUARFCN . . . 565
NASBandPreferenceTlv . . . 566
nasCellLocationInfoResp . . . 566
NASEmergencyModeTlv . . . 567
nasGet3GPP2SubscriptionInfoReq . . . 568
nasGet3GPP2SubscriptionInfoResp . . . 568
nasGeteDRXParamsResp . . . 569
nasGetHDRColorCodeResp . . . 570
nasGetLTECphyCa . . . 570
NasGetLTECphyCaInfo . . . 571
nasGetLTECphyCaResp . . . 572
NasGetRFInfoResp . . . 572
nasGetSigInfoResp . . . 573
nasGetSysInfoResp . . . 574
nasGetTxRxInfoReq . . . 577
nasGetTxRxInfoResp . . . 577
NASGWAcqOrderPrefTlv . . . 578
nasIndicationRegisterReq . . . 578
nasInitNetworkReg . . . 582
NASLTEBandPreferenceTlv . . . 582
NASLteNasReleaseInfoTlv . . . 583
NASModePreferenceTlv . . . 584
NASNetSelPreferenceTlv . . . 584
nasNetworkReject . . . 584
nasNetworkTime . . . 587
nasOperatorNameResp . . . 588
NASOTAMessageTlv . . . 589
NASPhyCaAggPcellInfo . . . 590
NASPhyCaAggScellDlBw . . . 591

NASPhyCaAggScellIndex . . . 591
NASPhyCaAggScellIndType . . . 592
NASPhyCaAggScellInfo . . . 592
nasPLMNNameReq . . . 593
nasPLMNNameResp . . . 594
NASPRLPreferenceTlv . . . 597
NASQmiCbkNasSwiOTAMessageInd . . . 597
NASQmiCbkNasSystemSelPrefInd . . . 598
NASRoamPreferenceTlv . . . 599
NASServDomainPrefTlv . . . 599
NASServingSystemInfo . . . 600
nasSeteDRXParamsReq . . . 601
nasSigInfo . . . 602
nasSwiGetChannelLockResp . . . 603
NasSwiIndReg . . . 604
nasSwiSetChannelLockReq . . . 605
nasSysInfo . . . 606
NASTimeInfoTlv . . . 609
nasTimers . . . 609
netSelectionPref . . . 610
NetStats . . . 610
NetworkDebugResp . . . 612
NetworkStat1x . . . 612
NetworkStatEVDO . . . 615
newMTMessageTlv . . . 616
newPwdData . . . 617
nmrCellInfo . . . 617
nr5gBandPref . . . 619
NSSAudioCtrl . . . 619
NWProfile . . . 620
omaDmConfigTlv . . . 620
omaDmConfigTlvExt . . . 621
omaDmFotaTlv . . . 623
omaDmFotaTlvExt . . . 625
omaDmNotificationsTlv . . . 627
operatorNameString . . . 628
OperatorPLMNData . . . 629
operatorPLMNList . . . 629
pack_dms_ActivateAutomatic_t . . . 630
pack_dms_GetCustFeaturesV2_t . . . 630
pack_dms_ResetToFactoryDefaults_t . . . 631
pack_dms_SetActivationStatusCallback_t . . . 631
pack_dms_SetCrashAction_t . . . 632
pack_dms_SetCustFeature_t . . . 632
pack_dms_SetCustFeaturesV2_t . . . 634
pack_dms_SetEventReport_t . . . 635
pack_dms_SetIndicationRegister_t . . . 636
pack_dms_SetPower_t . . . 636
pack_dms_SetUSBComp_t . . . 637
pack_dms_SLQSDmsSwiIndicationRegister_t . . . 638
pack_dms_SLQSSetPowerSaveModeConfig_t . . . 639
pack_dms_SLQSSwiGetCrashInfo_t . . . 639
pack_dms_SLQSSwiSetDyingGaspCfg_t . . . 640
pack_dms_SLQSSwiSetHostDevInfo_t . . . 640
pack_dms_SLQSSwiSetOSInfo_t . . . 641
pack_dms_SwiSetEventReport_t . . . 642
pack_dms_UIMChangePIN_t . . . 642
pack_dms_UIMGetControlKeyStatus_t . . . 643

pack_dms_UIMGetICCID_t . . . 643
pack_dms_UIMSetControlKeyProtection_t . . . 644
pack_dms_UIMSetPINProtection_t . . . 645
pack_dms_UIMUnblockControlKey_t . . . 645
pack_dms_UIMUnblockPIN_t . . . 646
pack_dms_UIMVerifyPIN_t . . . 647
pack_fms_GetImagesPreference_t . . . 647
pack_fms_GetStoredImages_t . . . 647
pack_fms_SetImagesPreference_t . . . 648
pack_loc_Delete_Assist_Data_t . . . 649
pack_loc_EventRegister_t . . . 650
pack_loc_SetExtPowerState_t . . . 652
pack_loc_SetOperationMode_t . . . 652
pack_loc_SLQSLOCGetBestAvailPos_t . . . 653
pack_loc_SLQSLOCGetServer_t . . . 654
pack_loc_SLQSLOCInjectPosition_t . . . 654
pack_loc_SLQSLOCInjectSensorData_t . . . 659
pack_loc_SLQSLOCInjectUTCTime_t . . . 661
pack_loc_SLQSLOCSetCradleMountConfig_t . . . 661
pack_loc_SLQSLOCSetServer_t . . . 662
pack_loc_Start_t . . . 663
pack_loc_Stop_t . . . 665
pack_nas_InitiateDomainAttach_t . . . 665
pack_nas_SetACCOLC_t . . . 666
pack_nas_SetCDMANetworkParameters_t . . . 666
pack_nas_SetNetworkPreference_t . . . 668
pack_nas_SLQSConfigSigInfo_t . . . 668
pack_nas_SLQSGetPLMNName_t . . . 670
pack_nas_SLQSInitiateNetworkRegistration_t . . . 671
pack_nas_SLQSNasConfigSigInfo2_t . . . 672
pack_nas_SLQSNasGet3GPP2Subscription_t . . . 676
pack_nas_SLQSNasGetTxRxInfo_t . . . 677
pack_nas_SLQSNasIndicationRegisterExt_t . . . 677
pack_nas_SLQSNASSeteDRXParams_t . . . 680
pack_nas_SLQSNasSwiIndicationRegister_t . . . 681
pack_nas_SLQSNASSwiSetChannelLock_t . . . 682
pack_nas_SLQSSetBandPreference_t . . . 683
pack_nas_SLQSSetSignalStrengthsCallback_t . . . 685
pack_nas_SLQSSetSysSelectionPref_t . . . 685
pack_nas_SLQSSetSysSelectionPrefExt_t . . . 691
pack_nas_SLQSSwiPSDetach_t . . . 700
pack_pds_PDSInjectTimeReference_t . . . 700
pack_pds_ResetPDSData_t . . . 701
pack_pds_SetEventReportCallback_t . . . 702
pack_pds_SetPDSDefaults_t . . . 702
pack_pds_SetPDSState_t . . . 703
pack_pds_SetPortAutomaticTracking_t . . . 703
pack_pds_SetServiceAutomaticTracking_t . . . 704
pack_pds_SetXTRAAutomaticDownload_t . . . 704
pack_pds_SetXTRANetwork_t . . . 705
pack_pds_SLQSGetAGPSConfig_t . . . 705
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t . . . 706
pack_pds_SLQSPDSInjectPositionData_t . . . 707
pack_pds_SLQSSetAGPSConfig_t . . . 709
pack_pds_SLQSSetPositionMethodState_t . . . 709
pack_pds_StartPDSTrackingSessionExt_t . . . 710
pack_qmi_t . . . 711
pack_qos_BindDataPort_t . . . 712

pack_qos_SLQSQosSwiReadApnExtraParams_t . . . 713
pack_qos_SLQSQosSwiReadDataStats_t . . . 713
pack_qos_SLQSSetQosEventCallback_t . . . 713
pack_rms_SetSMSWake_t . . . 714
pack_sar_SLQSSetRfSarState_t . . . 714
pack_sms_SaveSMS_t . . . 715
pack_sms_SendSMS_t . . . 716
pack_sms_SetNewSMSCallback_t . . . 717
pack_sms_SetSMSCAddress_t . . . 717
pack_sms_SLQSDeleteSMS_t . . . 717
pack_sms_SLQSGetSMS_t . . . 718
pack_sms_SLQSGetSmsBroadcastConfig_t . . . 719
pack_sms_SLQSGetSMSList_t . . . 719
pack_sms_SLQSModifySMSStatus_t . . . 720
pack_sms_SLQSSendAsyncSMS_t . . . 721
pack_sms_SLQSSetIndicationRegister_t . . . 721
pack_sms_SLQSSetSmsBroadcastActivation_t . . . 722
pack_sms_SLQSSetSmsBroadcastConfig_t . . . 722
pack_sms_SLQSSetSmsStorage_t . . . 723
pack_sms_SLQSSmsGetMaxStorageSize_t . . . 724
pack_sms_SLQSSmsSetRoutes_t . . . 724
pack_swidms_SLQSSwiDmsSetMTU_t . . . 724
pack_swidms_SLQSSwiDmsSetUsbNetNum_t . . . 725
pack_swiloc_SwiLocSetAutoStart_t . . . 725
pack_swioma_SLQSOMADMCancelSession_t . . . 727
pack_swioma_SLQSOMADMGetSessionInfo_t . . . 727
pack_swioma_SLQSOMADMSendSelection_t . . . 728
pack_swioma_SLQSOMADMSetSettings_t . . . 728
pack_swioma_SLQSOMADMStartSession_t . . . 729
pack_uim_ChangePin_t . . . 730
pack_uim_ReadTransparent_t . . . 731
pack_uim_SetPinProtection_t . . . 732
pack_uim_SLQSUIMAuthenticate_t . . . 733
pack_uim_SLQSUIMDepersonalization_t . . . 734
pack_uim_SLQSUIMEventRegister_t . . . 734
pack_uim_SLQSUIMGetConfiguration_t . . . 735
pack_uim_SLQSUIMGetFileAttributes_t . . . 735
pack_uim_SLQSUIMPowerDown_t . . . 736
pack_uim_SLQSUIMPowerUp_t . . . 736
pack_uim_SLQSUIMRefreshComplete_t . . . 737
pack_uim_SLQSUIMRefreshGetLastEvent_t . . . 737
pack_uim_SLQSUIMRefreshOK_t . . . 738
pack_uim_SLQSUIMRefreshRegister_t . . . 738
pack_uim_SLQSUIMSwitchSlot_t . . . 739
pack_uim_UnblockPin_t . . . 740
pack_uim_VerifyPin_t . . . 741
pack_voice_AnswerUSSD_t . . . 742
pack_voice_OriginateUSSD_t . . . 742
pack_voice_SLQSOriginateUSSD_t . . . 742
pack_voice_SLQSVoiceALSSelectLine_t . . . 743
pack_voice_SLQSVoiceALSSetLineSwitching_t . . . 744
pack_voice_SLQSVoiceAnswerCall_t . . . 744
pack_voice_SLQSVoiceBindSubscription_t . . . 744
pack_voice_SLQSVoiceBurstDTMF_t . . . 745
pack_voice_SLQSVoiceDialCall_t . . . 745
pack_voice_SLQSVoiceEndCall_t . . . 747
pack_voice_SLQSVoiceGetCallBarring_t . . . 747
pack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 748

pack_voice_SLQSVoiceGetCallInfo_t . . . 749
pack_voice_SLQSVoiceGetCallWaiting_t . . . 749
pack_voice_SLQSVoiceGetConfig_t . . . 750
pack_voice_SLQSVoiceIndicationRegister_t . . . 752
pack_voice_SLQSVoiceManageCalls_t . . . 752
pack_voice_SLQSVoiceOrigUSSDNoWait_t . . . 753
pack_voice_SLQSVoiceSendFlash_t . . . 754
pack_voice_SLQSVoiceSetCallBarringPassword_t . . . 754
pack_voice_SLQSVoiceSetConfig_t . . . 755
pack_voice_SLQSVoiceSetPreferredPrivacy_t . . . 757
pack_voice_SLQSVoiceSetSUPSService_t . . . 757
pack_voice_SLQSVoiceStartContDTMF_t . . . 759
pack_voice_SLQSVoiceStopContDTMF_t . . . 760
pack_wds_DHCPv4ClientLeaseChange_t . . . 760
pack_wds_GetDefaultProfile_t . . . 761
pack_wds_GetDefaultProfileNum_t . . . 761
pack_wds_GetDormancyState_t . . . 762
pack_wds_GetLastMobileIPError_t . . . 762
pack_wds_GetMobileIP_t . . . 762
pack_wds_GetMobileIPProfile_t . . . 762
pack_wds_GetPacketStatistics_t . . . 763
pack_wds_GetPacketStatus_t . . . 763
pack_wds_GetSessionDuration_t . . . 764
pack_wds_RMSetTransferStatistics_t . . . 764
pack_wds_SetAutoconnect_t . . . 765
pack_wds_SetDefaultProfile_t . . . 765
pack_wds_SetDefaultProfileNum_t . . . 767
pack_wds_SetMobileIP_t . . . 768
pack_wds_SetMobileIPParameters_t . . . 768
pack_wds_SetMobileIPProfile_t . . . 769
pack_wds_SLQSCreateProfile_t . . . 771
pack_wds_SLQSDeleteProfile_t . . . 772
pack_wds_SLQSGetCurrDataSystemStat_t . . . 772
pack_wds_SLQSGetDataBearerTechnology_t . . . 773
pack_wds_SLQSGetDUNCallInfo_t . . . 773
pack_wds_SLQSGetProfileSettings_t . . . 774
pack_wds_SLQSGetRuntimeSettings_t . . . 775
pack_wds_SLQSModifyProfile_t . . . 776
pack_wds_SLQSSet3GPPConfigItem_t . . . 777
pack_wds_SLQSSetIPFamilyPreference_t . . . 779
pack_wds_SLQSSetWdsEventCallback_t . . . 779
pack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 780
pack_wds_SLQSSSetDHCPv4ClientConfig_t . . . 780
pack_wds_SLQSSSetLoopback_t . . . 781
pack_wds_SLQSStartDataSession_t . . . 781
pack_wds_SLQSStopDataSession_t . . . 783
pack_wds_SLQSWdsSetEventReport_t . . . 783
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 785
PackCreateProfileOut . . . 785
packgetDyingGaspCfg . . . 786
packgetDyingGaspStatistics . . . 786
PCMparams . . . 787
PCSCFFQDNAddress . . . 788
PCSCFFQDNAddressList . . . 788
PCSCFIPv4ServerAddressList . . . 789
PDSPositionData . . . 789
PDSPosMethodStateReq . . . 791
peerNumberInfo . . . 792

personalizationStatus . . . 794
PhyCaAggPcellInfo . . . 795
PhyCaAggScellDlBw . . . 795
PhyCaAggScellIndex . . . 796
PhyCaAggScellIndType . . . 796
PhyCaAggScellInfo . . . 797
PilotSetData . . . 800
PilotSetParams . . . 801
pktErrRate . . . 801
PlmnID . . . 802
PLMNNetworkName . . . 802
PLMNNetworkNameData . . . 803
Port . . . 805
PowerSaveModeConfigReq . . . 805
PowerSaveModeConfigResp . . . 806
precisionDilution_s . . . 807
PrefImageList . . . 808
prefVoiceSO . . . 808
Profile3GPP . . . 810
Profile3GPP2 . . . 816
ProfileIdentifier . . . 821
protocolSubtypeElement . . . 822
PSDetachReq . . . 824
PSMActiveTimerTlv . . . 824
PSMDurationDueToOOSTlv . . . 825
PSMDurationThresholdTlv . . . 825
PSMEarlyWakeupTimeTlv . . . 826
PSMEnableStateTlv . . . 826
PSMPeriodicUpdateTimerTlv . . . 827
PSMRandomizationWindowTlv . . . 827
qaQmi3Gpp2TimeZone . . . 828
qaQmiInterfaceInfo . . . 828
qaQmiServingSystemParam . . . 829
QmiCbkCatEventStatusReportInd . . . 833
QmiCbkLocBestAvailPosInd . . . 833
QmiCbkLocCradleMountInd . . . 839
QmiCbkLocEngineStateInd . . . 839
QmiCbkLocEventTimeSyncInd . . . 840
QmiCbkLocGetServerInd . . . 840
QmiCbkLocInjectPositionInd . . . 842
QmiCbkLocInjectSensorDataInd . . . 842
QmiCbkLocInjectTimeInd . . . 844
QmiCbkLocInjectUTCTimeInd . . . 844
QmiCbkLocPositionReportInd . . . 845
QmiCbkLocSensorStreamingInd . . . 851
QmiCbkLocSetExtPowerConfigInd . . . 851
QmiCbkLocSetServerInd . . . 852
QmiCbkNasLTECphyCaInfo . . . 853
QmiCbkSwiOmaDmEventStatusReportInd . . . 853
QmiCbkSwiOmaDmEventStatusReportIndExt . . . 854
QmiCbkTmdMitiLvlRptInd . . . 854
QmiCbkWdsStatisticsIndState . . . 854
qmifwinfo_s . . . 855
QmiNas3GppNetworkInfo . . . 856
QmiNasGetRFBandInfoResp . . . 857
QmiNasPerformNetworkScanResp . . . 858
qmiSmsMessageList . . . 858
qmiWDSDataBearerTechnology . . . 859

QmiWdsIpAddressInfo . . . 860
qmiWdsRunTimeSettings . . . 861
qos_BindDataPortMuxID_t . . . 865
qos_BindDataPortPeripheralEndPointID_t . . . 865
qos_BindDataPortSIODataPort_t . . . 866
QosClassID . . . 866
QosEventInfo . . . 867
QosFlowInfo . . . 869
QosFlowInfoState . . . 870
QosMap . . . 870
RankIndicatorInd . . . 871
readResult . . . 871
readTransparentInfo . . . 872
redirNumInfo . . . 872
registerRefresh . . . 874
remainingRetries . . . 875
remotePartyName . . . 875
remotePartyNum . . . 876
ReqFieldsList . . . 877
RespFieldsList . . . 878
RFBandInfoElements . . . 878
RfBandInfoExtFormat . . . 879
RfBandInfoExtFormatElements . . . 880
RfBandInfoList . . . 881
RfBandwidthInfo . . . 881
RfBandwidthInfoElements . . . 882
RfDedicatedBandInfo . . . 882
RfDedicatedBandInfoElements . . . 883
rmTrasnferStaticsReq . . . 884
roamIndList . . . 884
RoamingInfo . . . 885
roamTimer . . . 885
RSRPThresh . . . 886
rsrqInformation . . . 887
RSRQThresh . . . 887
RSSIThresh . . . 888
RXAGCList . . . 889
RXAVCList . . . 890
rxInfo . . . 890
RXPCMIIRFltr . . . 892
RxSigInfo . . . 893
rxSignalStrengthListElement . . . 894
sApnExtraParams . . . 895
satelliteInfo . . . 896
SccRxInfo . . . 899
sensorData . . . 899
sensorData_t . . . 901
sensorDataUsage_s . . . 903
serialNumbersInfo . . . 903
serviceProviderName . . . 905
ServingSystemInfo . . . 905
servSystem . . . 907
sessionInfo . . . 908
sessionInfoExt . . . 909
sessionInfoTlv . . . 909
sessionInfoTlvExt . . . 910
SetAudioPathConfigReq . . . 910
SetAudioProfileReq . . . 912

SetAudioVolTLBConfigReq . . . 913
SetAudioVolTLBConfigResp . . . 914
setCustomSettingV2 . . . 915
setDyingGaspCfg . . . 915
SetIMSSMSConfigReq . . . 916
SetIMSSMSConfigResp . . . 916
SetIMSUserConfigReq . . . 917
SetIMSUserConfigResp . . . 917
SetIMSVoIPConfigReq . . . 918
SetIMSVoIPConfigResp . . . 920
SetM2MAudioAVCFGReq . . . 921
SetM2MAudioLPBKReq . . . 922
SetM2MAudioProfileReq . . . 922
SetM2MAudioVolumeReq . . . 923
SetM2MAVMuteReq . . . 924
SetM2MSpkrGainReq . . . 925
setPINProtection . . . 925
SetRegMgrConfigReq . . . 926
SetRegMgrConfigResp . . . 927
setSignalStrengthInfo . . . 927
SetSIPConfigReq . . . 932
SetSIPConfigResp . . . 933
sGetDeviceSeriesResult . . . 934
sidNid . . . 934
sigInfo . . . 935
signalInfo . . . 936
SignalStrengthDataType . . . 937
slot_t . . . 937
slotInf . . . 938
slotInfo . . . 940
slots_t . . . 941
slqsautoconnect . . . 942
SLQSDeleteProfileParams . . . 942
slqsfwinfo_s . . . 943
SlqsNas3GppNetworkInfo . . . 944
SlqsNasPcsDigit . . . 946
slqssendasyncsmsparams_s . . . 946
slqssendsmsparams_s . . . 948
slqsSessionStateInfo . . . 950
slqsSignalStrengthInfo . . . 950
SLQSSignalStrengthsIndReq . . . 953
SLQSSignalStrengthsInformation . . . 955
slqsWdsEventInfo . . . 956
sms_BroadcastConfig . . . 958
sms_CDMABroadcastConfig . . . 958
sms_getIndicationReg . . . 959
sms_getMsgWaitingInfo . . . 960
sms_getTransLayerInfo . . . 960
sms_getTransNWRegInfo . . . 961
sms_maxStorageSizeReq . . . 962
sms_maxStorageSizeResp . . . 962
sms_messageWaitingInfoContent . . . 963
sms_msgProtocolResp . . . 964
sms_qaQmi3GPP2BroadcastCfgInfo . . . 964
sms_qaQmi3GPPBroadcastCfgInfo . . . 965
sms_routeEntry . . . 965
sms_sendAsyncsmsParams . . . 966
sms_setIndicationReg . . . 969

sms_setRoutesReq . . . 969
sms_transLayerInfo . . . 970
SMSAsyncRawSend_s . . . 971
SMSCAddress . . . 972
sMSCAddressInfo . . . 973
sMSCAddressTlv . . . 973
SMSEtwsMessage . . . 974
sMSEtwsMessageInfo . . . 975
sMSEtwsMessageTlv . . . 975
SMSEtwsPlmn . . . 976
sMSEtwsPlmnInfo . . . 976
SMSEventInfo_s . . . 977
smsMaxStorageSizeReq . . . 978
smsMaxStorageSizeResp . . . 979
SMSMemoryInfo . . . 979
SMSMessageMode . . . 980
sMSMessageModeInfo . . . 980
smsMsgprotocolResp . . . 980
SMSMTMessage . . . 981
sMSMTMessageInfo . . . 981
SMSOnIMS . . . 982
sMSOnIMSInfo . . . 982
sMSOnIMSTlv . . . 983
smsRouteEntry . . . 983
smsSetRoutesReq . . . 984
SMSTransferRouteMTMessage . . . 985
sMSTransferRouteMTMessageInfo . . . 986
sQosFlowStat . . . 987
sQosStat . . . 987
SrvStatusInfo . . . 989
ssdatasession_params . . . 989
SupportedMsgList . . . 992
SUPSInfo . . . 993
SV . . . 993
SVInfo . . . 994
svUsedforFix_s . . . 995
SWI_STRUCT_CarrierImage . . . 995
swidms_ehrpdMTUSizeTlv . . . 997
swidms_hrpdMTUSizeTlv . . . 997
swidms_mtuSize3gppTlv . . . 998
swidms_usbMTUSizeTlv . . . 998
SwiDmsGetMTUResp . . . 999
SwiDmsSLQSGetUsbNetNum . . . 999
SwiLocGetAutoStartResp . . . 1000
SwiLocSetAutoStartReq . . . 1001
swiModemStatusResp . . . 1003
SwiOTAMsg_s . . . 1004
swiPDPRuntimeSettingsReq . . . 1005
swiPDPRuntimeSettingsResp . . . 1005
swiQosFilter . . . 1007
swiQosFlow . . . 1010
swiQosGranted . . . 1013
swiQosIds . . . 1013
swiQosModifyReq . . . 1014
swiQosReq . . . 1015
swiRMTrasnferStaticsReq . . . 1015
sysInfoCommon . . . 1016
sysSelectPrefParamsExt . . . 1019

t_gpsTime . . . 1022
t_sensor . . . 1022
t_Sv . . . 1023
TDSCDMAECIOThresh . . . 1023
TDSCDMARSCPThresh . . . 1024
TDSCDMARSSIThresh . . . 1024
TDSCDMASigInfoExt . . . 1025
tdscdmaSigInfoExt . . . 1025
TDSCDMASINRCONFThresh . . . 1026
TDSCDMASINRThresh . . . 1027
tempData_t . . . 1027
tempratureData . . . 1028
TFTIDParams . . . 1029
timeInfo . . . 1031
TmdDeRegNotMitigationLvlReq . . . 1033
TmdGetMitigationDevListResp . . . 1033
TmdGetMitigationLvlReq . . . 1034
TmdGetMitigationLvlResp . . . 1035
TmdMitigationLvlIndReq . . . 1035
TmdRegNotMitigationLvlReq . . . 1036
tokenBucket . . . 1036
Tos . . . 1037
transferRouteMessageTlv . . . 1037
TransferStatInd . . . 1038
TransferStatsDataType . . . 1038
TrStatInd . . . 1039
trueIMSI . . . 1039
TXAGCList . . . 1040
txInfo . . . 1041
TXPCMIIRFltr . . . 1042
uim_appStatus . . . 1043
uim_authenticateResult . . . 1046
uim_authenticationData . . . 1047
uim_cardResult . . . 1048
uim_cardStatus . . . 1049
uim_changeUIMPIN . . . 1050
uim_depersonalizationInformation . . . 1051
uim_encryptedPIN1 . . . 1052
uim_fileAttributes . . . 1053
uim_fileInfo . . . 1056
uim_hotSwapStatus . . . 1057
uim_personalizationStatus . . . 1057
uim_readResult . . . 1058
uim_readTransparentInfo . . . 1059
uim_refreshevent . . . 1059
uim_registerRefresh . . . 1061
uim_remainingRetries . . . 1062
uim_sessionInformation . . . 1063
uim_setPINProtection . . . 1063
uim_slotInfo . . . 1064
uim_UIMSessionInformation . . . 1066
uim_unblockUIMPIN . . . 1067
uim_verifyUIMPIN . . . 1068
UIMAuthenticateReq . . . 1068
UIMAuthenticateResp . . . 1069
UIMChangePinReq . . . 1070
UIMDepersonalizationReq . . . 1071
UIMDepersonalizationResp . . . 1071

UIMEventRegisterReqResp . . . 1071
UIMGetCardStatusResp . . . 1072
UIMGetConfigurationReq . . . 1072
UIMGetConfigurationResp . . . 1073
UIMGetFileAttributesReq . . . 1074
UIMGetFileAttributesResp . . . 1074
UIMGetSlotsStatusResp . . . 1075
UIMPinResp . . . 1076
UIMPowerDownReq . . . 1076
UIMPowerUpReq . . . 1077
UIMReadTransparentReq . . . 1077
UIMReadTransparentResp . . . 1078
UIMRefreshCompleteReq . . . 1079
UIMRefreshEvent . . . 1080
UIMRefreshGetLastEventReq . . . 1081
UIMRefreshGetLastEventResp . . . 1082
UIMRefreshOKReq . . . 1082
UIMRefreshRegisterReq . . . 1083
UIMSessionInformation . . . 1083
UIMSetPinProtectionReq . . . 1084
UIMSlotsStatus . . . 1085
UIMSlotStatus . . . 1085
UIMSlotStatusChangeInfo . . . 1087
UIMStatusChangeInfo . . . 1087
UIMSwitchSlotReq . . . 1087
UIMUnblockPinReq . . . 1088
UIMVerifyPinReq . . . 1089
UMTSInfo . . . 1090
UMTSinstInfo . . . 1092
umtsLTENbrCell . . . 1092
UMTSMinQoS . . . 1093
UMTSQoS . . . 1096
UMTSReqQoSSigInd . . . 1098
unblockUIMPIN . . . 1099
UniversalTime . . . 1100
unpack_dms_GetActivationState_t . . . 1101
unpack_dms_GetBandCapability_t . . . 1102
unpack_dms_GetCrashAction_t . . . 1104
unpack_dms_GetCustFeature_t . . . 1104
unpack_dms_GetCustFeaturesV2_t . . . 1107
unpack_dms_GetDeviceCap_t . . . 1107
unpack_dms_GetDeviceCapabilities_t . . . 1108
unpack_dms_GetDeviceHardwareRev_t . . . 1110
unpack_dms_GetDeviceMfr_t . . . 1110
unpack_dms_GetDeviceSerialNumbers_t . . . 1111
unpack_dms_GetFirmwareInfo_t . . . 1112
unpack_dms_GetFirmwareRevision_t . . . 1114
unpack_dms_GetFirmwareRevisions_t . . . 1115
unpack_dms_GetFSN_t . . . 1116
unpack_dms_GetHardwareRevision_t . . . 1117
unpack_dms_GetIMSI_t . . . 1117
unpack_dms_GetManufacturer_t . . . 1118
unpack_dms_GetModelID_t . . . 1118
unpack_dms_GetNetworkTime_t . . . 1119
unpack_dms_GetOfflineReason_t . . . 1119
unpack_dms_GetPower_t . . . 1120
unpack_dms_GetPRLVersion_t . . . 1122
unpack_dms_GetSerialNumbers_t . . . 1122

unpack_dms_GetUSBComp_t . . . 1123
unpack_dms_GetVoiceNumber_t . . . 1125
unpack_dms_PSMCfgChange_ind_t . . . 1126
unpack_dms_ResetToFactoryDefaults_t . . . 1127
unpack_dms_SetActivationStatusCallback_t . . . 1127
unpack_dms_SetCrashAction_t . . . 1128
unpack_dms_SetCustFeature_t . . . 1128
unpack_dms_SetCustFeaturesV2_t . . . 1129
unpack_dms_SetEventReport_ind_t . . . 1129
unpack_dms_SetEventReport_t . . . 1130
unpack_dms_SetFirmwarePreference_t . . . 1130
unpack_dms_SetIndicationRegister_t . . . 1130
unpack_dms_SetPower_t . . . 1131
unpack_dms_SetUSBComp_t . . . 1131
unpack_dms_SLQSDmsSwiGetPCInfo_t . . . 1131
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t . . . 1133
unpack_dms_SLQSDmsSwiGetResetInfo_t . . . 1134
unpack_dms_SLQSDmsSwiIndicationRegister_t . . . 1135
unpack_dms_SLQSGetBandCapability_t . . . 1136
unpack_dms_SLQSGetERIFile_t . . . 1139
unpack_dms_SLQSGetPowerSaveModeConfig_t . . . 1139
unpack_dms_SLQSSetPowerSaveModeConfig_t . . . 1141
unpack_dms_SLQSSwiClearDyingGaspStatistics_t . . . 1141
unpack_dms_SLQSSwiGetCrashInfo_t . . . 1141
unpack_dms_SLQSSwiGetDyingGaspCfg_t . . . 1142
unpack_dms_SLQSSwiGetDyingGaspStatistics_t . . . 1142
unpack_dms_SLQSSwiGetFirmwareCurr_t . . . 1143
unpack_dms_SLQSSwiGetFwUpdateStatus_t . . . 1144
unpack_dms_SLQSSwiGetHostDevInfo_t . . . 1146
unpack_dms_SLQSSwiGetOSInfo_t . . . 1147
unpack_dms_SLQSSwiGetSerialNoExt_t . . . 1147
unpack_dms_SLQSSwiSetDyingGaspCfg_t . . . 1148
unpack_dms_SLQSSwiSetHostDevInfo_t . . . 1148
unpack_dms_SLQSSwiSetOSInfo_t . . . 1149
unpack_dms_SLQSUIMGetState_t . . . 1149
unpack_dms_SwiEventReportCallBack_ind_t . . . 1149
unpack_dms_SwiSetEventReport_t . . . 1150
unpack_dms_UIMGetControlKeyStatus_t . . . 1150
unpack_dms_UIMGetICCID_t . . . 1151
unpack_dms_UIMGetPINStatus_t . . . 1152
unpack_dms_UIMSetControlKeyProtection_t . . . 1154
unpack_dms_UIMSetPINProtection_t . . . 1154
unpack_dms_UIMUnblockControlKey_t . . . 1155
unpack_fms_GetImagesPreference_t . . . 1155
unpack_fms_GetStoredImages_t . . . 1156
unpack_fms_SetImagesPreference_t . . . 1157
unpack_loc_BestAvailPos_Ind_t . . . 1157
unpack_loc_CradleMountCallback_Ind_t . . . 1163
unpack_loc_Delete_Assist_Data_t . . . 1163
unpack_loc_DeleteAssistData_Ind_t . . . 1164
unpack_loc_EngineState_Ind_t . . . 1165
unpack_loc_EventNMEA_Ind_t . . . 1165
unpack_loc_EventRegister_t . . . 1166
unpack_loc_EventTimeSyncCallback_Ind_t . . . 1166
unpack_loc_GetServer_Ind_t . . . 1166
unpack_loc_GnssSvInfo_Ind_t . . . 1168
unpack_loc_InjectPositionCallback_Ind_t . . . 1168
unpack_loc_InjectSensorDataCallback_Ind_t . . . 1169

unpack_loc_InjectTimeSyncDataCallback_Ind_t . . . 1171
unpack_loc_InjectUTCTimeCallback_Ind_t . . . 1171
unpack_loc_PositionRpt_Ind_t . . . 1172
unpack_loc_SensorStreamingCallback_Ind_t . . . 1179
unpack_loc_SetExtPowerConfig_Ind_t . . . 1179
unpack_loc_SetExtPowerState_t . . . 1180
unpack_loc_SetOperationMode_Ind_t . . . 1181
unpack_loc_SetOperationMode_t . . . 1181
unpack_loc_SetServer_Ind_t . . . 1182
unpack_loc_SLQSLOCGetBestAvailPos_t . . . 1183
unpack_loc_Start_t . . . 1183
unpack_loc_Stop_t . . . 1183
unpack_nas_GetCDMANetworkParameters_t . . . 1184
unpack_nas_GetHomeNetwork3GPP2_t . . . 1186
unpack_nas_GetHomeNetwork_t . . . 1188
unpack_nas_GetNetworkPreference_t . . . 1189
unpack_nas_GetRFInfo_t . . . 1190
unpack_nas_GetServingNetwork_t . . . 1191
unpack_nas_GetServingNetworkCapabilities_t . . . 1193
unpack_nas_GetSignalStrengths_t . . . 1194
unpack_nas_PerformNetworkScan_t . . . 1195
unpack_nas_SetDataCapabilitiesCallback_ind_t . . . 1196
unpack_nas_SetEventReportInd_t . . . 1196
unpack_nas_SetNasLTECphyCaIndCallback_ind_t . . . 1197
unpack_nas_SetNetworkPreference_t . . . 1198
unpack_nas_SetRoamingIndicatorCallback_ind_t . . . 1198
unpack_nas_SetServingSystemCallback_ind_t . . . 1199
unpack_nas_SLQSGetErrorRate_t . . . 1199
unpack_nas_SlqsGetLTECphyCAInfo_t . . . 1200
unpack_nas_SLQSGetNetworkTime_t . . . 1201
unpack_nas_SLQSGetOperatorNameData_t . . . 1201
unpack_nas_SLQSGetPLMNName_t . . . 1202
unpack_nas_SLQSGetServingSystem_t . . . 1205
unpack_nas_SLQSGetSignalStrength_t . . . 1209
unpack_nas_SLQSGetSysInfo_t . . . 1211
unpack_nas_SLQSGetSysSelectionPref_t . . . 1215
unpack_nas_SLQSNasGet3GPP2Subscription_t . . . 1219
unpack_nas_SLQSNasGetCellLocationInfo_t . . . 1220
unpack_nas_SLQSNASGeteDRXParams_t . . . 1222
unpack_nas_SLQSNasGetHDRColorCode_t . . . 1223
unpack_nas_SLQSNasGetRFInfo_t . . . 1223
unpack_nas_SLQSNasGetSigInfo_t . . . 1224
unpack_nas_SLQSNasGetTxRxInfo_t . . . 1225
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t . . . 1225
unpack_nas_SLQSNasSigInfoCallback_ind_t . . . 1226
unpack_nas_SLQSNASSwiGetChannelLock_t . . . 1228
unpack_nas_SLQSNasSwiModemStatus_t . . . 1228
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t . . . 1229
unpack_nas_SLQSNasTimerCallback_ind_t . . . 1229
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t . . . 1230
unpack_nas_SLQSSwiGetHDRPersonality_t . . . 1230
unpack_nas_SLQSSwiGetHDRProtSubtype_t . . . 1231
unpack_nas_SLQSSwiGetHRPDStats_t . . . 1232
unpack_nas_SLQSSwiGetLteCQI_t . . . 1233
unpack_nas_SLQSSwiGetLteSccRxInfo_t . . . 1234
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t . . . 1234
unpack_nas_SLQSSwiNetworkDebug_t . . . 1234
unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t . . . 1235

unpack_nas_SLQSSysInfoCallback_ind_t . . . 1236
unpack_omaDmConfigTlv_t . . . 1239
unpack_omaDmFotaTlv_t . . . 1240
unpack_omaDmNotificationsTlv_t . . . 1242
unpack_pds_GetPDSDefaults_t . . . 1243
unpack_pds_GetPDSState_t . . . 1243
unpack_pds_GetPortAutomaticTracking_t . . . 1244
unpack_pds_GetServiceAutomaticTracking_t . . . 1245
unpack_pds_GetXTRAAutomaticDownload_t . . . 1245
unpack_pds_GetXTRANetwork_t . . . 1246
unpack_pds_GetXTRAValidity_t . . . 1246
unpack_pds_SetEventReport_Ind_t . . . 1247
unpack_pds_SetPdsState_Ind_t . . . 1249
unpack_pds_SLQSGetAGPSConfig_t . . . 1249
unpack_pds_SLQSGetGPSStateInfo_t . . . 1250
unpack_qmi_t . . . 1254
unpack_qos_dataRate_t . . . 1254
unpack_qos_IPv4Addr_t . . . 1255
unpack_qos_IPv6Addr_t . . . 1255
unpack_qos_IPv6TrafCls_t . . . 1256
unpack_qos_pktErrRate_t . . . 1256
unpack_qos_Port_t . . . 1257
unpack_qos_QosFlowInfo_t . . . 1257
unpack_qos_QosFlowInfoState_t . . . 1259
unpack_qos_SLQSQosGetNetworkStatus_t . . . 1259
unpack_qos_SLQSQosSwiReadApnExtraParams_t . . . 1260
unpack_qos_SLQSQosSwiReadDataStats_t . . . 1261
unpack_qos_SLQSSetQosEventCallback_ind_t . . . 1262
unpack_qos_SLQSSetQosNWStatusCallback_ind_t . . . 1263
unpack_qos_SLQSSetQosPriEventCallback_ind_t . . . 1263
unpack_qos_SLQSSetQosStatusCallback_ind_t . . . 1264
unpack_qos_swiQosFilter_t . . . 1265
unpack_qos_swiQosFlow_t . . . 1269
unpack_qos_tokenBucket_t . . . 1273
unpack_qos_Tos_t . . . 1273
unpack_QosFlowStat_t . . . 1274
unpack_rms_GetSMSWake_t . . . 1274
unpack_RMTransferStatistics_ind_t . . . 1275
unpack_sar_SLQSGetRfSarState_t . . . 1276
unpack_sms_GetSMSCAddress_t . . . 1277
unpack_sms_SaveSMS_t . . . 1277
unpack_sms_SendSMS_t . . . 1278
unpack_sms_SetNewSMSCallback_ind_t . . . 1278
unpack_sms_SetNewSMSCallback_t . . . 1279
unpack_sms_SLQSDeleteSMS_t . . . 1280
unpack_sms_SLQSGetIndicationRegister_t . . . 1280
unpack_sms_SLQSGetMessageWaiting_t . . . 1280
unpack_sms_SLQSGetSMS_t . . . 1281
unpack_sms_SLQSGetSmsBroadcastConfig_t . . . 1282
unpack_sms_SLQSGetSMSList_t . . . 1282
unpack_sms_SLQSGetTransLayerInfo_t . . . 1283
unpack_sms_SLQSGetTransNWRegInfo_t . . . 1283
unpack_sms_SLQSModifySMSStatus_t . . . 1283
unpack_sms_SLQSNWRegInfoCallback_ind_t . . . 1284
unpack_sms_SLQSSmsGetMaxStorageSize_t . . . 1284
unpack_sms_SLQSSmsGetMessageProtocol_t . . . 1285
unpack_sms_SLQSSwiGetSMSStorage_t . . . 1285
unpack_sms_SLQSTransLayerInfoCallback_ind_t . . . 1286

unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t . . . 1286
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t . . . 1288
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t . . . 1289
unpack_swidms_SLQSSwiDmsGetMTU_t . . . 1289
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t . . . 1290
unpack_swidms_SLQSSwiDmsSetMTU_t . . . 1290
unpack_swiloc_SwiLocGetAutoStart_t . . . 1291
unpack_swioma_SLQSOMADMAlertCallback_ind_t . . . 1292
unpack_swioma_SLQSOMADMGetSessionInfo_t . . . 1293
unpack_swioma_SLQSOMADMGetSettings_t . . . 1296
unpack_swioma_SLQSOMADMStartSession_t . . . 1297
unpack_uim_ChangePin_t . . . 1298
unpack_uim_GetCardStatus_t . . . 1299
unpack_uim_ReadTransparent_t . . . 1299
unpack_uim_SetPinProtection_t . . . 1300
unpack_uim_SetUimSlotStatusChangeCallback_ind_t . . . 1301
unpack_uim_SLQSUIMAuthenticate_t . . . 1302
unpack_uim_SLQSUIMDepersonalization_t . . . 1302
unpack_uim_SLQSUIMEventRegister_t . . . 1303
unpack_uim_SLQSUIMGetConfiguration_t . . . 1303
unpack_uim_SLQSUIMGetFileAttributes_t . . . 1304
unpack_uim_SLQSUIMGetSlotsStatus_t . . . 1305
unpack_uim_SLQSUIMRefreshCallback_Ind_t . . . 1305
unpack_uim_SLQSUIMRefreshGetLastEvent_t . . . 1306
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t . . . 1306
unpack_uim_UnblockPin_t . . . 1307
unpack_uim_VerifyPin_t . . . 1307
unpack_voice_allCallStatusCallback_ind_t . . . 1308
unpack_voice_DTMFEventCallback_ind_t . . . 1311
unpack_voice_OTASPStatusCallback_ind_t . . . 1312
unpack_voice_SLQSOriginateUSSD_t . . . 1313
unpack_voice_SLQSVoiceAnswerCall_t . . . 1316
unpack_voice_SLQSVoiceBurstDTMF_t . . . 1317
unpack_voice_SLQSVoiceDialCall_t . . . 1317
unpack_voice_SLQSVoiceEndCall_t . . . 1318
unpack_voice_SLQSVoiceGetAllCallInfo_t . . . 1319
unpack_voice_SLQSVoiceGetCallBarring_t . . . 1321
unpack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 1323
unpack_voice_SLQSVoiceGetCallInfo_t . . . 1324
unpack_voice_SLQSVoiceGetCallWaiting_t . . . 1328
unpack_voice_SLQSVoiceGetCLIP_t . . . 1329
unpack_voice_SLQSVoiceGetCLIR_t . . . 1330
unpack_voice_SLQSVoiceGetCNAP_t . . . 1332
unpack_voice_SLQSVoiceGetCOLP_t . . . 1333
unpack_voice_SLQSVoiceGetCOLR_t . . . 1334
unpack_voice_SLQSVoiceGetConfig_t . . . 1336
unpack_voice_SLQSVoiceManageCalls_t . . . 1338
unpack_voice_SLQSVoiceSendFlash_t . . . 1338
unpack_voice_SLQSVoiceSetCallBarringPassword_t . . . 1339
unpack_voice_SLQSVoiceSetConfig_t . . . 1340
unpack_voice_SLQSVoiceSetSUPSService_t . . . 1342
unpack_voice_SLQSVoiceStartContDTMF_t . . . 1343
unpack_voice_SLQSVoiceStopContDTMF_t . . . 1343
unpack_voice_SLQSVoiceSUPSCallback_ind_t . . . 1344
unpack_voice_SUPSNotificationCallback_ind_t . . . 1346
unpack_voice_USSDNotificationCallback_ind_t . . . 1348
unpack_voice_VoiceInfoRecCallback_ind_t . . . 1348
unpack_voice_voicePrivacyChangeCallback_ind_t . . . 1351

unpack_wds_DHCPv4ClientLease_ind_t . . . 1351
unpack_wds_GetAutoconnect_t . . . 1352
unpack_wds_GetByteTotals_t . . . 1352
unpack_wds_GetConnectionRate_t . . . 1353
unpack_wds_GetDataBearerTechnology_t . . . 1354
unpack_wds_GetDefaultProfile_t . . . 1354
unpack_wds_GetDefaultProfileNum_t . . . 1357
unpack_wds_GetDormancyState_t . . . 1357
unpack_wds_GetLastMobileIPError_t . . . 1358
unpack_wds_GetMobileIP_t . . . 1358
unpack_wds_GetMobileIPProfile_t . . . 1359
unpack_wds_GetPacketStatistics_t . . . 1360
unpack_wds_GetPacketStatus_t . . . 1362
unpack_wds_GetSessionDuration_t . . . 1364
unpack_wds_GetSessionState_t . . . 1365
unpack_wds_RMSetTransferStatistics_t . . . 1365
unpack_wds_SetMobileIPProfile_t . . . 1365
unpack_wds_SLQSCreateProfile_t . . . 1366
unpack_wds_SLQSDeleteProfile_t . . . 1366
unpack_wds_SLQSDUNCallInfoCallBack_ind_t . . . 1367
unpack_wds_SLQSGet3GPPConfigItem_t . . . 1368
unpack_wds_SLQSGetCurrDataSystemStat_t . . . 1370
unpack_wds_SLQSGetCurrentChannelRate_t . . . 1370
unpack_wds_SLQSGetDataBearerTechnology_t . . . 1371
unpack_wds_SLQSGetDUNCallInfo_t . . . 1372
unpack_wds_SLQSGetProfileSettings_t . . . 1375
unpack_wds_SLQSGetRuntimeSettings_t . . . 1375
unpack_wds_SLQSModifyProfile_t . . . 1378
unpack_wds_SLQSSetIPFamilyPreference_t . . . 1379
unpack_wds_SLQSSetPacketSrvStatusCallback_t . . . 1379
unpack_wds_SLQSSetWdsEventCallback_ind_t . . . 1381
unpack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 1383
unpack_wds_SLQSSGetLoopback_t . . . 1383
unpack_wds_SLQSStartDataSession_t . . . 1384
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 1385
UnPackGetProfileSettingOut . . . 1387
unpackWdsProfileParam . . . 1388
urlAddr . . . 1388
URLAddrInfo . . . 1389
USBCompConfig . . . 1389
USBCompParams . . . 1390
usbMTUSizeTlv . . . 1392
USSDNoWaitIndicationInfo . . . 1393
USSDRespFNetwork . . . 1393
USSInfo . . . 1394
USSResp . . . 1395
UUSInfo . . . 1395
verifyUIMPIN . . . 1396
voice_airTimer . . . 1397
voice_allCallsAlphaIDInfo . . . 1397
voice_allCallsDiagInfo . . . 1398
voice_allCallsUUSInfo . . . 1398
voice_alphaIDInfo . . . 1399
voice_arrAlertingPattern . . . 1400
voice_arrAlertingType . . . 1400
voice_arrAlphaID . . . 1401
voice_arrCalledPartyNum . . . 1402
voice_arrCallEndReason . . . 1402

voice_arrCallInfo . . . 1403
voice_arrConnectPartyNum . . . 1403
voice_arrDiagInfo . . . 1404
voice_arrRedirPartyNum . . . 1404
voice_arrRemotePartyName . . . 1405
voice_arrRemotePartyNum . . . 1406
voice_arrSvcOption . . . 1406
voice_arrUUSInfo . . . 1407
voice_burstDTMFInfo . . . 1407
voice_calledPartyInfo . . . 1408
voice_calledPartySubAdd . . . 1410
voice_callerIDInfo . . . 1411
voice_callFwdTypeAndPlan . . . 1411
voice_callFWExtInfo . . . 1412
voice_callFWInfo . . . 1414
voice_callInfo . . . 1415
voice_callingPartyInfo . . . 1417
voice_ccSUPSType . . . 1418
voice_CLIPResp . . . 1419
voice_CLIRResp . . . 1420
voice_CNAPResp . . . 1421
voice_COLPResp . . . 1421
voice_COLRResp . . . 1422
voice_connectNumInfo . . . 1423
voice_CUGInfo . . . 1424
voice_curAMRConfig . . . 1425
voice_diagInfo . . . 1426
voice_DTMFInfo . . . 1426
voice_DTMFLengths . . . 1427
voice_ECTNum . . . 1428
voice_extDispRecInfo . . . 1429
voice_getAllCallInformation . . . 1429
voice_getAllCallRmtPtyName . . . 1430
voice_getAllCallRmtPtyNum . . . 1431
voice_getCallFWExtInfo . . . 1431
voice_getCallFWInfo . . . 1432
voice_lineCtrlInfo . . . 1432
voice_newPwdData . . . 1433
voice_NSSAudioCtrl . . . 1434
voice_peerNumberInfo . . . 1434
voice_prefVoiceSO . . . 1436
voice_redirNumInfo . . . 1438
voice_remotePartyName . . . 1440
voice_remotePartyNum . . . 1440
voice_roamTimer . . . 1441
voice_signalInfo . . . 1442
voice_SUPSInfo . . . 1443
voice_USSDNotificationNetworkInfo . . . 1443
voice_USSInfo . . . 1444
voice_UUSInfo . . . 1444
voiceALSSelectLineInfo . . . 1446
voiceALSSetLineSwitchInfo . . . 1446
voiceAnswerCall . . . 1446
voiceBindSubscriptionInfo . . . 1447
voiceBurstDTMFInfo . . . 1447
voiceCallInfoReq . . . 1448
voiceCallInfoResp . . . 1448
voiceCallRequestParams . . . 1451

voiceCallResponseParams . . . 1452
voiceContDTMFinfo . . . 1453
voiceDTMFEventInfo . . . 1454
voiceFlashInfo . . . 1455
voiceGetAllCallInfo . . . 1455
voiceGetCallBarringReq . . . 1458
voiceGetCallBarringResp . . . 1459
voiceGetCallFWReq . . . 1460
voiceGetCallFWResp . . . 1461
voiceGetCallWaitInfo . . . 1462
voiceGetCLIPResp . . . 1464
voiceGetCLIRResp . . . 1465
voiceGetCNAPResp . . . 1466
voiceGetCOLPResp . . . 1467
voiceGetCOLRResp . . . 1469
voiceGetConfigReq . . . 1470
voiceGetConfigResp . . . 1472
voiceIndicationRegisterInfo . . . 1473
voiceInfoRec . . . 1474
voiceManageCallsReq . . . 1477
voiceManageCallsResp . . . 1477
voiceOrigUSSDNoWaitInfo . . . 1478
voiceOTASPStatusInfo . . . 1478
voicePrivacyInfo . . . 1479
voiceSetAllCallStatusCbkInfo . . . 1480
voiceSetCallBarringPwdInfo . . . 1482
voiceSetCallBarringPwdResp . . . 1483
voiceSetConfigReq . . . 1484
voiceSetConfigResp . . . 1486
voiceSetPrefPrivacy . . . 1487
voiceSetSUPSServiceReq . . . 1488
voiceSetSUPSServiceResp . . . 1490
voiceStopContDTMFinfo . . . 1491
voiceSUPSInfo . . . 1492
voiceSUPSNotification . . . 1494
wcdmaCellInfo . . . 1496
WCDMAECIOThresh . . . 1497
WCDMAInfoLTENeighborCell . . . 1497
wcdmaLongMsgDecodingParams . . . 1498
wcdmaMsgDecodingParams . . . 1500
wcdmaMsgEncodingParams . . . 1501
WCDMARSSIThresh . . . 1502
WCDMASysInfo . . . 1503
wcdmaUARFCN . . . 1506
wds_channelRate . . . 1507
wds_ChannelRateTlv . . . 1507
wds_ConnStatusTlv . . . 1508
wds_currNetworkInfo . . . 1508
wds_DataBearTechTlv . . . 1510
wds_DataULongLongTlv . . . 1511
wds_DataULongTlv . . . 1512
wds_DHCPLeaseOptTlv . . . 1512
wds_DHCPLeaseStateTlv . . . 1513
wds_DHCPOpt . . . 1513
wds_DHCPProfileIdTlv . . . 1514
wds_DHCPv4HWConfig . . . 1514
wds_DHCPv4Option . . . 1515
wds_DHCPv4OptionList . . . 1516

wds_DHCPv4ProfileId . . . 1516
wds_Domain . . . 1517
wds_DomainNameList . . . 1517
wds_DormStatTlv . . . 1518
wds_GPRSQoS . . . 1518
wds_IPv4AdTlv . . . 1519
wds_IPV6AddressInfo . . . 1520
wds_IPV6GWAddressInfo . . . 1520
wds_LastMdmCallEndRsnTlv . . . 1521
wds_PCSCFFQDNAddress . . . 1521
wds_PCSCFFQDNAddressList . . . 1522
wds_PCSCFIPv4ServerAddressList . . . 1522
wds_ProfileIdentifier . . . 1523
wds_profileInfo . . . 1523
wds_RXBytesOKTlv . . . 1524
wds_transferStatInd . . . 1524
wds_TrStatInd . . . 1525
wds_TXBytesOKTlv . . . 1526
wds_UMTSMinQoS . . . 1526
WdsByteTotals . . . 1528
WdsByteTotalsElmnts . . . 1529
WdsClientLeaseChange . . . 1530
WdsConnectionRate . . . 1530
WdsConnectionRateElmnts . . . 1531
WdsDHCPv4ClientLeaseInd . . . 1532
WdsDHCPv4Config . . . 1533
wdsDhcpv4HwConfig . . . 1533
WdsDHCPv4HWConfig . . . 1534
WdsDHCPv4Option . . . 1534
wdsDhcpv4Option . . . 1535
WdsDHCPv4OptionList . . . 1536
wdsDhcpv4OptionList . . . 1536
wdsDhcpv4ProfileId . . . 1537
WdsDHCPv4ProfileId . . . 1537
WDSGetLoopbackData . . . 1538
WdsIpAddressInfoReq . . . 1538
WdsPktStatisticsElmnts . . . 1539
WdsPktStatisticsReq . . . 1541
WdsPktStatisticsResp . . . 1541
WdsProfileParam . . . 1542
WdsRunTimeSettings . . . 1543
wdsSetEventReportReq . . . 1543
WDSSetLoopbackData . . . 1545
WDSSWICurrentChannelRates . . . 1546

# 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 . . . 1547
common.h . . . 1547
dms.h . . . 1553
fms.h . . . 1611
loc.h . . . 1614
nas.h . . . 1635
pds.h . . . 1688
qaCbkCatEventReportInd.h . . . 1709
qaCbkSwiOmaDmEventReportInd.h . . . 1710
qaGobiApiAudio.h
Audio Service API function prototypes . . . 1711
qaGobiApiCat.h
Card Application Toolkit API function headers . . . 1715
qaGobiApiCbk.h
Callback Service API function prototypes . . . 1716
qaGobiApiDcs.h
Device Connectivity Service API function prototypes . . . 1804
qaGobiApiDms.h
Device Management Service API function prototypes . . . 1813
qaGobiApiFms.h
Firmware Management Service API function prototypes . . . 1853
qaGobiApiIms.h
IMS Service API function prototypes . . . 1873
qaGobiApiImsa.h
IMSA Service API function prototypes . . . 1880
qaGobiApiLoc.h
Location API function prototypes . . . 1883
qaGobiApiNas.h
Network Access Service API function prototypes . . . 1890
qaGobiApiOmadm.h
Open Mobile Alliance Device Management Service API function prototypes . . . 1940
qaGobiApiPds.h
Position Determination Service API function prototypes . . . 1943
qaGobiApiQos.h
Quality of Service API function prototypes . . . 1958

qaGobiApiRms.h
Remote Management Service API function prototypes . . . 1965
qaGobiApiSar.h
Specific Absorption Rate API function prototypes . . . 1966
qaGobiApiSms.h
Short Message Service API function prototypes . . . 1969
qaGobiApiSwi.h
SWI API function prototypes . . . 1991
qaGobiApiSwiAudio.h
M2M Audio Service API function prototypes . . . 1992
qaGobiApiSwiAvms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI AVMS QMI Service revision 1.14 . . . 1998
qaGobiApiSwiDms.h
SWI DMS API function prototypes . . . 2010
qaGobiApiSwiOmadms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6 . . . 2012
qaGobiApiTableBandClasses.h
Network Access Service API Band Classes table . . . 2023
qaGobiApiTableCallControlReturnReasons.h
Call Control Return Reasons table . . . 2026
qaGobiApiTableCallEndReasons.h
Wireless Data Service Call End Reasons . . . 2027
qaGobiApiTableCarrierCodes.h
Carrier Codes table . . . 2043
qaGobiApiTableCodingScheme.h
Data Coding Scheme . . . 2045
qaGobiApiTableGpsCapabilityCodes.h
Position Determination Service API GPS Capability Codes . . . 2048
qaGobiApiTablePowerModes.h
Device Management Service API Power Modes table . . . 2049
qaGobiApiTableRadioInterfaces.h
Network Access Service API Radio Interfaces table . . . 2049
qaGobiApiTableRegionCodes.h
Region Codes table . . . 2050
qaGobiApiTableServiceOptions.h
Voice Service Options . . . 2050
qaGobiApiTableSupServiceInfoClasses.h
Voice Supplementary Service Information Classes . . . 2053
qaGobiApiTableSwiAudio.h
Swi Audio related tables . . . 2053
qaGobiApiTableSwiOMADMSessionStatus.h
Session Status table . . . 2054
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h
Update Complete Status table . . . 2055
qaGobiApiTableVoiceCallEndReasons.h
Voice Service Call and supplementary services end reasons . . . 2056
qaGobiApiTmd.h
Thermal Mitigation Device API function prototypes . . . 2063
qaGobiApiUim.h
Uim Service API function prototypes . . . 2065
qaGobiApiVoice.h
Voice Service API function prototypes . . . 2077
qaGobiApiWds.h
Wireless Data Service API function prototypes . . . 2098
qaNasGetRFBandInfo.h . . . 2142
qaNasPerformNetworkScan.h . . . 2142

qmerrno.h . . . . . 2143
qos.h . . . . . 2150
rms.h . . . . . 2160
sar.h . . . . . 2162
sms.h . . . . . 2163
SwiDataTypes.h
SWI data types . . . . . 2185
swidms.h . . . . . 2187
swiloc.h . . . . . 2190
swioma.h . . . . . 2192
SWIWWANCMAPI.h . . . . . 2201
uim.h . . . . . 2201
voice.h . . . . . 2219
wds.h . . . . . 2250

# Chapter 6

# Module Documentation

## 6.1 Device Connectivity Service (DCS)

### Files

- file qaGobiApiDcs.h

  *Device Connectivity Service API function prototypes.*

### 6.1.1 Detailed Description

## 6.2 Wireless Data Service (WDS)

### Files

- file qaGobiApiTableCallEndReasons.h

  *Wireless Data Service Call End Reasons.*
- file qaGobiApiTableCarrierCodes.h

  *Carrier Codes table.*
- file qaGobiApiTableRegionCodes.h

  *Region Codes table.*
- file qaGobiApiWds.h

  *Wireless Data Service API function prototypes.*

### 6.2.1 Detailed Description

## 6.3 Device Management Service (DMS)

### Files

- file qaGobiApiTablePowerModes.h

  *Device Management Service API Power Modes table.*
- file qaGobiApiDms.h

  *Device Management Service API function prototypes.*

### 6.3.1 Detailed Description

## 6.4 Network Access Service (NAS)

### Files

- file qaGobiApiTableRadioInterfaces.h

  *Network Access Service API Radio Interfaces table.*
- file qaGobiApiTableBandClasses.h

  *Network Access Service API Band Classes table.*
- file qaGobiApiNas.h

  *Network Access Service API function prototypes.*

### 6.4.1 Detailed Description

## 6.5 CallBacK registration (CBK)

### Files

- file qaGobiApiCbk.h

  *Callback Service API function prototypes.*

### 6.5.1 Detailed Description

## 6.6 Short Message Service (SMS)

### Files

- file qaGobiApiSms.h

  *Short Message Service API function prototypes.*

### 6.6.1 Detailed Description

## 6.7 Position Determination Service (PDS)

### Files

- file qaGobiApiTableGpsCapabilityCodes.h

  *Position Determination Service API GPS Capability Codes.*
- file qaGobiApiPds.h

  *Position Determination Service API function prototypes.*

### 6.7.1 Detailed Description

## 6.8 Card Application Toolkit (CAT)

### Files

- file qaGobiApiCat.h

  *Card Application Toolkit API function headers.*

### 6.8.1 Detailed Description

## 6.9 Remote Management Service (RMS)

### Files

- file qaGobiApiRms.h

  *Remote Management Service API function prototypes.*

### 6.9.1 Detailed Description

## 6.10 Firmware Management Service (FMS)

### Files

- file qaGobiApiFms.h

  *Firmware Management Service API function prototypes.*

### 6.10.1 Detailed Description

## 6.11 Open Mobile Alliance Service (OMA)

### Files

- file qaGobiApiOmadm.h

  *Open Mobile Alliance Device Management Service API function prototypes.*

### 6.11.1 Detailed Description

## 6.12 Specific Absorption Rate (SAR)

### Files

- file qaGobiApiSar.h

  *Specific Absorption Rate API function prototypes.*

### 6.12.1 Detailed Description

## 6.13 SWI Open Mobile Alliance Service (SWIOMA)

### Files

- file qaGobiApiTableSwiOMADMUpdateCompleteStatus.h

  *Update Complete Status table.*
- file qaGobiApiTableSwiOMADMSessionStatus.h

  *Session Status table.*
- file qaGobiApiSwiOmadms.h

  *SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6.*

### 6.13.1 Detailed Description

## 6.14 Voice Service (VOICE)

### Files

- file qaGobiApiTableServiceOptions.h

  *Voice Service Options.*
- file qaGobiApiTableVoiceCallEndReasons.h

  *Voice Service Call and supplementary services end reasons.*
- file qaGobiApiTableCodingScheme.h

  *Data Coding Scheme.*
- file qaGobiApiTableCallControlReturnReasons.h

  *Call Control Return Reasons table.*
- file qaGobiApiTableSupServiceInfoClasses.h

  *Voice Supplementary Service Information Classes.*
- file qaGobiApiVoice.h

  *Voice Service API function prototypes.*

### 6.14.1 Detailed Description

## 6.15 Non-service specific APIs (SWI)

### Files

- file qaGobiApiSwi.h

  *SWI API function prototypes.*

### 6.15.1 Detailed Description

## 6.16 User Identity Module Service (UIM)

### Files

- file qaGobiApiUim.h

  *Uim Service API function prototypes.*

### 6.16.1 Detailed Description

## 6.17 Audio Service (AUDIO)

### Files

- file qaGobiApiAudio.h

  *Audio Service API function prototypes.*

### 6.17.1 Detailed Description

## 6.18 Quality of Service (QOS)

### Files

- file qaGobiApiQos.h

  *Quality of Service API function prototypes.*

### 6.18.1 Detailed Description

## 6.19 IMS Service (IMS)

### Files

- file qaGobiApiIms.h

  *IMS Service API function prototypes.*
- file qaGobiApiImsa.h

  *IMSA Service API function prototypes.*

### 6.19.1 Detailed Description

## 6.20 SWI Audio Service(SWIAUDIO)

### Files

- file qaGobiApiTableSwiAudio.h

  *Swi Audio related tables.*
- file qaGobiApiSwiAudio.h

  *M2M Audio Service API function prototypes.*

### 6.20.1 Detailed Description

## 6.21 Location Service(LOC)

### Files

- file qaGobiApiLoc.h

  *Location API function prototypes.*

### 6.21.1 Detailed Description

## 6.22 Thermal Mitigation Device(TMD)

### Files

- file qaGobiApiTmd.h

  *Thermal Mitigation Device API function prototypes.*

### 6.22.1 Detailed Description

## 6.23 AVMS (SWIAVMS)

### Files

- file qaGobiApiSwiAvms.h

  *SWI Open Mobile Alliance Device Management Service API function prototypes SWI AVMS QMI Service revision 1.14.*

### 6.23.1 Detailed Description

## 6.24 SWIDMS

### Files

- file qaGobiApiSwiDms.h

  *SWI DMS API function prototypes.*

### 6.24.1 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 _getIndicationRegResp Struct Reference

### Data Fields

- BYTE * pRegTransLayerInfoEvt
- BYTE * pRegTransNWRegInfoEvt
- BYTE * pRegCallStatInfoEvt

### 8.1.1 Detailed Description

This structure contains Get Indication Register Response parameters

**Parameters**

| | |
|---|---|
| *pRegTrans-LayerInfoEvt* | -<br>• Optional 1 BYTE parameter indicating registration status of transport layer information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• function SLQSGetIndicationRegister() returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |
| *pRegTransNW-RegInfoEvt* | -<br>• Optional 1 BYTEparameter indicating registration status of transport network registration information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• function SLQSGetIndicationRegister() returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |
| *pRegCallStat-InfoEvt* | -<br>• Optional 1 BYTE parameter indicating registration status of call status information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• function SLQSGetIndicationRegister() returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |

### 8.1.2 Field Documentation

#### 8.1.2.1 BYTE* _getIndicationRegResp::pRegCallStatInfoEvt

#### 8.1.2.2 BYTE* _getIndicationRegResp::pRegTransLayerInfoEvt

#### 8.1.2.3 BYTE* _getIndicationRegResp::pRegTransNWRegInfoEvt

## 8.2 _GetProfileSettingIn Struct Reference

### Data Fields

- BYTE ProfileType
- BYTE ProfileID

### 8.2.1 Detailed Description

This structure contains the input parameters for SLQSGetProfileSettings

**Parameters**

| | |
|---|---|
| *ProfileType* | • Identifies the technology type of the profile<br>– 0x00 - 3GPP<br>– 0x01 - 3GPP2 |
| *ProfileID* | • index identifying the profile |

### 8.2.2 Field Documentation

#### 8.2.2.1 BYTE _GetProfileSettingIn::ProfileID

#### 8.2.2.2 BYTE _GetProfileSettingIn::ProfileType

## 8.3 _GetProfileSettingOut Struct Reference

### Data Fields

- QmiProfileInfo curProfile
- WORD * pExtErrCode

### 8.3.1 Detailed Description

This structure contains the profile settings retrieved by the API SLQSGetProfileSettings

**Parameters**

| | |
|---|---|
| *curProfile* | • Structure containing details of the profile<br>• See QmiProfileInfo for more details |

| | |
|---|---|
| *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. |

### 8.3.2 Field Documentation

#### 8.3.2.1 QmiProfileInfo _GetProfileSettingOut::curProfile

#### 8.3.2.2 WORD* _GetProfileSettingOut::pExtErrCode

## 8.4 _getResetInfoNotification Struct Reference

### Data Fields

- BYTE type
- BYTE source

### 8.4.1 Detailed Description

Contains the parameters passed for SLQSSetSwiGetResetInfoCallback by the device.

**Parameters**

| | |
|---|---|
| *type* | • 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 |
| *source* | • entity which initiated the reset or power down, possible values listed below:<br>– 0 - unknown<br>– 1 - user requested<br>– 2 - hardware switch<br>– 3 - temperature critical<br>– 4 - voltage critical<br>– 5 - configuration update<br>– 6 - LWM2M<br>– 7 - OMA-DM<br>– 8 - FOTA |

**Note**

None

### 8.4.2 Field Documentation

#### 8.4.2.1 BYTE _getResetInfoNotification::source

#### 8.4.2.2 BYTE _getResetInfoNotification::type

## 8.5 _getTransLayerInfoResp Struct Reference

### Data Fields

- BYTE * pRegInd
- transLayerInfo * pTransLayerInfo

### 8.5.1 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>• function SLQSGetTransLayerInfo() returns a default value 0xFF if no response is received from the device. |
| *pTransLayerInfo* | • Pointer to structure of transLayerInfo.<br>– Optional parameter<br>– See transLayerInfo for more information<br>• function SLQSGetTransLayerInfo() returns a default value 0xFF for parameter values if no response is received from the device. |

### 8.5.2 Field Documentation

#### 8.5.2.1 BYTE* _getTransLayerInfoResp::pRegInd

#### 8.5.2.2 transLayerInfo* _getTransLayerInfoResp::pTransLayerInfo

## 8.6 _getTransNWRegInfoResp Struct Reference

### Data Fields

- BYTE * pRegStatus

### 8.6.1 Detailed Description

This structure contains transport network registration info parameter

**Parameters**

| | |
|---|---|
| *pRegStatus* | - <br>• Optional 1 BYTE 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>• function SLQSGetTransNWRegInfo() returns a default value 0xFF if no response is received from the device. |

### 8.6.2 Field Documentation

#### 8.6.2.1 BYTE* _getTransNWRegInfoResp::pRegStatus

## 8.7 _MitigationDevInfo Struct Reference

### Data Fields

- BYTE deviceIdLen
- CHAR deviceID [255]

### 8.7.1 Detailed Description

This structure contains mitigation Level Indication request parameters

**Parameters**

| | |
|---|---|
| *deviceIdLen* | • Number of sets of the following elements <br>– deviceID |
| *deviceID* | • Mitigation device ID |

### 8.7.2 Field Documentation

#### 8.7.2.1 CHAR _MitigationDevInfo::deviceID[255]

#### 8.7.2.2 BYTE _MitigationDevInfo::deviceIdLen

## 8.8 _modemTempNotification Struct Reference

### Data Fields

- BYTE ModemTempState
- WORD ModemTemperature

### 8.8.1 Detailed Description

Contains the parameters passed for SLQSSetModemTempCallback by the device.

**Parameters**

| | |
|---|---|
| *ModemTemp-State* | • provides the temperature state of the modem<br>• Values:<br>– 0 - unknown<br>– 1 - normal<br>– 2 - high(warning)<br>– 3 - high(critical)<br>– 4 - low(critical) |
| *Modem-Temperature* | • provides the temperature of the modem |

**Note**

None

### 8.8.2 Field Documentation

#### 8.8.2.1 WORD _modemTempNotification::ModemTemperature

#### 8.8.2.2 BYTE _modemTempNotification::ModemTempState

## 8.9 _packetSrvStatus Struct Reference

### Data Fields

- qaQmiInterfaceInfo * pQmiInterfaceInfo
- BYTE connStatus
- BYTE reconfigReqd
- WORD sessionEndReason
- WORD verboseSessnEndReasonType
- WORD verboseSessnEndReason
- BYTE ipFamily
- WORD techName
- BYTE bearerID

### 8.9.1 Detailed Description

Contains the parameters passed for SLQSSetPacketSrvStatusCallback by the device.

**Parameters**

| | |
|---|---|
| *pQmiInterface-Info* | • See qaQmiInterfaceInfo for more information |

| | |
|---|---|
| *connStatus* | • Current Link Status<br>– 1 - Disconnected<br>– 2 - Connected<br>– 3 - Suspended<br>– 4 - Authenticating |
| *reconfigReqd* | • Indicates if the network interface on the host needs to be reconfigured<br>– 0 - No need to reconfigure<br>– 1 - Reconfiguration required |
| *sessionEnd-Reason* | • See qaGobiApiTableCallEndReasons.h for Call End Reason, 0xFFFF means invalid value |
| *verboseSessn-EndReasonType* | • Call End Reason Type<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 value |
| *verboseSessn-EndReason* | • See qaGobiApiTableCallEndReasons.h for verbose Call End Reason. The values depend on verboseSessnEndReasonType parameter 0xFFFF means invalid value |
| *ipFamily* | • IP Family of the packet data connection<br>– 4 - IPv4<br>– 6 - IPv6<br>– 0xFF - invalid value |
| *techName* | • Technology name of the packet data connection.<br>– 32767 - CDMA<br>– 32764 - UMTS<br>– 30592 - EPC<br>– 30590 - EMBMS<br>– 30584 - Modem Link Local<br>– 0xFFFF - invalid value EPC is a logical interface to support LTE/eHRPD handoff. Modem Link is an interface for transferring data between entities on the AP and modem. |
| *bearerID* | • Bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection 0xFF means invalid value |

**Note**

Any parameter not returned by the device is returned as its maximum unsigned value by the callback.

### 8.9.2 Field Documentation

#### 8.9.2.1 BYTE _packetSrvStatus::bearerID

#### 8.9.2.2 BYTE _packetSrvStatus::connStatus

#### 8.9.2.3 BYTE _packetSrvStatus::ipFamily

#### 8.9.2.4 qaQmiInterfaceInfo* _packetSrvStatus::pQmiInterfaceInfo

#### 8.9.2.5 BYTE _packetSrvStatus::reconfigReqd

#### 8.9.2.6 WORD _packetSrvStatus::sessionEndReason

#### 8.9.2.7 WORD _packetSrvStatus::techName

#### 8.9.2.8 WORD _packetSrvStatus::verboseSessnEndReason

#### 8.9.2.9 WORD _packetSrvStatus::verboseSessnEndReasonType

## 8.10 _qaQmi3GPP2BroadcastCfgInfo Struct Reference

### Data Fields

- BYTE activated_ind
- WORD num_instances
- struct CDMABroadcastConfig CDMABroadcastConfig [0x05]

### 8.10.1 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 CDMABroadcastConfig |

### 8.10.2 Field Documentation

#### 8.10.2.1 BYTE _qaQmi3GPP2BroadcastCfgInfo::activated_ind

#### 8.10.2.2 struct CDMABroadcastConfig _qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]

#### 8.10.2.3 WORD _qaQmi3GPP2BroadcastCfgInfo::num_instances

## 8.11 _qaQmi3GPPBroadcastCfgInfo Struct Reference

### Data Fields

- BYTE activated_ind
- WORD num_instances
- struct BroadcastConfig broadcastConfig [0x05]

### 8.11.1 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 BroadcastConfig |

### 8.11.2 Field Documentation

#### 8.11.2.1 BYTE _qaQmi3GPPBroadcastCfgInfo::activated_ind

#### 8.11.2.2 struct BroadcastConfig _qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

#### 8.11.2.3 WORD _qaQmi3GPPBroadcastCfgInfo::num_instances

## 8.12 _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp Struct Reference

### Data Fields

- BYTE bBinaryType
- BYTE bState

- BYTE bUserInputRequest
- WORD wUserInputTimeout
- ULONG ulPkgDownloadSize
- ULONG ulPkgDownloadComplete
- WORD wUpdateCompeteStatus
- BYTE bSerity
- WORD wVersionLength
- CHAR szVersion [128]
- WORD wNameLength
- CHAR szName [128]
- WORD wDescriptionLength
- CHAR szDescription [1024]
- BYTE TlvPresent

### 8.12.1 Detailed Description

**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 |

### 8.12.2 Field Documentation

#### 8.12.2.1 BYTE _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::bBinaryType

#### 8.12.2.2 BYTE _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::bSerity

#### 8.12.2.3 BYTE _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::bState

#### 8.12.2.4 BYTE _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::bUserInputRequest

#### 8.12.2.5 CHAR _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::szDescription[1024]

#### 8.12.2.6 CHAR _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::szName[128]

#### 8.12.2.7 CHAR _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::szVersion[128]

#### 8.12.2.8 BYTE _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::TlvPresent

#### 8.12.2.9 ULONG _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::ulPkgDownloadComplete

#### 8.12.2.10 ULONG _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::ulPkgDownloadSize

#### 8.12.2.11 WORD _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::wDescriptionLength

#### 8.12.2.12 WORD _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::wNameLength

#### 8.12.2.13 WORD _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::wUpdateCompeteStatus

#### 8.12.2.14 WORD _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::wUserInputTimeout

#### 8.12.2.15 WORD _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp::wVersionLength

## 8.13 _QmiSwiAvmsEventReportConfigResp Struct Reference

### Data Fields

- BYTE bState
- BYTE bUserInputRequest
- WORD wUserInputTimeout
- WORD wAlertMsgLength
- CHAR szAlertMsg [200]
- BYTE TlvPresent

### 8.13.1 Detailed Description

**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 |

### 8.13.2 Field Documentation

#### 8.13.2.1 BYTE _QmiSwiAvmsEventReportConfigResp::bState

#### 8.13.2.2 BYTE _QmiSwiAvmsEventReportConfigResp::bUserInputRequest

#### 8.13.2.3 CHAR _QmiSwiAvmsEventReportConfigResp::szAlertMsg[200]

#### 8.13.2.4 BYTE _QmiSwiAvmsEventReportConfigResp::TlvPresent

#### 8.13.2.5 WORD _QmiSwiAvmsEventReportConfigResp::wAlertMsgLength

#### 8.13.2.6 WORD _QmiSwiAvmsEventReportConfigResp::wUserInputTimeout

## 8.14 _QmiSwiAvmsEventReportConnectionRequestResp Struct Reference

### Data Fields

- BYTE bUserInputRequest
- WORD wUserInputTimeout
- BYTE TlvPresent

### 8.14.1 Detailed Description

**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 |

### 8.14.2 Field Documentation

#### 8.14.2.1 BYTE _QmiSwiAvmsEventReportConnectionRequestResp::bUserInputRequest

#### 8.14.2.2 BYTE _QmiSwiAvmsEventReportConnectionRequestResp::TlvPresent

#### 8.14.2.3 WORD _QmiSwiAvmsEventReportConnectionRequestResp::wUserInputTimeout

## 8.15 _QmiSwiAvmsEventReportDataSessionStatusResp Struct Reference

### Data Fields

- BYTE bType
- WORD wErrorCode
- BYTE TlvPresent

### 8.15.1 Detailed Description

**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 |

### 8.15.2 Field Documentation

#### 8.15.2.1 BYTE _QmiSwiAvmsEventReportDataSessionStatusResp::bType

#### 8.15.2.2 BYTE _QmiSwiAvmsEventReportDataSessionStatusResp::TlvPresent

#### 8.15.2.3 WORD _QmiSwiAvmsEventReportDataSessionStatusResp::wErrorCode

## 8.16 _QmiSwiAvmsEventReportHTTPStatusResp Struct Reference

### Data Fields

- WORD wHTTPStatus
- BYTE TlvPresent

### 8.16.1 Detailed Description

**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 |

## 8.16.2 Field Documentation

### 8.16.2.1 BYTE _QmiSwiAvmsEventReportHTTPStatusResp::TlvPresent

#### 8.16.2.2 WORD _QmiSwiAvmsEventReportHTTPStatusResp::wHTTPStatus

## 8.17 _QmiSwiAvmsEventReportNotificationResp Struct Reference

### Data Fields

- BYTE bNotification
- WORD wSessionStatus
- BYTE TlvPresent

### 8.17.1 Detailed Description

**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 |

### 8.17.2 Field Documentation

#### 8.17.2.1 BYTE _QmiSwiAvmsEventReportNotificationResp::bNotification

#### 8.17.2.2 BYTE _QmiSwiAvmsEventReportNotificationResp::TlvPresent

#### 8.17.2.3 WORD _QmiSwiAvmsEventReportNotificationResp::wSessionStatus

## 8.18 _QmiSwiAvmsEventReportPackageIDResp Struct Reference

### Data Fields

- BYTE bPackageID
- BYTE TlvPresent

### 8.18.1 Detailed Description

**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 |

### 8.18.2 Field Documentation

#### 8.18.2.1 BYTE _QmiSwiAvmsEventReportPackageIDResp::bPackageID

#### 8.18.2.2 BYTE _QmiSwiAvmsEventReportPackageIDResp::TlvPresent

## 8.19 _QmiSwiAvmsEventReportRegStatusResp Struct Reference

### Data Fields

- BYTE bRegStatus
- BYTE TlvPresent

### 8.19.1 Detailed Description

**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 |

### 8.19.2 Field Documentation

#### 8.19.2.1 BYTE _QmiSwiAvmsEventReportRegStatusResp::bRegStatus

#### 8.19.2.2 BYTE _QmiSwiAvmsEventReportRegStatusResp::TlvPresent

## 8.20 _QmiSwiAvmsEventReportSessionTypeResp Struct Reference

### Data Fields

- BYTE bType
- BYTE TlvPresent

### 8.20.1 Detailed Description

**Parameters**

| | |
|---|---|
| *bType* | Session Type.<br>• 0: Bootstrap session.<br>• 1. DM session |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.20.2 Field Documentation

#### 8.20.2.1 BYTE _QmiSwiAvmsEventReportSessionTypeResp::bType

#### 8.20.2.2 BYTE _QmiSwiAvmsEventReportSessionTypeResp::TlvPresent

## 8.21 _QmiSwiAvmsEventReportWAMSParamChangeResp Struct Reference

### Data Fields

- WORD wWamsChangeMask
- BYTE TlvPresent

### 8.21.1 Detailed Description

**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 |

### 8.21.2 Field Documentation

#### 8.21.2.1 BYTE _QmiSwiAvmsEventReportWAMSParamChangeResp::TlvPresent

#### 8.21.2.2 WORD _QmiSwiAvmsEventReportWAMSParamChangeResp::wWamsChangeMask

## 8.22 _QmiSwiAvmsSetSettingsAPNInfo Struct Reference

### Data Fields

- BYTE bAPNLength
- CHAR szAPN [49]
- BYTE bUnameLength
- CHAR szUname [29]
- BYTE bPWDLength
- CHAR szPWD [29]

### 8.22.1 Field Documentation

#### 8.22.1.1 BYTE _QmiSwiAvmsSetSettingsAPNInfo::bAPNLength

#### 8.22.1.2 BYTE _QmiSwiAvmsSetSettingsAPNInfo::bPWDLength

#### 8.22.1.3 BYTE _QmiSwiAvmsSetSettingsAPNInfo::bUnameLength

#### 8.22.1.4 CHAR _QmiSwiAvmsSetSettingsAPNInfo::szAPN[49]

#### 8.22.1.5 CHAR _QmiSwiAvmsSetSettingsAPNInfo::szPWD[29]

#### 8.22.1.6 CHAR _QmiSwiAvmsSetSettingsAPNInfo::szUname[29]

## 8.23 _QmiSwiAvmsSetSettingsConnectionRetryTimers Struct Reference

### Data Fields

- WORD Timers [8]

### 8.23.1 Field Documentation

#### 8.23.1.1 WORD _QmiSwiAvmsSetSettingsConnectionRetryTimers::Timers[8]

## 8.24 _QmiSwiAvmsSetSettingsPeriodInfo Struct Reference

### Data Fields

- ULONG ulMin
- ULONG ulMax

### 8.24.1 Field Documentation

#### 8.24.1.1 ULONG _QmiSwiAvmsSetSettingsPeriodInfo::ulMax

#### 8.24.1.2 ULONG _QmiSwiAvmsSetSettingsPeriodInfo::ulMin

## 8.25 _setIndicationRegReq Struct Reference

### Data Fields

- BYTE * pRegTransLayerInfoEvt
- BYTE * pRegTransNWRegInfoEvt

- BYTE * pRegCallStatInfoEvt

### 8.25.1 Detailed Description

This structure contains Indication Register request parameters

**Parameters**

| | |
|---|---|
| *pRegTrans-LayerInfoEvt* | -<br>• Optional 1 BYTE 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 1 BYTE 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 1 BYTE 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 |

### 8.25.2 Field Documentation

#### 8.25.2.1 BYTE* _setIndicationRegReq::pRegCallStatInfoEvt

#### 8.25.2.2 BYTE* _setIndicationRegReq::pRegTransLayerInfoEvt

#### 8.25.2.3 BYTE* _setIndicationRegReq::pRegTransNWRegInfoEvt

## 8.26 _slqs3GPPConfigItem Struct Reference

### Data Fields

- WORD * pLTEAttachProfile
- WORD * pProfileList
- BYTE * pDefaultPDNEnabled
- BYTE * p3gppRelease
- WORD LTEAttachProfileListLen
- WORD * pLTEAttachProfileList

### 8.26.1 Detailed Description

This structure contains the 3gpp Configuration Item information.

**Parameters**

| | |
|---|---|
| *pLTEAttach-Profile* | • 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>– function SLQSGet3GPPConfigItem() returns a default value 255 if no LTE Attach Profile is configured<br>• This setting is deprecated on MC/EM74xx |
| *pProfileList* | • 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* | • Always Connect Default PDN<br>– A single BYTE value indicating the status of Always connect default PDN<br>* 0 - disabled<br>* 1 - enabled<br>– Optional parameter<br>– function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present |
| *p3gppRelease* | • 3gpp release<br>– A single BYTE value indicating the 3gpp release<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<br>– Optional parameter<br>– function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present |

| | |
|---|---|
| *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>– The new equivalent option for "pLTEAttachProfile" on 74xx modems is "pLTEAttachProfile-List". 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 |

### 8.26.2 Field Documentation

#### 8.26.2.1 WORD _slqs3GPPConfigItem::LTEAttachProfileListLen

#### 8.26.2.2 BYTE* _slqs3GPPConfigItem::p3gppRelease

#### 8.26.2.3 BYTE* _slqs3GPPConfigItem::pDefaultPDNEnabled

#### 8.26.2.4 WORD* _slqs3GPPConfigItem::pLTEAttachProfile

#### 8.26.2.5 WORD* _slqs3GPPConfigItem::pLTEAttachProfileList

#### 8.26.2.6 WORD* _slqs3GPPConfigItem::pProfileList

## 8.27 _SLQSAVMSSetSettingsReqPara Struct Reference

### Data Fields

- BYTE * pAutoConnect
- BYTE * pAutoReboot
- BYTE * pFwPromptdownload
- BYTE * pFwPromptUpdate
- BYTE * pFwAutoSDM
- ULONG * pPollingTimer
- QmiSwiAvmsSetSettingsConnectionRetryTimers * pConnectionRetryTimers
- QmiSwiAvmsSetSettingsAPNInfo * pAPNInfo
- BYTE * pNotifStore
- QmiSwiAvmsSetSettingsPeriodInfo * pPeroidInfo

### 8.27.1 Detailed Description

Sets the AVMS settings requested.

**Parameters**

| | |
|---|---|
| *bAutoConnect[IN]* | • Auto Connect to AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *bAutoReboot[IN]* | • Automatic device reboot when the request is received from AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *bFwPromptdownload[IN]* | • Start Firmware Download promptly/automatically<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *bFwPromptUpdate[IN]* | • Start Firmware Download promptly/automatically<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *FwAutoSDM[Optional]* | - OMA Automatic UI Alert Response<br>• 0x00 - DISABLED<br>• 0x01 - ENABLED ACCEPT<br>• 0x02 - ENABLED REJECT |
| *PollingTimer[Optional]* | • 0-525600 (min)<br>• 0:disabled |
| *ConnectionRetryTimers[Optional]* | • See QmiSwiAvmsSetSettingsConnectionRetryTimers for more information |
| *APNInfo[Optional]* | • See QmiSwiAvmsSetSettingsAPNInfo for more information |
| *NotifStore[Optional]* | • 0 : Disabled.<br>• 1 : Enabeld. |
| *PeroidInfo[Optional]* | • See QmiSwiAvmsSetSettingsPeriodInfo 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

### 8.27.2 Field Documentation

#### 8.27.2.1 QmiSwiAvmsSetSettingsAPNInfo* _SLQSAVMSSetSettingsReqPara::pAPNInfo

#### 8.27.2.2 BYTE* _SLQSAVMSSetSettingsReqPara::pAutoConnect

#### 8.27.2.3 BYTE* _SLQSAVMSSetSettingsReqPara::pAutoReboot

#### 8.27.2.4 QmiSwiAvmsSetSettingsConnectionRetryTimers* _SLQSAVMSSetSettingsReqPara::pConnectionRetryTimers

#### 8.27.2.5 BYTE* _SLQSAVMSSetSettingsReqPara::pFwAutoSDM

#### 8.27.2.6 BYTE* _SLQSAVMSSetSettingsReqPara::pFwPromptdownload

#### 8.27.2.7 BYTE* _SLQSAVMSSetSettingsReqPara::pFwPromptUpdate

#### 8.27.2.8 BYTE* _SLQSAVMSSetSettingsReqPara::pNotifStore

#### 8.27.2.9 QmiSwiAvmsSetSettingsPeriodInfo* _SLQSAVMSSetSettingsReqPara::pPeroidInfo

#### 8.27.2.10 ULONG* _SLQSAVMSSetSettingsReqPara::pPollingTimer

## 8.28 _SLQSAVMSSettings Struct Reference

### Data Fields

- ULONG * pOMADMEnabled
- BYTE * pAutoConnect
- BYTE * pAutoReboot
- BYTE * pFwPromptdownload
- BYTE * pFwPromptUpdate
- BYTE * pFwAutoSDM
- ULONG * pPollingTimer
- SLQSAVMSSettingsConnectionRetryTimers * pConnectionRetryTimers
- SLQSAVMSSettingsAPNInfo * pAPNInfo
- BYTE * pNotificationStore
- SLQSAVMSSettingsPeriodsInfo * pPeroidsInfo

### 8.28.1 Field Documentation

#### 8.28.1.1 SLQSAVMSSettingsAPNInfo* _SLQSAVMSSettings::pAPNInfo

#### 8.28.1.2 BYTE* _SLQSAVMSSettings::pAutoConnect

#### 8.28.1.3 BYTE* _SLQSAVMSSettings::pAutoReboot

#### 8.28.1.4 SLQSAVMSSettingsConnectionRetryTimers* _SLQSAVMSSettings::pConnectionRetryTimers

#### 8.28.1.5 BYTE* _SLQSAVMSSettings::pFwAutoSDM

#### 8.28.1.6 BYTE* _SLQSAVMSSettings::pFwPromptdownload

#### 8.28.1.7 BYTE* _SLQSAVMSSettings::pFwPromptUpdate

#### 8.28.1.8 BYTE* _SLQSAVMSSettings::pNotificationStore

#### 8.28.1.9 ULONG* _SLQSAVMSSettings::pOMADMEnabled

#### 8.28.1.10 SLQSAVMSSettingsPeriodsInfo* _SLQSAVMSSettings::pPeroidsInfo

#### 8.28.1.11 ULONG* _SLQSAVMSSettings::pPollingTimer

## 8.29 _SLQSAVMSSettingsAPNInfo Struct Reference

### Data Fields

- BYTE bAPNLength
- CHAR * pAPN
- BYTE bUnameLength
- CHAR * pUname
- BYTE bPWDLength
- CHAR * pPWD

### 8.29.1 Field Documentation

#### 8.29.1.1 BYTE _SLQSAVMSSettingsAPNInfo::bAPNLength

#### 8.29.1.2 BYTE _SLQSAVMSSettingsAPNInfo::bPWDLength

#### 8.29.1.3 BYTE _SLQSAVMSSettingsAPNInfo::bUnameLength

#### 8.29.1.4 CHAR* _SLQSAVMSSettingsAPNInfo::pAPN

#### 8.29.1.5 CHAR* _SLQSAVMSSettingsAPNInfo::pPWD

#### 8.29.1.6 CHAR* _SLQSAVMSSettingsAPNInfo::pUname

## 8.30 _SLQSAVMSSettingsConnectionRetryTimers Struct Reference

### Data Fields

- WORD Timers [8]

### 8.30.1 Field Documentation

#### 8.30.1.1 WORD _SLQSAVMSSettingsConnectionRetryTimers::Timers[8]

## 8.31 _SLQSAVMSSettingsPeriodsInfo Struct Reference

### Data Fields

- ULONG min
- ULONG max

### 8.31.1 Field Documentation

#### 8.31.1.1 ULONG _SLQSAVMSSettingsPeriodsInfo::max

#### 8.31.1.2 ULONG _SLQSAVMSSettingsPeriodsInfo::min

## 8.32 _SlqsNas3GppNetworkRAT_ Struct Reference

### Data Fields

- WORD MCC
- WORD MNC
- BYTE RAT

### 8.32.1 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 |

### 8.32.2 Field Documentation

#### 8.32.2.1 WORD _SlqsNas3GppNetworkRAT_::MCC

#### 8.32.2.2 WORD _SlqsNas3GppNetworkRAT_::MNC

#### 8.32.2.3 BYTE _SlqsNas3GppNetworkRAT_::RAT

## 8.33 _slqsNetworkScanInfo Struct Reference

### Data Fields

- BYTE * pNetworkInfoInstances
- struct SlqsNas3GppNetworkInfo * pNetworkInfo
- BYTE * pRATInstances
- SlqsNas3GppNetworkRAT * pRATInfo
- BYTE * pPCSDigitInstances
- struct SlqsNasPcsDigit * pPCSDigitInfo
- ULONG * pScanResult

### 8.33.1 Detailed Description

Contain the network scan information.

**Parameters**

| | |
|---|---|
| *pNetworkInfoInstances[IN/OUT]* | • 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. |
| *pNetworkInfo[OUT]* | • Network info instance array<br>– See SlqsNas3GppNetworkInfo for more information |
| *pRATInstances[IN/OUT]* | • 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. |
| *pRATInfo[OUT]* | • RAT info instance array<br>– See SlqsNas3GppNetworkRAT for more information |
| *pPCSDigitInstances[IN/OUT]* | • 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. |
| *pPCSDigitInfo[OUT]* | • PCS Digit info instance array<br>– See SlqsNasPcsDigit for more information |
| *pScanResult[OUT]* | • 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 |

### 8.33.2 Field Documentation

#### 8.33.2.1 struct SlqsNas3GppNetworkInfo* _slqsNetworkScanInfo::pNetworkInfo

#### 8.33.2.2 BYTE* _slqsNetworkScanInfo::pNetworkInfoInstances

#### 8.33.2.3 struct SlqsNasPcsDigit* _slqsNetworkScanInfo::pPCSDigitInfo

#### 8.33.2.4 BYTE* _slqsNetworkScanInfo::pPCSDigitInstances

#### 8.33.2.5 SlqsNas3GppNetworkRAT* _slqsNetworkScanInfo::pRATInfo

#### 8.33.2.6 BYTE* _slqsNetworkScanInfo::pRATInstances

#### 8.33.2.7 ULONG* _slqsNetworkScanInfo::pScanResult

## 8.34 _SLQSOMADMSessionInfo Struct Reference

### Data Fields

- BYTE * pStatus
- WORD * pUpdateCompleteStatus
- BYTE * pSeverity
- WORD * pSourceLength
- BYTE * pSource
- WORD * pPkgNameLength
- BYTE * pPkgName
- WORD * pPkgDescLength
- BYTE * pPkgDescription
- WORD * pDateLength
- BYTE * pDate
- WORD * pTimeLength
- BYTE * pTime
- BYTE * pSessionType
- BYTE * pSessionState
- WORD * pRetryCount

### 8.34.1 Detailed Description

Structure containing the OMA DM Session Info returned by the device. Also used as input parameter to specify the size of variable parameters. (ref. notes)

**Parameters**

| | |
|---|---|
| *pStatus* | • 1 Byte parameter indicating status(optional)<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 |
| *pUpdate-CompleteStatus* | • 2 byte parameter indicating Update Complete Status(optional)<br>– See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h Update Complete Status |
| *pSeverity* | • 1 byte parameter indicating severity(optional)<br>– 0x01 - Mandatory<br>– 0x02 - Optional |
| *pSourceLength* | • 2 byte parameter indicating Length of Vendor Name String in Bytes.(optional) |
| *pSource* | • Variable length parameter indicating Vendor Name in ASCII(optional) |

| | |
|---|---|
| *pPkgName-Length* | • 2 byte parameter indicating Length of Package Name String in Bytes.(optional) |
| *pPkgName* | • Variable length parameter indicating Package Name in ASCII(optional) |
| *pPkgDesc-Length* | • 2 byte parameter indicating Length of Package Description String in Bytes.(optional) |
| *pPkgDescription* | • Variable length parameter indicating Package Description in ASCII(optional) |
| *pDateLength* | • 2 byte parameter indicating Length of Package Description String in Bytes.(optional) |
| *pDate* | • Variable length parameter indicating Package Description in ASCII |
| *pTimeLength* | • 2 byte parameter indicating Length of Time String in Bytes.(optional) |
| *pTime* | • Variable length parameter indicating Time String in ASCII(optional) |
| *pSessionType* | • 1 byte parameter reflects the last session started for Sprint(optional)<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 |
| *pSessionState* | • 1 byte parameter indicating session state(optional)<br>– 0x01 - idle<br>– 0x02 - active<br>– 0x03 - pending |
| *pRetryCount* | • 1 byte parameter indicating retries left count(optional)<br>– valid values 0 to 6 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

As input parameter the members pSourceLength, pPkgNameLength pPkgDescLength, pDateLength, pTimeLength have to be specified. These should contain the initialized size of pSource, pPkgName, pPkgDescription, pDate, pTime respectively.

### 8.34.2 Field Documentation

#### 8.34.2.1 BYTE* _SLQSOMADMSessionInfo::pDate

#### 8.34.2.2 WORD* _SLQSOMADMSessionInfo::pDateLength

#### 8.34.2.3 WORD* _SLQSOMADMSessionInfo::pPkgDescLength

#### 8.34.2.4 BYTE* _SLQSOMADMSessionInfo::pPkgDescription

#### 8.34.2.5 BYTE* _SLQSOMADMSessionInfo::pPkgName

#### 8.34.2.6 WORD* _SLQSOMADMSessionInfo::pPkgNameLength

#### 8.34.2.7 WORD* _SLQSOMADMSessionInfo::pRetryCount

#### 8.34.2.8 BYTE* _SLQSOMADMSessionInfo::pSessionState

#### 8.34.2.9 BYTE* _SLQSOMADMSessionInfo::pSessionType

#### 8.34.2.10 BYTE* _SLQSOMADMSessionInfo::pSeverity

#### 8.34.2.11 BYTE* _SLQSOMADMSessionInfo::pSource

#### 8.34.2.12 WORD* _SLQSOMADMSessionInfo::pSourceLength

#### 8.34.2.13 BYTE* _SLQSOMADMSessionInfo::pStatus

#### 8.34.2.14 BYTE* _SLQSOMADMSessionInfo::pTime

#### 8.34.2.15 WORD* _SLQSOMADMSessionInfo::pTimeLength

#### 8.34.2.16 WORD* _SLQSOMADMSessionInfo::pUpdateCompleteStatus

## 8.35 _SLQSOMADMSettings Struct Reference

### Data Fields

- ULONG * pOMADMEnabled
- BYTE * pFOTAdownload
- BYTE * pFOTAUpdate
- BYTE * pAutosdm
- BYTE * pFwAutoCheck

### 8.35.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

**Parameters**

| | |
|---|---|
| *pOMADM-Enabled[OUT]* | • 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. |
| *pFOT-Adownload[OUT]* | • 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>• function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem. |
| *pFOTAUpdate[-OUT]* | • 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. |
| *pAutosdm[OUT]* | • 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. |
| *pFwAutoCheck[-OUT]* | • 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. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 8.35.2 Field Documentation

#### 8.35.2.1 BYTE* _SLQSOMADMSettings::pAutosdm

#### 8.35.2.2 BYTE* _SLQSOMADMSettings::pFOTAdownload

#### 8.35.2.3 BYTE* _SLQSOMADMSettings::pFOTAUpdate

#### 8.35.2.4 BYTE* _SLQSOMADMSettings::pFwAutoCheck

#### 8.35.2.5 ULONG* _SLQSOMADMSettings::pOMADMEnabled

## 8.36 _SLQSOMADMSettingsReqParams Struct Reference

### Data Fields

- BYTE FOTAdownload
- BYTE FOTAUpdate
- BYTE * pAutosdm

### 8.36.1 Detailed Description

Structure containing the OMA DM settings to be set on the device

**Parameters**

| | |
|---|---|
| *FOTAdownload* | • 1 Byte parameter indicating support for FOTA Automatic download<br>– 0x00 - Firmware auto download FALSE<br>– 0x01 - Firmware auto download TRUE |
| *FOTAUpdate* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware auto update FALSE<br>– 0x01 - Firmware auto update TRUE |
| *pAutosdm[IN]* | • Optional 1 byte parameter indicating OMA Automatic UI Alert Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled Accept<br>– 0x02 - Enabled Reject |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 8.36.2 Field Documentation

#### 8.36.2.1 BYTE _SLQSOMADMSettingsReqParams::FOTAdownload

#### 8.36.2.2 BYTE _SLQSOMADMSettingsReqParams::FOTAUpdate

#### 8.36.2.3 BYTE* _SLQSOMADMSettingsReqParams::pAutosdm

## 8.37 _SLQSOMADMSettingsReqParams3 Struct Reference

### Data Fields

- BYTE FOTAdownload
- BYTE FOTAUpdate
- BYTE * pAutosdm
- BYTE * pFwAutoCheck

### 8.37.1 Detailed Description

Structure containing the OMA DM settings to be set on the device

**Parameters**

| | |
|---|---|
| *FOTAdownload* | • 1 Byte parameter indicating support for FOTA Automatic download<br>– 0x00 - Firmware auto download FALSE<br>– 0x01 - Firmware auto download TRUE |
| *FOTAUpdate* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware auto update FALSE<br>– 0x01 - Firmware auto update TRUE |
| *pAutosdm[IN]* | • Optional 1 byte parameter indicating OMA Automatic UI Alert Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled Accept<br>– 0x02 - Enabled Reject |
| *pFwAutoCheck[IN]* | • Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 8.37.2 Field Documentation

#### 8.37.2.1 BYTE _SLQSOMADMSettingsReqParams3::FOTAdownload

#### 8.37.2.2 BYTE _SLQSOMADMSettingsReqParams3::FOTAUpdate

#### 8.37.2.3 BYTE* _SLQSOMADMSettingsReqParams3::pAutosdm

#### 8.37.2.4 BYTE* _SLQSOMADMSettingsReqParams3::pFwAutoCheck

## 8.38 _SLQSSwiGetHostDevInfoParams Struct Reference

### Data Fields

- BYTE bManSize
- CHAR * pManString
- BYTE bModelSize
- CHAR * pModelString
- BYTE bSWVerSize
- CHAR * pSWVerString
- BYTE bPlasmaIDSize
- CHAR * pPlasmaIDString

### 8.38.1 Detailed Description

This structure is used to Get Host Device Information

**Parameters**

| | |
|---|---|
| *bManSize[IN/OUT]* | • Host Device Manufacturer String Size |
| *pManString[OUT]* | • Host Device Manufacturer Name(Optional parameter)<br>• Null terminated ASCII String |
| *bModelSize[IN/OUT]* | • Host Device Model String Size |
| *pModelString[OUT]* | • Host Device Model String(Optional parameter)<br>• Null terminated ASCII string. |
| *bSWVerSize[IN/OUT]* | • Host Device Software Version String Size |
| *pSWVerString[OUT]* | • Host Device Software Version String(Optional parameter)<br>• Null terminated ASCII string |
| *bPlasmaIDSize[IN/OUT]* | • Host Device Plasma ID String Size |
| *pPlasmaIDString[OUT]* | • Host Device Plasma ID String(Optional parameter)<br>• Null terminated alphanumeric ASCII String. |

### 8.38.2 Field Documentation

#### 8.38.2.1 BYTE _SLQSSwiGetHostDevInfoParams::bManSize

#### 8.38.2.2 BYTE _SLQSSwiGetHostDevInfoParams::bModelSize

#### 8.38.2.3 BYTE _SLQSSwiGetHostDevInfoParams::bPlasmaIDSize

#### 8.38.2.4 BYTE _SLQSSwiGetHostDevInfoParams::bSWVerSize

#### 8.38.2.5 CHAR* _SLQSSwiGetHostDevInfoParams::pManString

#### 8.38.2.6 CHAR* _SLQSSwiGetHostDevInfoParams::pModelString

#### 8.38.2.7 CHAR* _SLQSSwiGetHostDevInfoParams::pPlasmaIDString

#### 8.38.2.8 CHAR* _SLQSSwiGetHostDevInfoParams::pSWVerString

## 8.39 _SLQSSwiGetOSInfoParams Struct Reference

### Data Fields

- BYTE bNameSize
- CHAR * pNameString
- BYTE bVersionSize
- CHAR * pVersionString

### 8.39.1 Detailed Description

This structure is used to Get OS Information

**Parameters**

| | |
|---|---|
| *bNameSize[IN/-OUT]* | • Size of Operating System Name |
| *pNameString[O-UT]* | • Operating System Name(Optional parameter)<br>• Null terminated ASCII string |
| *bVersionSize[I-N/OUT]* | • Operating System Version Size |
| *pVersionString[-OUT]* | • Operating System Version String(Optional parameter)<br>• Null terminated ASCII string. |

### 8.39.2 Field Documentation

#### 8.39.2.1 BYTE _SLQSSwiGetOSInfoParams::bNameSize

#### 8.39.2.2 BYTE _SLQSSwiGetOSInfoParams::bVersionSize

#### 8.39.2.3 CHAR* _SLQSSwiGetOSInfoParams::pNameString

#### 8.39.2.4 CHAR* _SLQSSwiGetOSInfoParams::pVersionString

## 8.40 _SLQSSwiGetSerialNoExtParams Struct Reference

### Data Fields

- BYTE meidLength
- CHAR * pMeidString

### 8.40.1 Detailed Description

This structure is used to store MEID Information

**Parameters**

| | |
|---|---|
| *meidLength[OUT]* | • String length of the of MEID received |
| *pMeidString[OUT]* | • Optional parameter<br>• Pointer to receive String containing the Mobile Equipment Identifier(MEID) of the device. |

### 8.40.2 Field Documentation

#### 8.40.2.1 BYTE _SLQSSwiGetSerialNoExtParams::meidLength

#### 8.40.2.2 CHAR* _SLQSSwiGetSerialNoExtParams::pMeidString

## 8.41 _SLQSSwiSetHostDevInfoParams Struct Reference

### Data Fields

- BYTE bManSize
- CHAR * pManString
- BYTE bModelSize
- CHAR * pModelString
- BYTE bSWVerSize
- CHAR * pSWVerString
- BYTE bPlasmaIDSize
- CHAR * pPlasmaIDString

### 8.41.1 Detailed Description

This structure is used to Set Host Device Information

**Parameters**

| | |
|---|---|
| *bManSize[IN]* | • Host Device Manufacturer String Size |

| | |
|---|---|
| *pManString[IN]* | • Host Device Manufacturer Name(Optional parameter)<br>• Null terminated ASCII String |
| *bModelSize[IN]* | • Host Device Model String Size |
| *pModelString[IN]* | • Host Device Model String(Optional parameter)<br>• Null terminated ASCII string. |
| *bSWVerSize[IN]* | • Host Device Software Version String Size |
| *pSWVerString[IN]* | • Host Device Software Version String(Optional parameter)<br>• Null terminated ASCII string |
| *bPlasmaIDSize[IN]* | • Host Device Plasma ID String Size |
| *pPlasmaIDString[IN]* | • Host Device Plasma ID String(Optional parameter)<br>• Null terminated alphanumeric ASCII String. |

### 8.41.2 Field Documentation

#### 8.41.2.1 BYTE _SLQSSwiSetHostDevInfoParams::bManSize

#### 8.41.2.2 BYTE _SLQSSwiSetHostDevInfoParams::bModelSize

#### 8.41.2.3 BYTE _SLQSSwiSetHostDevInfoParams::bPlasmaIDSize

#### 8.41.2.4 BYTE _SLQSSwiSetHostDevInfoParams::bSWVerSize

#### 8.41.2.5 CHAR* _SLQSSwiSetHostDevInfoParams::pManString

#### 8.41.2.6 CHAR* _SLQSSwiSetHostDevInfoParams::pModelString

#### 8.41.2.7 CHAR* _SLQSSwiSetHostDevInfoParams::pPlasmaIDString

#### 8.41.2.8 CHAR* _SLQSSwiSetHostDevInfoParams::pSWVerString

## 8.42 _SLQSSwiSetOSInfoParams Struct Reference

### Data Fields

- BYTE bNameSize
- CHAR * pNameString
- BYTE bVersionSize
- CHAR * pVersionString

### 8.42.1 Detailed Description

This structure is used to Set OS Information

**Parameters**

| | |
|---|---|
| *bNameSize[IN]* | • Size of Operating System Name |
| *pNameString[IN]* | • Operating System Name(Optional parameter)<br>• Null terminated ASCII string |
| *bVersionSize[IN]* | • Operating System Version Size |
| *pVersionString[IN]* | • Operating System Version String(Optional parameter)<br>• Null terminated ASCII string. |

### 8.42.2 Field Documentation

#### 8.42.2.1 BYTE _SLQSSwiSetOSInfoParams::bNameSize

#### 8.42.2.2 BYTE _SLQSSwiSetOSInfoParams::bVersionSize

#### 8.42.2.3 CHAR* _SLQSSwiSetOSInfoParams::pNameString

#### 8.42.2.4 CHAR* _SLQSSwiSetOSInfoParams::pVersionString

## 8.43 _SwiAvmsEventInd Struct Reference

### Data Fields

- QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp BinaryUpdateSessionInfo
- QmiSwiAvmsEventReportConfigResp Config
- QmiSwiAvmsEventReportNotificationResp Notification
- QmiSwiAvmsEventReportConnectionRequestResp ConnectionRequest
- QmiSwiAvmsEventReportWAMSParamChangeResp WAMSParaChanged
- QmiSwiAvmsEventReportPackageIDResp PackageID
- QmiSwiAvmsEventReportDataSessionStatusResp DataSessionStatus
- QmiSwiAvmsEventReportSessionTypeResp SessionType
- QmiSwiAvmsEventReportHTTPStatusResp HTTPStatus
- QmiSwiAvmsEventReportRegStatusResp RegStatus

### 8.43.1 Field Documentation

#### 8.43.1.1 QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp _SwiAvmsEventInd::BinaryUpdateSessionInfo

#### 8.43.1.2 QmiSwiAvmsEventReportConfigResp _SwiAvmsEventInd::Config

#### 8.43.1.3 QmiSwiAvmsEventReportConnectionRequestResp _SwiAvmsEventInd::ConnectionRequest

#### 8.43.1.4 QmiSwiAvmsEventReportDataSessionStatusResp _SwiAvmsEventInd::DataSessionStatus

#### 8.43.1.5 QmiSwiAvmsEventReportHTTPStatusResp _SwiAvmsEventInd::HTTPStatus

#### 8.43.1.6 QmiSwiAvmsEventReportNotificationResp _SwiAvmsEventInd::Notification

#### 8.43.1.7 QmiSwiAvmsEventReportPackageIDResp _SwiAvmsEventInd::PackageID

#### 8.43.1.8 QmiSwiAvmsEventReportRegStatusResp _SwiAvmsEventInd::RegStatus

#### 8.43.1.9 QmiSwiAvmsEventReportSessionTypeResp _SwiAvmsEventInd::SessionType

#### 8.43.1.10 QmiSwiAvmsEventReportWAMSParamChangeResp _SwiAvmsEventInd::WAMSParaChanged

## 8.44 _sysSelectPrefInfo Struct Reference

### Data Fields

- BYTE * pEmerMode
- WORD * pModePref
- ULONGLONG * pBandPref
- WORD * pPRLPref
- WORD * pRoamPref
- ULONGLONG * pLTEBandPref
- BYTE * pNetSelPref
- ULONG * pSrvDomainPref
- ULONG * pGWAcqOrderPref

### 8.44.1 Detailed Description

Structure for storing the current preferred system selection settings for the device.

**Parameters**

| | |
|---|---|
| *pEmerMode* | • Optional parameter specifying the emergency Mode<br>• Values:<br>– 0 - OFF (normal)<br>– 1 - ON (Emergency)<br>• function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device. |
| *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>• function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device. |

| | |
|---|---|
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device. |

| | |
|---|---|
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device. |
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device. |

| | |
|---|---|
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device. |

| | |
|---|---|
| *pNetSelPref* | • Optional parameter indicating network selection preference<br>• Values:<br>– 0x00 - Automatic network selection<br>– 0x01 - Manual network selection<br>• function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device. |
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFF if no value is returned by the device. |
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFF if no value is returned by the device. |

**Note**

None

### 8.44.2 Field Documentation

#### 8.44.2.1 ULONGLONG* _sysSelectPrefInfo::pBandPref

#### 8.44.2.2 BYTE* _sysSelectPrefInfo::pEmerMode

#### 8.44.2.3 ULONG* _sysSelectPrefInfo::pGWAcqOrderPref

#### 8.44.2.4 ULONGLONG* _sysSelectPrefInfo::pLTEBandPref

#### 8.44.2.5 WORD* _sysSelectPrefInfo::pModePref

#### 8.44.2.6 BYTE* _sysSelectPrefInfo::pNetSelPref

#### 8.44.2.7 WORD* _sysSelectPrefInfo::pPRLPref

#### 8.44.2.8 WORD* _sysSelectPrefInfo::pRoamPref

#### 8.44.2.9 ULONG* _sysSelectPrefInfo::pSrvDomainPref

## 8.45 _sysSelectPrefParams Struct Reference

### Data Fields

- BYTE * pEmerMode
- WORD * pModePref
- ULONGLONG * pBandPref
- WORD * pPRLPref
- WORD * pRoamPref
- ULONGLONG * pLTEBandPref
- struct netSelectionPref * pNetSelPref
- BYTE * pChgDuration
- BYTE * pMNCIncPCSDigStat
- ULONG * pSrvDomainPref
- ULONG * pGWAcqOrderPref
- ULONGLONG * pTdscdmaBandPref
- struct acqOrderPref * pAcqOrderPref
- ULONG * pSrvRegRestriction
- struct CSGID * pCSGID
- BYTE * pRAT

### 8.45.1 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 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 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 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 |

### 8.45.2 Field Documentation

#### 8.45.2.1 struct acqOrderPref* _sysSelectPrefParams::pAcqOrderPref

#### 8.45.2.2 ULONGLONG* _sysSelectPrefParams::pBandPref

#### 8.45.2.3 BYTE* _sysSelectPrefParams::pChgDuration

#### 8.45.2.4 struct CSGID* _sysSelectPrefParams::pCSGID

#### 8.45.2.5 BYTE* _sysSelectPrefParams::pEmerMode

#### 8.45.2.6 ULONG* _sysSelectPrefParams::pGWAcqOrderPref

#### 8.45.2.7 ULONGLONG* _sysSelectPrefParams::pLTEBandPref

#### 8.45.2.8 BYTE* _sysSelectPrefParams::pMNCIncPCSDigStat

#### 8.45.2.9 WORD* _sysSelectPrefParams::pModePref

#### 8.45.2.10 struct netSelectionPref* _sysSelectPrefParams::pNetSelPref

#### 8.45.2.11 WORD* _sysSelectPrefParams::pPRLPref

#### 8.45.2.12 BYTE* _sysSelectPrefParams::pRAT

#### 8.45.2.13 WORD* _sysSelectPrefParams::pRoamPref

#### 8.45.2.14 ULONG* _sysSelectPrefParams::pSrvDomainPref

#### 8.45.2.15 ULONG* _sysSelectPrefParams::pSrvRegRestriction

#### 8.45.2.16 ULONGLONG* _sysSelectPrefParams::pTdscdmaBandPref

## 8.46 _transLayerinfo Struct Reference

### Data Fields

- BYTE TransType
- BYTE TransCap

### 8.46.1 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 |

### 8.46.2 Field Documentation

#### 8.46.2.1 BYTE _transLayerinfo::TransCap

#### 8.46.2.2 BYTE _transLayerinfo::TransType

## 8.47 _transLayerInfoNotification Struct Reference

### Data Fields

- BYTE regInd
- transLayerInfo * pTransLayerInfo

### 8.47.1 Detailed Description

Contains the parameters passed for SLQSSetTransLayerInfoCallback by the device.

**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 |
| *pTransLayerInfo* | • Optional parameter<br>• See transLayerInfo for more information |

**Note**

None

### 8.47.2 Field Documentation

#### 8.47.2.1 transLayerInfo* _transLayerInfoNotification::pTransLayerInfo

#### 8.47.2.2 BYTE _transLayerInfoNotification::regInd

## 8.48 _transNWRegInfoNotification Struct Reference

### Data Fields

- BYTE NWRegStat

### 8.48.1 Detailed Description

Contains the parameters passed for SLQSSetTransNWRegInfoCallback 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 |

**Note**

None

### 8.48.2 Field Documentation

#### 8.48.2.1 BYTE _transNWRegInfoNotification::NWRegStat

## 8.49 accelAcceptReady_s Struct Reference

### Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

### 8.49.1 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. |

### 8.49.2 Field Documentation

#### 8.49.2.1 WORD accelAcceptReady_s::batchPerSec

#### 8.49.2.2 BYTE accelAcceptReady_s::injectEnable

#### 8.49.2.3 WORD accelAcceptReady_s::samplesPerBatch

## 8.50 accelTempAcceptReady_s Struct Reference

### Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

### 8.50.1 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. |

### 8.50.2 Field Documentation

#### 8.50.2.1 WORD accelTempAcceptReady_s::batchPerSec

#### 8.50.2.2 BYTE accelTempAcceptReady_s::injectEnable

#### 8.50.2.3 WORD accelTempAcceptReady_s::samplesPerBatch

## 8.51 acqOrderPref Struct Reference

### Data Fields

- BYTE acqOrdeLen
- BYTE ∗ pAcqOrder

### 8.51.1 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 |

### 8.51.2 Field Documentation

#### 8.51.2.1 BYTE acqOrderPref::acqOrdeLen

#### 8.51.2.2 BYTE* acqOrderPref::pAcqOrder

## 8.52 ActPilotPNElement Struct Reference

### Data Fields

- WORD ActSetPilotPN
- BYTE ActSetPilotPNStrength

### 8.52.1 Detailed Description

This structure describes Active Pilot PN elements

**Parameters**

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

### 8.52.2 Field Documentation

#### 8.52.2.1 WORD ActPilotPNElement::ActSetPilotPN

#### 8.52.2.2 BYTE ActPilotPNElement::ActSetPilotPNStrength

## 8.53 AddCDMASysInfo Struct Reference

### Data Fields

- WORD geoSysIdx
- WORD regPrd

### 8.53.1 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. |

### 8.53.2 Field Documentation

#### 8.53.2.1 WORD AddCDMASysInfo::geoSysIdx

#### 8.53.2.2 WORD AddCDMASysInfo::regPrd

## 8.54 AddSysInfo Struct Reference

### Data Fields

- WORD geoSysIdx
- ULONG cellBroadcastCap

### 8.54.1 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 |

### 8.54.2 Field Documentation

#### 8.54.2.1 ULONG AddSysInfo::cellBroadcastCap

#### 8.54.2.2 WORD AddSysInfo::geoSysIdx

## 8.55 airTimer Struct Reference

### Data Fields

- BYTE namID
- ULONG airTimerValue

### 8.55.1 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. |

### 8.55.2 Field Documentation

#### 8.55.2.1 ULONG airTimer::airTimerValue

#### 8.55.2.2 BYTE airTimer::namID

## 8.56 allCallsAlphaIDInfo Struct Reference

### Data Fields

- BYTE callID
- alphaIDInfo AlphaIDInfo

### 8.56.1 Detailed Description

This structure contains information for Alpha Identifier for All Calls

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *AlphaIDInfo* | • See alphaIDInfo for more information. |

### 8.56.2 Field Documentation

#### 8.56.2.1 alphaIDInfo allCallsAlphaIDInfo::AlphaIDInfo

#### 8.56.2.2 BYTE allCallsAlphaIDInfo::callID

## 8.57 allCallsDiagInfo Struct Reference

### Data Fields

- BYTE callID
- diagInfo DiagInfo

### 8.57.1 Detailed Description

This structure contains Diagnostic Information for All Calls

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *DiagInfo* | • See diagInfo for more information. |

### 8.57.2 Field Documentation

#### 8.57.2.1 BYTE allCallsDiagInfo::callID

#### 8.57.2.2 diagInfo allCallsDiagInfo::DiagInfo

## 8.58 allCallsUUSInfo Struct Reference

### Data Fields

- BYTE callID
- UUSInfo uusInfo

### 8.58.1 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 UUSInfo for more information. |

### 8.58.2 Field Documentation

#### 8.58.2.1 BYTE allCallsUUSInfo::callID

#### 8.58.2.2 UUSInfo allCallsUUSInfo::uusInfo

## 8.59 alphaIDInfo Struct Reference

### Data Fields

- BYTE alphaDcs
- BYTE alphaLen
- BYTE alphaText [255]

### 8.59.1 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_-DESCRIPTION-_LENGTH]* | • Data encoded as per the alpha_dcs |

### 8.59.2 Field Documentation

#### 8.59.2.1 BYTE alphaIDInfo::alphaDcs

#### 8.59.2.2 BYTE alphaIDInfo::alphaLen

#### 8.59.2.3 BYTE alphaIDInfo::alphaText[255]

## 8.60 altitudeSrcInfo Struct Reference

### Data Fields

- ULONG source
- ULONG linkage
- ULONG coverage

### 8.60.1 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 |

### 8.60.2 Field Documentation

#### 8.60.2.1 ULONG altitudeSrcInfo::coverage

#### 8.60.2.2 ULONG altitudeSrcInfo::linkage

#### 8.60.2.3 ULONG altitudeSrcInfo::source

## 8.61 altSrcInfo_t Struct Reference

### Data Fields

- uint32_t source
- uint32_t linkage
- uint32_t coverage

### 8.61.1 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 |

### 8.61.2 Field Documentation

#### 8.61.2.1 uint32_t altSrcInfo_t::coverage

#### 8.61.2.2 uint32_t altSrcInfo_t::linkage

#### 8.61.2.3 uint32_t altSrcInfo_t::source

## 8.62 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

### 8.62.1 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_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. |

### 8.62.2 Field Documentation

#### 8.62.2.1 uint8_t appStats::aidLength

#### 8.62.2.2 uint8_t appStats::aidVal[255]

#### 8.62.2.3 uint8_t appStats::appState

#### 8.62.2.4 uint8_t appStats::appType

#### 8.62.2.5 uint8_t appStats::persoFeature

#### 8.62.2.6 uint8_t appStats::persoRetries

#### 8.62.2.7 uint8_t appStats::persoState

#### 8.62.2.8 uint8_t appStats::persoUnblockRetries

#### 8.62.2.9 uint8_t appStats::pin1Retries

#### 8.62.2.10 uint8_t appStats::pin1State

#### 8.62.2.11 uint8_t appStats::pin2Retries

#### 8.62.2.12 uint8_t appStats::pin2State

#### 8.62.2.13 uint8_t appStats::puk1Retries

#### 8.62.2.14 uint8_t appStats::puk2Retries

#### 8.62.2.15 uint8_t appStats::univPin

## 8.63 appStatus Struct Reference

### Data Fields

- BYTE appType

- BYTE appState
- BYTE persoState
- BYTE persoFeature
- BYTE persoRetries
- BYTE persoUnblockRetries
- BYTE aidLength
- BYTE aidVal [255]
- BYTE univPin
- BYTE pin1State
- BYTE pin1Retries
- BYTE puk1Retries
- BYTE pin2State
- BYTE pin2Retries
- BYTE puk2Retries

### 8.63.1 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_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. |

### 8.63.2 Field Documentation

#### 8.63.2.1 BYTE appStatus::aidLength

#### 8.63.2.2 BYTE appStatus::aidVal[255]

#### 8.63.2.3 BYTE appStatus::appState

#### 8.63.2.4 BYTE appStatus::appType

#### 8.63.2.5 BYTE appStatus::persoFeature

#### 8.63.2.6 BYTE appStatus::persoRetries

#### 8.63.2.7 BYTE appStatus::persoState

#### 8.63.2.8 BYTE appStatus::persoUnblockRetries

#### 8.63.2.9 BYTE appStatus::pin1Retries

#### 8.63.2.10 BYTE appStatus::pin1State

#### 8.63.2.11 BYTE appStatus::pin2Retries

#### 8.63.2.12 BYTE appStatus::pin2State

#### 8.63.2.13 BYTE appStatus::puk1Retries

#### 8.63.2.14 BYTE appStatus::puk2Retries

#### 8.63.2.15 BYTE appStatus::univPin

## 8.64 arrAlertingPattern Struct Reference

### Data Fields

- BYTE numInstances

- BYTE callID [20]
- ULONG alertingPattern [20]

### 8.64.1 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[MAX_NO_OF_CALLS]* | • Array of Unique call identifier for the call. |
| *alertingPattern[MAX_NO_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 |

### 8.64.2 Field Documentation

#### 8.64.2.1 ULONG arrAlertingPattern::alertingPattern[20]

#### 8.64.2.2 BYTE arrAlertingPattern::callID[20]

#### 8.64.2.3 BYTE arrAlertingPattern::numInstances

## 8.65 arrAlertingType Struct Reference

### Data Fields

- BYTE numInstances
- BYTE callID [20]
- BYTE AlertingType [20]

### 8.65.1 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[MAX_NO-_OF_CALLS]* | • Array of Unique call identifier for the call. |
| *AlertingType[M-AX_NO_OF_C-ALLS]* | • Array of Alerting type.<br>– 0x00 - ALERTING_LOCAL - Local<br>– 0x01 - ALERTING_REMOTE - Remote |

### 8.65.2 Field Documentation

#### 8.65.2.1 BYTE arrAlertingType::AlertingType[20]

#### 8.65.2.2 BYTE arrAlertingType::callID[20]

#### 8.65.2.3 BYTE arrAlertingType::numInstances

## 8.66 arrAlphaID Struct Reference

### Data Fields

- BYTE numInstances
- allCallsAlphaIDInfo allCallsAlphaIDInfoArr [20]

### 8.66.1 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. |
| *allCallsAlphaID-Info[MAX_NO_-OF_CALLS]* | • Array of allCallsAlphaIDInfo.<br>• See allCallsAlphaIDInfo for more information. |

### 8.66.2 Field Documentation

#### 8.66.2.1 allCallsAlphaIDInfo arrAlphaID::allCallsAlphaIDInfoArr[20]

#### 8.66.2.2 BYTE arrAlphaID::numInstances

## 8.67 arrCalledPartyNum Struct Reference

### Data Fields

- BYTE numInstances
- peerNumberInfo CalledPartyNum [20]

### 8.67.1 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[MAX_NO_OF_CALLS]* | • Array of CalledPartyNum.<br>• See peerNumberInfo for more information. |

### 8.67.2 Field Documentation

#### 8.67.2.1 peerNumberInfo arrCalledPartyNum::CalledPartyNum[20]

#### 8.67.2.2 BYTE arrCalledPartyNum::numInstances

## 8.68 arrCallEndReason Struct Reference

### Data Fields

- BYTE numInstances
- BYTE callID [20]
- WORD callEndReason [20]

### 8.68.1 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[MAX_NO_OF_CALLS]* | • Array of Unique call identifier for the call. |
| *callEndReason[MAX_NO_OF_CALLS]* | • Array of Call End Reason .<br>• See Table9 qaGobiApiTableVoiceCallEndReasons.h for a list of valid voice-related call end reasons |

### 8.68.2 Field Documentation

#### 8.68.2.1 WORD arrCallEndReason::callEndReason[20]

#### 8.68.2.2 BYTE arrCallEndReason::callID[20]

#### 8.68.2.3 BYTE arrCallEndReason::numInstances

## 8.69 arrCallInfo Struct Reference

### Data Fields

- BYTE numInstances
- getAllCallInformation getAllCallInfo [20]

### 8.69.1 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[MAX_NO_OF_CALLS]* | • Array of CallInfo.<br>• See getAllCallInfo for more information. |

### 8.69.2 Field Documentation

#### 8.69.2.1 getAllCallInformation arrCallInfo::getAllCallInfo[20]

#### 8.69.2.2 BYTE arrCallInfo::numInstances

## 8.70 arrConnectPartyNum Struct Reference

### Data Fields

- BYTE numInstances
- peerNumberInfo ConnectedPartyNum [20]

### 8.70.1 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. |

| | |
|---|---|
| *ConnectedParty-Num[MAX_NO_-OF_CALLS]* | • Array of ConnectedPartyNum.<br>• See peerNumberInfo for more information. |

### 8.70.2 Field Documentation

#### 8.70.2.1 peerNumberInfo arrConnectPartyNum::ConnectedPartyNum[20]

#### 8.70.2.2 BYTE arrConnectPartyNum::numInstances

## 8.71 arrDiagInfo Struct Reference

### Data Fields

- BYTE numInstances
- allCallsDiagInfo DiagInfo [20]

### 8.71.1 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[MAX_-NO_OF_CALLS]* | • Array of DiagInfo.<br>• See allCallsDiagInfo for more information. |

### 8.71.2 Field Documentation

#### 8.71.2.1 allCallsDiagInfo arrDiagInfo::DiagInfo[20]

#### 8.71.2.2 BYTE arrDiagInfo::numInstances

## 8.72 arrRedirPartyNum Struct Reference

### Data Fields

- BYTE numInstances
- peerNumberInfo RedirPartyNum [20]

### 8.72.1 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[MAX_NO_OF_CALLS]* | • Array of RedirPartyNum.<br>• See peerNumberInfo for more information. |

### 8.72.2 Field Documentation

#### 8.72.2.1 BYTE arrRedirPartyNum::numInstances

#### 8.72.2.2 peerNumberInfo arrRedirPartyNum::RedirPartyNum[20]

## 8.73 arrRemotePartyName Struct Reference

### Data Fields

- BYTE numInstances
- getAllCallRmtPtyName GetAllCallRmtPtyName [20]

### 8.73.1 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. |
| *GetAllCallRmtPtyName[MAX_NO_OF_CALLS]* | • Array of remotePartyName.<br>• See getAllCallRmtPtyName for more information. |

### 8.73.2 Field Documentation

#### 8.73.2.1 getAllCallRmtPtyName arrRemotePartyName::GetAllCallRmtPtyName[20]

#### 8.73.2.2 BYTE arrRemotePartyName::numInstances

## 8.74 arrRemotePartyNum Struct Reference

### Data Fields

- BYTE numInstances
- getAllCallRmtPtyNum RmtPtyNum [20]

### 8.74.1 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[MAX_NO_OF_CALLS]* | • Array of remotePartyNum.<br>• See getAllCallRmtPtyNum for more information. |

### 8.74.2 Field Documentation

#### 8.74.2.1 BYTE arrRemotePartyNum::numInstances

#### 8.74.2.2 getAllCallRmtPtyNum arrRemotePartyNum::RmtPtyNum[20]

## 8.75 arrSvcOption Struct Reference

### Data Fields

- BYTE numInstances
- BYTE callID [20]
- WORD srvOption [20]

### 8.75.1 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[MAX_NO_OF_CALLS]* | • Array of Unique call identifier for the call. |
| *srvOption[MAX_NO_OF_CALLS]* | • Array of Service option.<br>• See Table9 qaGobiApiTableServiceOptions.h for standard service option number assignments. |

### 8.75.2 Field Documentation

#### 8.75.2.1 BYTE arrSvcOption::callID[20]

#### 8.75.2.2 BYTE arrSvcOption::numInstances

#### 8.75.2.3 WORD arrSvcOption::srvOption[20]

## 8.76 arrUUSInfo Struct Reference

### Data Fields

- BYTE numInstances
- allCallsUUSInfo AllCallsUUSInfo [20]

### 8.76.1 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[MAX_NO_OF_CALLS]* | • Array of allCallsUUSInfo.<br>• See allCallsUUSInfo for more information. |

### 8.76.2 Field Documentation

#### 8.76.2.1 allCallsUUSInfo arrUUSInfo::AllCallsUUSInfo[20]

#### 8.76.2.2 BYTE arrUUSInfo::numInstances

## 8.77 authenticateResult Struct Reference

### Data Fields

- WORD contentLen
- BYTE content [1024]

### 8.77.1 Detailed Description

This structure contains the information about the authenticate result.

**Parameters**

| | |
|---|---|
| *contentLen* | • Length of the following elements i.e. content. |
| *content[MAX_CONTENT_LENGTH]* | • Authenticate data.<br>• This value is a sequence of bytes returned from the card. |

### 8.77.2 Field Documentation

#### 8.77.2.1 BYTE authenticateResult::content[1024]

#### 8.77.2.2 WORD authenticateResult::contentLen

## 8.78 authenticationData Struct Reference

### Data Fields

- BYTE context
- WORD dataLen
- BYTE data [1024]

### 8.78.1 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[MAX_DESCRIPTION_LENGTH]* | • Authenticate Data. |

### 8.78.2 Field Documentation

#### 8.78.2.1 BYTE authenticationData::context

#### 8.78.2.2 BYTE authenticationData::data[1024]

#### 8.78.2.3 WORD authenticationData::dataLen

## 8.79 BandCapabilityResp Struct Reference

### Data Fields

- ULONGLONG bandCapability
- ULONGLONG * pLteBandCapability
- ULONGLONG * pTdsBandCapability

### 8.79.1 Detailed Description

This structure contains the TLV required to Get Band Capability.

**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 |

| | |
|---|---|
| *OUT]* | pLteBandCapability[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>• Bits 29 through 31 - Reserved<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 |

| | |
|---|---|
| *OUT]* | pTdsBandCapability[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 |

### 8.79.2 Field Documentation

#### 8.79.2.1 ULONGLONG BandCapabilityResp::bandCapability

#### 8.79.2.2 ULONGLONG* BandCapabilityResp::pLteBandCapability

#### 8.79.2.3 ULONGLONG* BandCapabilityResp::pTdsBandCapability

## 8.80 BdsSV Struct Reference

### Data Fields

- WORD id
- BYTE mask

### 8.80.1 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 |

### 8.80.2 Field Documentation

#### 8.80.2.1 WORD BdsSV::id

#### 8.80.2.2 BYTE BdsSV::mask

## 8.81 BdsSVInfo Struct Reference

### Data Fields

- BYTE len
- BdsSV * pSV

### 8.81.1 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 BdsSV. See BdsSV for more information |

### 8.81.2 Field Documentation

#### 8.81.2.1 BYTE BdsSVInfo::len

#### 8.81.2.2 BdsSV* BdsSVInfo::pSV

## 8.82 BroadcastConfig Struct Reference

### Data Fields

- WORD fromServiceId
- WORD toServiceId
- BYTE selected

### 8.82.1 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 |

### 8.82.2 Field Documentation

#### 8.82.2.1 WORD BroadcastConfig::fromServiceId

#### 8.82.2.2 BYTE BroadcastConfig::selected

#### 8.82.2.3 WORD BroadcastConfig::toServiceId

## 8.83 burstDTMFInfo Struct Reference

### Data Fields

- BYTE * pCallID
- BYTE digitCnt
- BYTE pDigitBuff [255]

### 8.83.1 Detailed Description

This structure contains Voice Burst DTMF Information

**Parameters**

| | |
|---|---|
| *pCallID[IN/OUT]* | • Call ID associated with call on which the DTMF information has to be sent. 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 |
| *digitCnt* | • Length of DTMF digit buffer which follows |
| *pDigitBuff[MAX_DESCRIPTION_LENGTH]* | • DTMF digit buffer in ASCII, NULL terminated |

### 8.83.2 Field Documentation

#### 8.83.2.1 BYTE burstDTMFInfo::digitCnt

#### 8.83.2.2 BYTE* burstDTMFInfo::pCallID

#### 8.83.2.3 BYTE burstDTMFInfo::pDigitBuff[255]

## 8.84 CallBarringSysInfo Struct Reference

### Data Fields

- ULONG csBarStatus
- ULONG psBarStatus

### 8.84.1 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 |

### 8.84.2 Field Documentation

#### 8.84.2.1 ULONG CallBarringSysInfo::csBarStatus

#### 8.84.2.2 ULONG CallBarringSysInfo::psBarStatus

## 8.85 callBarStatus Struct Reference

### Data Fields

- ULONG csBarStatus
- ULONG psBarStatus

### 8.85.1 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 |

### 8.85.2 Field Documentation

#### 8.85.2.1 ULONG callBarStatus::csBarStatus

#### 8.85.2.2 ULONG callBarStatus::psBarStatus

## 8.86 calledPartyInfo Struct Reference

### Data Fields

- BYTE PI
- BYTE SI
- BYTE numType
- BYTE numPlan
- BYTE numLen
- BYTE number [255]

### 8.86.1 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. |

### 8.86.2 Field Documentation

#### 8.86.2.1 BYTE calledPartyInfo::number[255]

#### 8.86.2.2 BYTE calledPartyInfo::numLen

#### 8.86.2.3 BYTE calledPartyInfo::numPlan

#### 8.86.2.4 BYTE calledPartyInfo::numType

#### 8.86.2.5 BYTE calledPartyInfo::PI

#### 8.86.2.6 BYTE calledPartyInfo::SI

## 8.87 calledPartySubAdd Struct Reference

### Data Fields

- BYTE extBit
- BYTE subAddrType
- BYTE oddEvenInd
- BYTE subAddrLen
- BYTE subAddr [255]

### 8.87.1 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_DESCRIPTION_LENGTH]* | • Array of the SubAddress in BCD number format. |

### 8.87.2 Field Documentation

#### 8.87.2.1 BYTE calledPartySubAdd::extBit

#### 8.87.2.2 BYTE calledPartySubAdd::oddEvenInd

#### 8.87.2.3 BYTE calledPartySubAdd::subAddr[255]

#### 8.87.2.4 BYTE calledPartySubAdd::subAddrLen

#### 8.87.2.5 BYTE calledPartySubAdd::subAddrType

## 8.88 callerIDInfo Struct Reference

### Data Fields

- BYTE PI
- BYTE callerIDLen
- BYTE callerID [255]

### 8.88.1 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. |

### 8.88.2 Field Documentation

#### 8.88.2.1 BYTE callerIDInfo::callerID[255]

#### 8.88.2.2 BYTE callerIDInfo::callerIDLen

#### 8.88.2.3 BYTE callerIDInfo::PI

## 8.89 callFwdTypeAndPlan Struct Reference

### Data Fields

- BYTE numberType
- BYTE numberPlan

### 8.89.1 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 |

### 8.89.2 Field Documentation

#### 8.89.2.1 BYTE callFwdTypeAndPlan::numberPlan

#### 8.89.2.2 BYTE callFwdTypeAndPlan::numberType

## 8.90 callFWExtInfo Struct Reference

### Data Fields

- BYTE SvcStatus
- BYTE SvcClass
- BYTE noReplyTimer
- BYTE PI
- BYTE SI
- BYTE numType
- BYTE numPlan
- BYTE numLen
- BYTE number [255]

### 8.90.1 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. |

### 8.90.2 Field Documentation

#### 8.90.2.1 BYTE callFWExtInfo::noReplyTimer

#### 8.90.2.2 BYTE callFWExtInfo::number[255]

#### 8.90.2.3 BYTE callFWExtInfo::numLen

#### 8.90.2.4 BYTE callFWExtInfo::numPlan

#### 8.90.2.5 BYTE callFWExtInfo::numType

#### 8.90.2.6 BYTE callFWExtInfo::PI

#### 8.90.2.7 BYTE callFWExtInfo::SI

#### 8.90.2.8 BYTE callFWExtInfo::SvcClass

#### 8.90.2.9 BYTE callFWExtInfo::SvcStatus

## 8.91 callFWInfo Struct Reference

### Data Fields

- BYTE SvcStatus
- BYTE SvcClass
- BYTE numLen
- BYTE number [255]
- BYTE noReplyTimer

### 8.91.1 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. |

### 8.91.2 Field Documentation

#### 8.91.2.1 BYTE callFWInfo::noReplyTimer

#### 8.91.2.2 BYTE callFWInfo::number[255]

#### 8.91.2.3 BYTE callFWInfo::numLen

#### 8.91.2.4 BYTE callFWInfo::SvcClass

#### 8.91.2.5 BYTE callFWInfo::SvcStatus

## 8.92 callInfo Struct Reference

### Data Fields

- BYTE callID
- BYTE callState
- BYTE callType
- BYTE direction
- BYTE mode

### 8.92.1 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 |

## 8.92.2 Field Documentation

### 8.92.2.1 BYTE callInfo::callID

#### 8.92.2.2 BYTE callInfo::callState

#### 8.92.2.3 BYTE callInfo::callType

#### 8.92.2.4 BYTE callInfo::direction

#### 8.92.2.5 BYTE callInfo::mode

## 8.93 callingPartyInfo Struct Reference

### Data Fields

- BYTE PI
- BYTE SI
- BYTE numType
- BYTE numPlan
- BYTE numLen
- BYTE number [255]

### 8.93.1 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. |

### 8.93.2 Field Documentation

#### 8.93.2.1 BYTE callingPartyInfo::number[255]

#### 8.93.2.2 BYTE callingPartyInfo::numLen

#### 8.93.2.3 BYTE callingPartyInfo::numPlan

#### 8.93.2.4 BYTE callingPartyInfo::numType

#### 8.93.2.5 BYTE callingPartyInfo::PI

#### 8.93.2.6 BYTE callingPartyInfo::SI

## 8.94 cardResult Struct Reference

### Data Fields

- BYTE sw1

- BYTE sw2

### 8.94.1 Detailed Description

This structure contains the information about the card result.

**Parameters**

| | |
|---|---|
| *sw1* | • SW1 received from the card. |
| *sw2* | • SW2 received from the card. |

### 8.94.2 Field Documentation

#### 8.94.2.1 BYTE cardResult::sw1

#### 8.94.2.2 BYTE cardResult::sw2

## 8.95 cardStatus Struct Reference

### Data Fields

- WORD indexGwPri
- WORD index1xPri
- WORD indexGwSec
- WORD index1xSec
- BYTE numSlot
- slotInfo SlotInfo [5]

### 8.95.1 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[MAX_NO_OF_SLOTS]* | • See slotInfo for more information. |

### 8.95.2 Field Documentation

#### 8.95.2.1 WORD cardStatus::index1xPri

#### 8.95.2.2 WORD cardStatus::index1xSec

#### 8.95.2.3 WORD cardStatus::indexGwPri

#### 8.95.2.4 WORD cardStatus::indexGwSec

#### 8.95.2.5 BYTE cardStatus::numSlot

#### 8.95.2.6 slotInfo cardStatus::SlotInfo[5]

## 8.96 CarrierImage_t Struct Reference

### Data Fields

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

### 8.96.1 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 |

### 8.96.2 Field Documentation

#### 8.96.2.1 uint8_t CarrierImage_t::m_FwBuildId[32]

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

#### 8.96.2.3 uint32_t CarrierImage_t::m_nCarrierId

#### 8.96.2.4 uint32_t CarrierImage_t::m_nFolderId

#### 8.96.2.5 uint32_t CarrierImage_t::m_nStorage

#### 8.96.2.6 uint8_t CarrierImage_t::m_PriBuildId[32]

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

## 8.97 CatAIPhaIdentifierTlv Struct Reference

### Data Fields

- BYTE ReferenceID
- USHORT AlphaIDLength
- BYTE AlphaID [255]

### 8.97.1 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] |

### 8.97.2 Field Documentation

#### 8.97.2.1 BYTE CatAlPhaIdentifierTlv::AlphaID[255]

#### 8.97.2.2 USHORT CatAlPhaIdentifierTlv::AlphaIDLength

#### 8.97.2.3 BYTE CatAlPhaIdentifierTlv::ReferenceID

## 8.98 CatCommonEventTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- BYTE EventID
- WORD EventLength
- union currentCatEvent CatEvent

### 8.98.1 Field Documentation

#### 8.98.1.1 union currentCatEvent CatCommonEventTlv::CatEvent

#### 8.98.1.2 BYTE CatCommonEventTlv::EventID

#### 8.98.1.3 WORD CatCommonEventTlv::EventLength

#### 8.98.1.4 BYTE CatCommonEventTlv::TlvPresent

## 8.99 CatEndProactiveSessionTlv Struct Reference

### Data Fields

- BYTE EndProactiveSession

### 8.99.1 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 |

### 8.99.2 Field Documentation

#### 8.99.2.1 BYTE CatEndProactiveSessionTlv::EndProactiveSession

## 8.100 CATEventDataType Struct Reference

### Data Fields

- ULONG eventMask
- ULONG * pErrorMask

### 8.100.1 Field Documentation

#### 8.100.1.1 ULONG CATEventDataType::eventMask

#### 8.100.1.2 ULONG* CATEventDataType::pErrorMask

## 8.101 CatEventIDDataTlv Struct Reference

### Data Fields

- ULONG ReferenceID
- USHORT DataLength
- BYTE Data [255]

### 8.101.1 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] |

### 8.101.2 Field Documentation

#### 8.101.2.1 BYTE CatEventIDDataTlv::Data[255]

#### 8.101.2.2 USHORT CatEventIDDataTlv::DataLength

#### 8.101.2.3 ULONG CatEventIDDataTlv::ReferenceID

## 8.102 CatEventListTlv Struct Reference

### Data Fields

- ULONG SetupEventList

### 8.102.1 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 |

### 8.102.2 Field Documentation

#### 8.102.2.1 ULONG CatEventListTlv::SetupEventList

## 8.103 CatRefreshTlv Struct Reference

### Data Fields

- USHORT RefreshMode
- BYTE RefreshStage

### 8.103.1 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 |

### 8.103.2 Field Documentation

#### 8.103.2.1 USHORT CatRefreshTlv::RefreshMode

#### 8.103.2.2 BYTE CatRefreshTlv::RefreshStage

## 8.104 ccSUPSType Struct Reference

### Data Fields

- BYTE svcType
- BYTE reason

### 8.104.1 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. |

### 8.104.2 Field Documentation

#### 8.104.2.1 BYTE ccSUPSType::reason

#### 8.104.2.2 BYTE ccSUPSType::svcType

## 8.105 CDMABroadcastConfig Struct Reference

### Data Fields

- WORD serviceCategory
- WORD language
- BYTE selected

### 8.105.1 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 |

### 8.105.2 Field Documentation

#### 8.105.2.1 WORD CDMABroadcastConfig::language

#### 8.105.2.2 BYTE CDMABroadcastConfig::selected

#### 8.105.2.3 WORD CDMABroadcastConfig::serviceCategory

## 8.106 CDMAChannel Struct Reference

### Data Fields

- WORD priChA
- WORD priChB
- WORD secChA
- WORD secChB

### 8.106.1 Detailed Description

This structure contains the parameters for CDMA Channel Information

**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 |

### 8.106.2 Field Documentation

#### 8.106.2.1 WORD CDMAChannel::priChA

#### 8.106.2.2 WORD CDMAChannel::priChB

#### 8.106.2.3 WORD CDMAChannel::secChA

#### 8.106.2.4 WORD CDMAChannel::secChB

## 8.107 CDMAECIOThresh Struct Reference

### Data Fields

- BYTE CDMAECIOThreshListLen
- WORD * pCDMAECIOThreshList

### 8.107.1 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) |

### 8.107.2 Field Documentation

#### 8.107.2.1 BYTE CDMAECIOThresh::CDMAECIOThreshListLen

#### 8.107.2.2 WORD* CDMAECIOThresh::pCDMAECIOThreshList

## 8.108 CDMAInfo Struct Reference

### Data Fields

- WORD sid
- WORD nid
- WORD baseId
- WORD refpn
- ULONG baseLat
- ULONG baseLong

### 8.108.1 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 |

### 8.108.2 Field Documentation

#### 8.108.2.1 WORD CDMAInfo::baseId

#### 8.108.2.2 ULONG CDMAInfo::baseLat

#### 8.108.2.3 ULONG CDMAInfo::baseLong

#### 8.108.2.4 WORD CDMAInfo::nid

#### 8.108.2.5 WORD CDMAInfo::refpn

#### 8.108.2.6 WORD CDMAInfo::sid

## 8.109 cdmaMsgDecodingParams Struct Reference

### Data Fields

- ULONG messageLength
- BYTE * pMessage
- ULONG * pMessageID
- BYTE * pSenderAddrLength
- CHAR * pSenderAddr
- BYTE * pTextMsgLength
- WORD * pTextMsg
- BYTE * pPriority
- BYTE * pPrivacy
- BYTE * pLanguage
- BYTE mcTimeStamp [0x08]
- BYTE absoluteValidity [0x08]
- BYTE * pRelativeValidity

- BYTE * pDisplayMode
- BOOL * pUserAcknowledgementReq
- BOOL * pReadAcknowledgementReq
- BYTE * pAlertPriority
- BYTE * pCallbkAddrLength
- CHAR * pCallbkAddr

### 8.109.1 Detailed Description

Structure contains parameters which need to be decoded from message

**Parameters**

| | |
|---|---|
| *messageLength[IN]* | • Length of the message to be decoded in bytes |
| *pMessage[IN]* | • Message read off the device via GetSMS |
| *pSenderAddrLength[IN/OUT]* | • Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. Note that a length of 14 is reasonable.Upon successful output, returns the length of originating address string (including the NULL termination) |
| *pSenderAddr[OUT]* | • Returns NULL-terminated ASCII String containing the originating address.International number will be prepended with a '+'character |
| *pTextMsgLength[IN/OUT]* | • Upon input, specifies the number of UCS2 characters the given text message buffer can accommodate.Upon successful output, returns the number of UCS2 characters returns in the given text messagebuffer(including NULL-terminator) |
| *pTextMsg[OUT]* | • Returns the text message as NULL-terminated UCS2 string |
| *pPriority[OUT]* | (optional parameter)<br>• Returns the priority setting of the message 0x00 - normal 0x01 - interactive 0x02 - urgent 0x03 - emergency 0xFF - unavailable setting |
| *pPrivacy[OUT](optional* | parameter)<br>• Returns the privacy setting of the message 0x00 - not restricted 0x01 - restricted 0x02 - confidential 0x03 - secret 0xFF - unavailable setting |
| *pLanguage[OUT]* | (optional parameter )<br>• Returns the language setting of the message 0x00 - unspecified 0x01 - english 0x02 - french 0x03 - spanish 0x04 - japanese 0x05 - korean 0x06 - chinese 0x07 - hebrew 0xFF - unavailable setting |
| *mcTimeStamp[8][OUT]* | (optional parameter)<br>• Returns the message center timestamp which takes the form: YYMMDDHHMMSSTZ where YY - year MM - month DD - day HH - hour MM - minute SS - second TZ - timezone All values are in decimal. Timezone is in relation to GMT, one unit is equal to 15 minutes and MSB indicates a negative value.If this information is unavailable for message then this field will be filled with 0xFF |

| | |
|---|---|
| *absoluteValidity[8][OUT]* | (optional parameter)<br>• Returns the absolute validity period setting for this message.This field takes the same form as mcTimeStamp |
| *pRelativeValidity[OUT]* | (optional parameter)<br>• Returns the relative validity period.Values have the following meanings: 0 to 143: validity period =(value + 1)∗ 5 minutes 144 to 167: validity period =12 hours+(value - 143)∗30 minutes 168 to 196: validity period = (value - 166) ∗ 1 day 197 to 244: validity period = (value - 192) ∗ 1 week 245: validity period = indefinite 246: validity period = immediate 247: validity period = valid until mobile becomes inactive 248: validity period = valid until registration area changes 249 to 254: reserved 255: unavailable information |
| *pDisplayMode[OUT]* | (optional parameter)<br>• Returns the display mode parameter 0x00 - immediate display 0x01 - mobile default setting 0x02 - user invoked 0x03 - reserved 0xFF - unavailable parameter |
| *pUserAcknowledgementReq[OUT]* | (optional parameter)<br>• Returns the user (manual) acknowledgment request parameter TRUE - means the user is requested to manually acknowledge the delivery of the message. FALSE - means no such user acknowledgement is requested |
| *pReadAcknowledgementReq[OUT]* | (optional parameter)<br>• Returns the read acknowledgement request parameter TRUE - means acknowledgment of the message being viewed is requested. FALSE - means no such read acknowledgement is requested |
| *pAlertPriority[OUT]* | (optional parameter)<br>• Returns the alerting parameter setting 0x00 - use default alert 0x01 - use low priority alert 0x02 - use medium priority alert 0x03 - use high priority alert 0xFF - unavailable parameter |
| *pCallbkAddrLength[OUT]* | (optional parameter)<br>• returns the length of Callback address string (including the NULL termination) |
| *pCallbkAddr[OUT]* | (optional parameter)<br>• returns NULL-terminated ASCII String containing callback address String containing the Call Back number with a 32 maximum characters. |

### 8.109.2 Field Documentation

#### 8.109.2.1 BYTE cdmaMsgDecodingParams::absoluteValidity[0x08]

#### 8.109.2.2 BYTE cdmaMsgDecodingParams::mcTimeStamp[0x08]

#### 8.109.2.3 ULONG cdmaMsgDecodingParams::messageLength

#### 8.109.2.4 BYTE∗ cdmaMsgDecodingParams::pAlertPriority

#### 8.109.2.5 CHAR∗ cdmaMsgDecodingParams::pCallbkAddr

#### 8.109.2.6 BYTE∗ cdmaMsgDecodingParams::pCallbkAddrLength

#### 8.109.2.7 BYTE∗ cdmaMsgDecodingParams::pDisplayMode

#### 8.109.2.8 BYTE* cdmaMsgDecodingParams::pLanguage

#### 8.109.2.9 BYTE* cdmaMsgDecodingParams::pMessage

#### 8.109.2.10 ULONG* cdmaMsgDecodingParams::pMessageID

#### 8.109.2.11 BYTE* cdmaMsgDecodingParams::pPriority

#### 8.109.2.12 BYTE* cdmaMsgDecodingParams::pPrivacy

#### 8.109.2.13 BOOL* cdmaMsgDecodingParams::pReadAcknowledgementReq

#### 8.109.2.14 BYTE* cdmaMsgDecodingParams::pRelativeValidity

#### 8.109.2.15 CHAR* cdmaMsgDecodingParams::pSenderAddr

#### 8.109.2.16 BYTE* cdmaMsgDecodingParams::pSenderAddrLength

#### 8.109.2.17 WORD* cdmaMsgDecodingParams::pTextMsg

#### 8.109.2.18 BYTE* cdmaMsgDecodingParams::pTextMsgLength

#### 8.109.2.19 BOOL* cdmaMsgDecodingParams::pUserAcknowledgementReq

## 8.110 cdmaMsgDecodingParamsExt Struct Reference

### Data Fields

- ULONG messageLength
- BYTE * pMessage
- ULONG * pMessageID
- BYTE * pSenderAddrLength
- CHAR * pSenderAddr
- BYTE * pTextMsgLength
- WORD * pTextMsg
- BYTE * pPriority
- BYTE * pPrivacy
- BYTE * pLanguage
- BYTE mcTimeStamp [0x08]
- BYTE absoluteValidity [0x08]
- BYTE * pRelativeValidity
- BYTE * pDisplayMode
- BOOL * pUserAcknowledgementReq
- BOOL * pReadAcknowledgementReq
- BYTE * pAlertPriority
- BYTE * pCallbkAddrLength
- CHAR * pCallbkAddr
- BYTE * pReferenceNum
- BYTE * pTotalNum
- BYTE * pPartNum
- BOOL * pIsUDHPresent

### 8.110.1 Detailed Description

Structure contains parameters which need to be decoded from message

**Parameters**

| | |
|---|---|
| *messageLength[IN]* | • Length of the message to be decoded in bytes |
| *pMessage[IN]* | • Message read off the device via GetSMS |
| *pSenderAddrLength[IN/OUT]* | • Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. Note that a length of 14 is reasonable.Upon successful output, returns the length of originating address string (including the NULL termination) |
| *pSenderAddr[OUT]* | • Returns NULL-terminated ASCII String containing the originating address.International number will be prepended with a '+'character |
| *pTextMsgLength[IN/OUT]* | • Upon input, specifies the number of UCS2 characters the given text message buffer can accommodate.Upon successful output, returns the number of UCS2 characters returns in the given text messagebuffer(including NULL-terminator) |
| *pTextMsg[OUT]* | • Returns the text message as NULL-terminated UCS2 string |
| *pPriority[OUT]* | (optional parameter)<br>• Returns the priority setting of the message 0x00 - normal 0x01 - interactive 0x02 - urgent 0x03 - emergency 0xFF - unavailable setting |
| *pPrivacy[OUT](optional* | parameter)<br>• Returns the privacy setting of the message 0x00 - not restricted 0x01 - restricted 0x02 - confidential 0x03 - secret 0xFF - unavailable setting |
| *pLanguage[OUT]* | (optional parameter )<br>• Returns the language setting of the message 0x00 - unspecified 0x01 - english 0x02 - french 0x03 - spanish 0x04 - japanese 0x05 - korean 0x06 - chinese 0x07 - hebrew 0xFF - unavailable setting |
| *mcTimeStamp[8][OUT]* | (optional parameter)<br>• Returns the message center timestamp which takes the form: YYMMDDHHMMSSTZ where YY - year MM - month DD - day HH - hour MM - minute SS - second TZ - timezone All values are in decimal. Timezone is in relation to GMT, one unit is equal to 15 minutes and MSB indicates a negative value.If this information is unavailable for message then this field will be filled with 0xFF |
| *absoluteValidity[8][OUT]* | (optional parameter)<br>• Returns the absolute validity period setting for this message.This field takes the same form as mcTimeStamp |

| | |
|---|---|
| *pRelative-Validity[OUT]* | (optional parameter)<br>• Returns the relative validity period.Values have the following meanings: 0 to 143: validity period =(value + 1)∗ 5 minutes 144 to 167: validity period =12 hours+(value - 143)∗30 minutes 168 to 196: validity period = (value - 166) ∗ 1 day 197 to 244: validity period = (value - 192) ∗ 1 week 245: validity period = indefinite 246: validity period = immediate 247: validity period = valid until mobile becomes inactive 248: validity period = valid until registration area changes 249 to 254: reserved 255: unavailable information |
| *pDisplayMode[-OUT]* | (optional parameter)<br>• Returns the display mode parameter 0x00 - immediate display 0x01 - mobile default setting 0x02 - user invoked 0x03 - reserved 0xFF - unavailable parameter |
| *pUser-Acknowledgement-Req[OUT]* | (optional parameter)<br>• Returns the user (manual) acknowledgment request parameter TRUE - means the user is requested to manually acknowledge the delivery of the message. FALSE - means no such user acknowledgement is requested |
| *pRead-Acknowledgement-Req[OUT]* | (optional parameter)<br>• Returns the read acknowledgement request parameter TRUE - means acknowledgment of the message being viewed is requested. FALSE - means no such read acknowledgement is requested |
| *pAlertPriority[O-UT]* | (optional parameter)<br>• Returns the alerting parameter setting 0x00 - use default alert 0x01 - use low priority alert 0x02 - use medium priority alert 0x03 - use high priority alert 0xFF - unavailable parameter |
| *pCallbkAddr-Length[OUT]* | (optional parameter)<br>• returns the length of Callback address string (including the NULL termination) |
| *pCallbkAddr[OU-T]* | (optional parameter)<br>• returns NULL-terminated ASCII String containing callback address String containing the Call Back number with a 32 maximum characters. |
| *pReference-Num[OUT]* | • Reference number of the sms |
| *pTotalNum[OUT]* | • Total number of the concatenated message |
| *pPartNum[OUT]* | • Sequence number of the current message |
| *pIsUDHPresent* | • Is User Data Header Present in the PDU? If yes, it means it is a<br>• concatenated SMS. |

### 8.110.2 Field Documentation

#### 8.110.2.1 BYTE cdmaMsgDecodingParamsExt::absoluteValidity[0x08]

#### 8.110.2.2 BYTE cdmaMsgDecodingParamsExt::mcTimeStamp[0x08]

#### 8.110.2.3 ULONG cdmaMsgDecodingParamsExt::messageLength

#### 8.110.2.4 BYTE* cdmaMsgDecodingParamsExt::pAlertPriority

#### 8.110.2.5 CHAR* cdmaMsgDecodingParamsExt::pCallbkAddr

#### 8.110.2.6 BYTE* cdmaMsgDecodingParamsExt::pCallbkAddrLength

#### 8.110.2.7 BYTE* cdmaMsgDecodingParamsExt::pDisplayMode

#### 8.110.2.8 BOOL* cdmaMsgDecodingParamsExt::pIsUDHPresent

#### 8.110.2.9 BYTE* cdmaMsgDecodingParamsExt::pLanguage

#### 8.110.2.10 BYTE* cdmaMsgDecodingParamsExt::pMessage

#### 8.110.2.11 ULONG* cdmaMsgDecodingParamsExt::pMessageID

#### 8.110.2.12 BYTE* cdmaMsgDecodingParamsExt::pPartNum

#### 8.110.2.13 BYTE* cdmaMsgDecodingParamsExt::pPriority

#### 8.110.2.14 BYTE* cdmaMsgDecodingParamsExt::pPrivacy

#### 8.110.2.15 BOOL* cdmaMsgDecodingParamsExt::pReadAcknowledgementReq

#### 8.110.2.16 BYTE* cdmaMsgDecodingParamsExt::pReferenceNum

#### 8.110.2.17 BYTE* cdmaMsgDecodingParamsExt::pRelativeValidity

#### 8.110.2.18 CHAR* cdmaMsgDecodingParamsExt::pSenderAddr

#### 8.110.2.19 BYTE* cdmaMsgDecodingParamsExt::pSenderAddrLength

#### 8.110.2.20 WORD* cdmaMsgDecodingParamsExt::pTextMsg

#### 8.110.2.21 BYTE* cdmaMsgDecodingParamsExt::pTextMsgLength

#### 8.110.2.22 BYTE* cdmaMsgDecodingParamsExt::pTotalNum

#### 8.110.2.23 BOOL* cdmaMsgDecodingParamsExt::pUserAcknowledgementReq

## 8.111 cdmaMsgEncodingParams Struct Reference

### Data Fields

- BYTE * pMessageSize
- BYTE * pMessage
- BYTE messageId
- CHAR * pDestAddr
- CHAR * pCallbackAddr
- ULONG textMsgLength
- WORD * pTextMsg
- BYTE * pPriority
- BYTE * pEncodingAlphabet
- BYTE * pRelValidity

### 8.111.1 Detailed Description

Structure contains parameters for message to be encoded

**Parameters**

| | |
|---|---|
| *pMessageSize[IN/OUT]* | • Upon input, specifies the total number of bytes that the given pMessage buffer can hold (a buffer of length 240 is recommended).Upon successful output, specifies the length of the constructed message placed in the pMessage buffer (in bytes) |
| *pMessage[OUT]* | - The constructed raw message |
| *messageId[IN]* | • The message reference number for this message. This value should be incremented for every message the host application sends |
| *pDestAddr[IN]* | • Gives NULL-terminated ASCII String containing a destination address.International number will be prepended with a '+' character |
| *pCallbackAddr[IN]* | • Gives NULL-terminated ASCII String containing a callback address.International number will be prepended with a '+' character |
| *textMsgLength[IN]* | • Number of UCS2 characters in the text message(excluding NULL) |
| *pTextMsg[IN]* | • Text message to be encoded |
| *pPriority[IN](optional* | parameter)<br>• Gives the priority of the outgoing message: 0 - normal (default if NULL pointer is given) 1 - interactive 2 - urgent 3 - emergency 64 - 64 is decoded value for URGENT VZAM Support interactive. 128 - 128 is decoded value for URGENT VZAM Support urgent. 192 - 128 is decoded value for URGENT VZAM Support emergency. |
| *pEncodingAlphabet[IN/OUT](optional* | parameter)<br>• Upon input, specifies the alphabet the text message should be encoded in 0 - 8bit ASCII (not supported at this time) 1 - IS91EP (not supported at this time) 2 - 7bit ASCII (default if NULL pointer is given) 3 - IA5 (not supported at this time) 4 - unicode (not supported at this time) 5 - shift JIS (not supported at this time) 6 - korean (not supported at this time) 7 - latin hebrew (not supported at this time) 8 - latin (not supported at this time) 9 - GSM 7 bit default Upon successful output, specifies the alphabet used to encode the message. |
| *pRelValidity[IN](optional* | parameter)<br>• Gives the relative validity period of the outgoing message 0 - Set Relative validity to 11 1 - Set Relative validity to 71 2 - Set Relative validity to 167 3 - Set Relative validity to 169 4 - Set Relative validity to 171 Values have the following meanings: 0 to 143: validity period = (value + 1)∗ 5 minutes 144 to 167: validity period = 12 hours + (value - 143)∗30 minutes 168 to 196: validity period = (value - 166) ∗ 1 day 197 to 244: validity period = (value - 192) ∗ 1 week 245: validity period = indefinite |

Currently only encoding of 7bit ASCII messages is supported.

### 8.111.2 Field Documentation

#### 8.111.2.1 BYTE cdmaMsgEncodingParams::messageId

#### 8.111.2.2 CHAR* cdmaMsgEncodingParams::pCallbackAddr

#### 8.111.2.3 CHAR* cdmaMsgEncodingParams::pDestAddr

#### 8.111.2.4 BYTE* cdmaMsgEncodingParams::pEncodingAlphabet

#### 8.111.2.5 BYTE* cdmaMsgEncodingParams::pMessage

#### 8.111.2.6 BYTE* cdmaMsgEncodingParams::pMessageSize

#### 8.111.2.7 BYTE* cdmaMsgEncodingParams::pPriority

#### 8.111.2.8 BYTE* cdmaMsgEncodingParams::pRelValidity

#### 8.111.2.9 WORD* cdmaMsgEncodingParams::pTextMsg

#### 8.111.2.10 ULONG cdmaMsgEncodingParams::textMsgLength

## 8.112 CDMARSSIThresh Struct Reference

### Data Fields

- BYTE CDMARSSIThreshListLen
- WORD * pCDMARSSIThreshList

### 8.112.1 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). |

### 8.112.2 Field Documentation

#### 8.112.2.1 BYTE CDMARSSIThresh::CDMARSSIThreshListLen

#### 8.112.2.2 WORD* CDMARSSIThresh::pCDMARSSIThreshList

## 8.113 CDMASSInfo Struct Reference

### Data Fields

- INT8 rssi
- SHORT ecio

### 8.113.1 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). |

### 8.113.2 Field Documentation

#### 8.113.2.1 SHORT CDMASSInfo::ecio

#### 8.113.2.2 INT8 CDMASSInfo::rssi

## 8.114 cdmaSSInfo Struct Reference

### Data Fields

- int8_t rssi
- int16_t ecio

### 8.114.1 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). |

### 8.114.2 Field Documentation

#### 8.114.2.1 int16_t cdmaSSInfo::ecio

#### 8.114.2.2 int8_t cdmaSSInfo::rssi

## 8.115 CDMASysInfo Struct Reference

### Data Fields

- sysInfoCommon sysInfoCDMA
- BYTE isSysPrlMatchValid
- BYTE isSysPrlMatch
- BYTE pRevInUseValid
- BYTE pRevInUse
- BYTE bsPRevValid
- BYTE bsPRev
- BYTE ccsSupportedValid
- BYTE ccsSupported
- BYTE cdmaSysIdValid
- WORD systemID
- WORD networkID
- BYTE bsInfoValid
- WORD baseId
- ULONG baseLat
- ULONG baseLong
- BYTE packetZoneValid
- WORD packetZone
- BYTE networkIdValid
- BYTE MCC [3]
- BYTE MNC [3]

### 8.115.1 Detailed Description

Structure for storing the CDMA System Information.

**Parameters**

| | |
|---|---|
| *sysInfoCDMA* | • See 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[PLMN_LENGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters<br>• MCC wildcard value is returned as {'3', 0xFF, 0xFF}. |
| *MNC[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}. |

## 8.115.2 Field Documentation

### 8.115.2.1 WORD CDMASysInfo::baseId

#### 8.115.2.2 ULONG CDMASysInfo::baseLat

#### 8.115.2.3 ULONG CDMASysInfo::baseLong

#### 8.115.2.4 BYTE CDMASysInfo::bsInfoValid

#### 8.115.2.5 BYTE CDMASysInfo::bsPRev

#### 8.115.2.6 BYTE CDMASysInfo::bsPRevValid

#### 8.115.2.7 BYTE CDMASysInfo::ccsSupported

#### 8.115.2.8 BYTE CDMASysInfo::ccsSupportedValid

#### 8.115.2.9 BYTE CDMASysInfo::cdmaSysIdValid

#### 8.115.2.10 BYTE CDMASysInfo::isSysPrlMatch

#### 8.115.2.11 BYTE CDMASysInfo::isSysPrlMatchValid

#### 8.115.2.12 BYTE CDMASysInfo::MCC[3]

#### 8.115.2.13 BYTE CDMASysInfo::MNC[3]

#### 8.115.2.14 WORD CDMASysInfo::networkID

#### 8.115.2.15 BYTE CDMASysInfo::networkIdValid

#### 8.115.2.16 WORD CDMASysInfo::packetZone

#### 8.115.2.17 BYTE CDMASysInfo::packetZoneValid

#### 8.115.2.18 BYTE CDMASysInfo::pRevInUse

#### 8.115.2.19 BYTE CDMASysInfo::pRevInUseValid

#### 8.115.2.20 sysInfoCommon CDMASysInfo::sysInfoCDMA

#### 8.115.2.21 WORD CDMASysInfo::systemID

## 8.116 CDMASysInfoExt Struct Reference

### Data Fields

- WORD MCC
- BYTE imsi_11_12

### 8.116.1 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 |

### 8.116.2 Field Documentation

#### 8.116.2.1 BYTE CDMASysInfoExt::imsi_11_12

#### 8.116.2.2 WORD CDMASysInfoExt::MCC

## 8.117 CellDb Struct Reference

### Data Fields

- ULONG mask

### 8.117.1 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 |

### 8.117.2 Field Documentation

#### 8.117.2.1 ULONG CellDb::mask

## 8.118 cellParams Struct Reference

### Data Fields

- WORD pci
- SHORT rsrq

- SHORT rsrp
- SHORT rssi
- SHORT srxlev

### 8.118.1 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. |

### 8.118.2 Field Documentation

#### 8.118.2.1 WORD cellParams::pci

#### 8.118.2.2 SHORT cellParams::rsrp

#### 8.118.2.3 SHORT cellParams::rsrq

#### 8.118.2.4 SHORT cellParams::rssi

#### 8.118.2.5 SHORT cellParams::srxlev

## 8.119 changeUIMPIN Struct Reference

### Data Fields

- BYTE pinID
- BYTE oldPINLen
- BYTE oldPINVal [255]
- BYTE pinLen
- BYTE pinVal [255]

### 8.119.1 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. |

### 8.119.2 Field Documentation

#### 8.119.2.1 BYTE changeUIMPIN::oldPINLen

#### 8.119.2.2 BYTE changeUIMPIN::oldPINVal[255]

#### 8.119.2.3 BYTE changeUIMPIN::pinID

#### 8.119.2.4 BYTE changeUIMPIN::pinLen

#### 8.119.2.5 BYTE changeUIMPIN::pinVal[255]

## 8.120 ChannelRate Struct Reference

### Data Fields

- ULONG CurrChanTxRate
- ULONG CurrChanRxRate
- ULONG MaxChanTxRate
- ULONG MaxChanRxRate

### 8.120.1 Detailed Description

This structure contains Channel Rate

**Parameters**

| | |
|---|---|
| *CurrChanTxRate* | • Instantaneous channel Tx rate in bits per second |
| *CurrChanRx-Rate* | • Instantaneous channel Rx rate in bits per second |
| *MaxChanTxRate* | • maximum Tx rate that can be assigned to the device by the serving system in bits per second |
| *MaxChanRx-Rate* | • maximum Rx rate that can be assigned to the device by the serving system in bits per second |

### 8.120.2 Field Documentation

#### 8.120.2.1 ULONG ChannelRate::CurrChanRxRate

#### 8.120.2.2 ULONG ChannelRate::CurrChanTxRate

#### 8.120.2.3 ULONG ChannelRate::MaxChanRxRate

#### 8.120.2.4 ULONG ChannelRate::MaxChanTxRate

## 8.121 channelRate Struct Reference

### Data Fields

- ULONG CurrChanTxRate
- ULONG CurrChanRxRate

### 8.121.1 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 |

### 8.121.2 Field Documentation

#### 8.121.2.1 ULONG channelRate::CurrChanRxRate

#### 8.121.2.2 ULONG channelRate::CurrChanTxRate

## 8.122 ciotAcqOrderPref Struct Reference

### Data Fields

- BYTE ciotAcqOrderLen
- ULONG * pCiotAcqOrder

#### 8.122.1 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 |

#### 8.122.2 Field Documentation

##### 8.122.2.1 BYTE ciotAcqOrderPref::ciotAcqOrderLen

##### 8.122.2.2 ULONG* ciotAcqOrderPref::pCiotAcqOrder

## 8.123 CLIPResp Struct Reference

### Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

#### 8.123.1 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 |

### 8.123.2 Field Documentation

#### 8.123.2.1 BYTE CLIPResp::ActiveStatus

#### 8.123.2.2 BYTE CLIPResp::ProvisionStatus

## 8.124 CLIRResp Struct Reference

### Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

### 8.124.1 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 |

### 8.124.2 Field Documentation

#### 8.124.2.1 BYTE CLIRResp::ActiveStatus

#### 8.124.2.2 BYTE CLIRResp::ProvisionStatus

## 8.125 ClkInfo Struct Reference

### Data Fields

- ULONG mask

### 8.125.1 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 |

### 8.125.2 Field Documentation

#### 8.125.2.1 ULONG ClkInfo::mask

## 8.126 CNAPResp Struct Reference

### Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

### 8.126.1 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 |

### 8.126.2 Field Documentation

#### 8.126.2.1 BYTE CNAPResp::ActiveStatus

#### 8.126.2.2 BYTE CNAPResp::ProvisionStatus

## 8.127 COLPResp Struct Reference

### Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

### 8.127.1 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 |

### 8.127.2 Field Documentation

#### 8.127.2.1 BYTE COLPResp::ActiveStatus

#### 8.127.2.2 BYTE COLPResp::ProvisionStatus

## 8.128 COLRResp Struct Reference

### Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

### 8.128.1 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 |

### 8.128.2 Field Documentation

#### 8.128.2.1 BYTE COLRResp::ActiveStatus

#### 8.128.2.2 BYTE COLRResp::ProvisionStatus

## 8.129 CommInfo Struct Reference

### Data Fields

- BYTE temperature
- BYTE modemMode
- BYTE systemMode
- BYTE imsRegState
- BYTE psState

### 8.129.1 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 |

### 8.129.2 Field Documentation

#### 8.129.2.1 BYTE CommInfo::imsRegState

#### 8.129.2.2 BYTE CommInfo::modemMode

#### 8.129.2.3 BYTE CommInfo::psState

#### 8.129.2.4 BYTE CommInfo::systemMode

#### 8.129.2.5 BYTE CommInfo::temperature

## 8.130 ConnectionStatus Struct Reference

### Data Fields

- BYTE MDMConnStatus
- ULONGLONG MDMCallDuration

### 8.130.1 Detailed Description

This structure contains modem connection status

**Parameters**

| | |
|---|---|
| *MDMConn-Status* | • Current link status<br>– 0x01 - DISCONNECTED<br>– 0x02 - CONNECTED |
| *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). |

### 8.130.2 Field Documentation

#### 8.130.2.1 ULONGLONG ConnectionStatus::MDMCallDuration

#### 8.130.2.2 BYTE ConnectionStatus::MDMConnStatus

## 8.131 connectionStatus Struct Reference

### Data Fields

- uint8_t MDMConnStatus
- uint64_t MDMCallDuration

### 8.131.1 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. |

### 8.131.2 Field Documentation

#### 8.131.2.1 uint64_t connectionStatus::MDMCallDuration

#### 8.131.2.2 uint8_t connectionStatus::MDMConnStatus

## 8.132 connectNumInfo Struct Reference

### Data Fields

- BYTE numPresInd
- BYTE screeningInd
- BYTE numType
- BYTE numPlan
- BYTE callerIDLen
- BYTE callerID [81]

### 8.132.1 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_CALL_NO_LEN]* | • callerID of numLen length, NULL terminated. |

### 8.132.2 Field Documentation

#### 8.132.2.1 BYTE connectNumInfo::callerID[81]

#### 8.132.2.2 BYTE connectNumInfo::callerIDLen

#### 8.132.2.3 BYTE connectNumInfo::numPlan

#### 8.132.2.4 BYTE connectNumInfo::numPresInd

#### 8.132.2.5 BYTE connectNumInfo::numType

#### 8.132.2.6 BYTE connectNumInfo::screeningInd

## 8.133 CrashInfo Struct Reference

### Data Fields

- WORD numCrashes
- ULONG crashId
- ULONG crashData
- WORD crashStrLen
- CHAR * pCrashString
- WORD gcDumpStrLen
- CHAR * pGCDumpString

### 8.133.1 Detailed Description

This structure is used to store Crash Information

**Parameters**

| | |
|---|---|
| *numCrashes[OUT]* | • Number of instances of the remaining fields |
| *crashId[OUT]* | • Random crash id assigned at crash |
| *crashData[OUT]* | • Crash Data |
| *crashStrLen[IN/OUT]* | • Length of pCrashString as an INPUT, length of the pCrashString field returned by API as an OUTPUT |
| *pCrashString[OUT]* | • Pointer to store crash string |
| *gcDumpStrLen[IN/OUT]* | • Length of pGCDumpString as an INPUT, Length of the pGCDumpString field returned by API as an OUTPUT |
| *pGCDumpString[OUT]* | • gcdump string for the crash |

### 8.133.2 Field Documentation

#### 8.133.2.1 ULONG CrashInfo::crashData

#### 8.133.2.2 ULONG CrashInfo::crashId

#### 8.133.2.3 WORD CrashInfo::crashStrLen

#### 8.133.2.4 WORD CrashInfo::gcDumpStrLen

#### 8.133.2.5 WORD CrashInfo::numCrashes

#### 8.133.2.6 CHAR* CrashInfo::pCrashString

#### 8.133.2.7 CHAR* CrashInfo::pGCDumpString

## 8.134 CrashInfoParams Struct Reference

### Data Fields

- BYTE * pDevCrashStatus
- CrashInfo * pCrashInfo

### 8.134.1 Detailed Description

This structure is used to store Crash Information

**Parameters**

| | |
|---|---|
| *pDevCrash-Status[OUT]* | • Device Crash Status<br>• 0 - no crash<br>• 1 - crash has occurred |
| *pCrashInfo[OUT]* | • Pointer to structure CrashInfo (Optional parameter)<br>• See CrashInfo for more information |

### 8.134.2 Field Documentation

#### 8.134.2.1 CrashInfo* CrashInfoParams::pCrashInfo

#### 8.134.2.2 BYTE* CrashInfoParams::pDevCrashStatus

## 8.135 crashInfoParams Struct Reference

### Data Fields

- uint8_t crashStatus
- crashInformation crashInfo

### 8.135.1 Detailed Description

This structure contains crash information parameters

**Parameters**

| | |
|---|---|
| *crashStatus[OU-T]* | • Device Crash Status<br>• 0 - no crash<br>• 1 - crash has occurred<br>• 0xff - invalid data. |
| *crashInfo[OUT]* | • See crashInformation |

### 8.135.2 Field Documentation

#### 8.135.2.1 crashInformation crashInfoParams::crashInfo

#### 8.135.2.2 uint8_t crashInfoParams::crashStatus

## 8.136 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]

### 8.136.1 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[OUT]* | • length of the gcdumpString field as an input, length of the gcdumpString field returned by API as an OUTPUT |
| *gcdumpString[OUT]* | • gcdump string for the crash |

### 8.136.2 Field Documentation

#### 8.136.2.1 uint32_t crashInformation::crashData

#### 8.136.2.2 uint32_t crashInformation::crashId

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

#### 8.136.2.4 uint16_t crashInformation::crashStrlen

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

#### 8.136.2.6 uint16_t crashInformation::gcdumpStrlen

#### 8.136.2.7 uint16_t crashInformation::numCrashes

## 8.137 CreateProfileIn Struct Reference

### Data Fields

- BYTE * pProfileID
- BYTE * pProfileType
- QmiProfileInfo curProfile

### 8.137.1 Detailed Description

This structure contains the input parameters for SLQSCreateProfile

**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<br>– NULL is not allowed |
| *curProfile* | • union of Profile3GPP and Profile3GPP2 |

**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

### 8.137.2 Field Documentation

#### 8.137.2.1 QmiProfileInfo CreateProfileIn::curProfile

#### 8.137.2.2 BYTE* CreateProfileIn::pProfileID

#### 8.137.2.3 BYTE* CreateProfileIn::pProfileType

## 8.138 CreateProfileOut Struct Reference

### Data Fields

- BYTE * pProfileType
- BYTE * pProfileIndex
- USHORT * pExtErrorCode

### 8.138.1 Detailed Description

structure contains out parameter Information

**Parameters**

| | |
|---|---|
| *profileType* | • Identifies the type of profile 0x00 = 3GPP 0x01 = 3GPP2 |
| *profileIndex* | • Index identifying the profile that was created |
| *pExtErrorCode* | • The extended error code received from DS Profile subsystem |

### 8.138.2 Field Documentation

#### 8.138.2.1 USHORT* CreateProfileOut::pExtErrorCode

#### 8.138.2.2 BYTE* CreateProfileOut::pProfileIndex

#### 8.138.2.3 BYTE* CreateProfileOut::pProfileType

## 8.139 CsgId Struct Reference

### Data Fields

- ULONG csgId
- BYTE TlvPresent

### 8.139.1 Detailed Description

This structure contains csg ID.

**Parameters**

| | |
|---|---|
| *csgId* | • Closed subscriber group identifier. |
| *TlvPresent* | • Tlv Present or not |

### 8.139.2 Field Documentation

#### 8.139.2.1 ULONG CsgId::csgId

#### 8.139.2.2 BYTE CsgId::TlvPresent

## 8.140 CSGID Struct Reference

### Data Fields

- WORD mcc
- WORD mnc
- BYTE mncPcsDigits

- ULONG id
- BYTE rat

### 8.140.1 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 |

### 8.140.2 Field Documentation

#### 8.140.2.1 ULONG CSGID::id

#### 8.140.2.2 WORD CSGID::mcc

#### 8.140.2.3 WORD CSGID::mnc

#### 8.140.2.4 BYTE CSGID::mncPcsDigits

#### 8.140.2.5 BYTE CSGID::rat

## 8.141 CUGInfo Struct Reference

### Data Fields

- WORD CUGIndex
- BYTE SuppPrefCUG
- BYTE SuppOA

### 8.141.1 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 |

### 8.141.2 Field Documentation

#### 8.141.2.1 WORD CUGInfo::CUGIndex

#### 8.141.2.2 BYTE CUGInfo::SuppOA

#### 8.141.2.3 BYTE CUGInfo::SuppPrefCUG

## 8.142 curAMRConfig Struct Reference

### Data Fields

- BYTE gsmAmrStat
- BYTE wcdmaAmrStat

### 8.142.1 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 |

### 8.142.2 Field Documentation

#### 8.142.2.1 BYTE curAMRConfig::gsmAmrStat

#### 8.142.2.2 BYTE curAMRConfig::wcdmaAmrStat

## 8.143 CurrDataSysStat Struct Reference

### Data Fields

- BYTE * pPrefNetwork
- BYTE * pNetworkInfoLen
- CurrNetworkInfo * pCurrNetworkInfo

### 8.143.1 Detailed Description

Data System Status

**Parameters**

| | |
|---|---|
| *pPrefNetwork* | [OUT]<br>• Preferred Network<br>• Values:<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *pNetworkInfoLen* | [IN/OUT]<br>• 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 |
| *pCurrNetwork-Info* | [OUT]<br>• Network information<br>• See CurrNetworkInfo for more details |

### 8.143.2 Field Documentation

#### 8.143.2.1 CurrNetworkInfo* CurrDataSysStat::pCurrNetworkInfo

#### 8.143.2.2 BYTE* CurrDataSysStat::pNetworkInfoLen

#### 8.143.2.3 BYTE* CurrDataSysStat::pPrefNetwork

## 8.144 currentCatEvent Union Reference

### Data Fields

- struct CatEventIDDataTlv CatEvIDData
- struct CatAlPhaIdentifierTlv CatAlphaIdtfr
- struct CatEventListTlv CatEventLst
- struct CatRefreshTlv CatRefresh
- struct CatEndProactiveSessionTlv CatEndPS

### 8.144.1 Detailed Description

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

- Use CatEventIDDataTlv if the Event ID is any of the below.
  - 16
  - 17
  - 18
  - 19
  - 20
  - 23
  - 24
- Use CatAlPhaIdentifierTlv if the Event ID is 21
- Use CatEventListTlv if the Event ID is 22
- Use CatRefreshTlv if the Event ID is 25
- Use CatEndProactiveSessionTlv if the Event ID is 26

### 8.144.2 Field Documentation

#### 8.144.2.1 struct CatAlPhaIdentifierTlv currentCatEvent::CatAlphaIdtfr

#### 8.144.2.2 struct CatEndProactiveSessionTlv currentCatEvent::CatEndPS

#### 8.144.2.3 struct CatEventListTlv currentCatEvent::CatEventLst

#### 8.144.2.4 struct CatEventIDDataTlv currentCatEvent::CatEvIDData

#### 8.144.2.5 struct CatRefreshTlv currentCatEvent::CatRefresh

## 8.145 CurrentImgList Struct Reference

### Data Fields

- BYTE numEntries
- CurrImageInfo * pCurrImgInfo
- CHAR priver [16]
- CHAR pkgver [16]
- CHAR fwvers [16]
- CHAR carrier [16]

### 8.145.1 Detailed Description

This structure is used to store image list

**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 |
| *pCurrImgInfo[O-UT]* | • Currently Active Image List |
| *priver[OUT]* | • PRI version of the currently running firmware |
| *pkgver[OUT]* | • Package version of the currently running firmware |
| *fwvers[OUT]* | • firmware version of the currently running firmware |
| *carrier[OUT]* | • Carrier string of the currently running firmware |

### 8.145.2 Field Documentation

#### 8.145.2.1 CHAR CurrentImgList::carrier[16]

#### 8.145.2.2 CHAR CurrentImgList::fwvers[16]

#### 8.145.2.3 BYTE CurrentImgList::numEntries

#### 8.145.2.4 CurrImageInfo* CurrentImgList::pCurrImgInfo

#### 8.145.2.5 CHAR CurrentImgList::pkgver[16]

#### 8.145.2.6 CHAR CurrentImgList::priver[16]

## 8.146 currentPLMN Struct Reference

### Data Fields

- WORD MCC

- WORD MNC
- BYTE netDescrLength
- BYTE netDescr [255]

### 8.146.1 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 |

### 8.146.2 Field Documentation

#### 8.146.2.1 WORD currentPLMN::MCC

#### 8.146.2.2 WORD currentPLMN::MNC

#### 8.146.2.3 BYTE currentPLMN::netDescr[255]

#### 8.146.2.4 BYTE currentPLMN::netDescrLength

## 8.147 CurrImageInfo Struct Reference

### Data Fields

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

### 8.147.1 Detailed Description

This structure is used to store image information

**Parameters**

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

### 8.147.2 Field Documentation

#### 8.147.2.1 BYTE CurrImageInfo::buildID[255]

#### 8.147.2.2 BYTE CurrImageInfo::buildIDLen

#### 8.147.2.3 BYTE CurrImageInfo::imageType

#### 8.147.2.4 BYTE CurrImageInfo::uniqueID[16]

## 8.148 CurrNetworkInfo Struct Reference

### Data Fields

- BYTE NetworkType
- ULONG RATMask
- ULONG SOMask

### 8.148.1 Detailed Description

Network information structure

**Parameters**

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

| | |
|---|---|
| *RATMask* | • Radio Access Technology (RAT) mask to indicate the type of technology.<br>• Values:<br>– 0 - Don't Care<br>– 0x8000 - NULL Bearer<br>• CDMA RAT mask values:<br>– 0x01 - CDMA_1x<br>– 0x02 - EVDO_REV0<br>– 0x04 - EVDO_REVA<br>– 0x08 - EVDO_REVB<br>– 0x10 - EHRPD<br>– 0x20 - FMC<br>• UMTS RAT mask values:<br>– 0x01 - WCDMA<br>– 0x02 - GPRS<br>– 0x04 - HSDPA<br>– 0x08 - HSUPA<br>– 0x10 - EDGE<br>– 0x20 - LTE<br>– 0x40 - HSDPA+<br>– 0x80 - DC_HSDPA+<br>– 0x100 - 64_QAM<br>– 0x200 - TDSCDMA |
| *SOMask* | • Service Option (SO) mask to indicate the service option or type of application.<br>• Values:<br>– 0 - Don't Care<br>• CDMA 1x SO mask values:<br>– 0x01 - CDMA_1X_IS95<br>– 0x02 - CDMA_1X_IS2000<br>– 0x04 - CDMA_1X_IS2000_REL_A<br>• CDMA EV-DO Rev 0 SO mask values:<br>– 0x01 - DPA<br>• CDMA EV-DO Rev A SO mask values:<br>– 0x01 - DPA<br>– 0x02 - MFPA<br>– 0x04 - EMPA<br>– 0x08 - EMPA_EHRPD<br>• CDMA EV-DO Rev B SO mask values:<br>– 0x01 - DPA<br>– 0x02 - MFPA<br>– 0x04 - EMPA<br>– 0x08 - EMPA_EHRPD<br>– 0x10 - MMPA<br>– 0x20 - MMPA_EHRPD |

### 8.148.2 Field Documentation

#### 8.148.2.1 BYTE CurrNetworkInfo::NetworkType

#### 8.148.2.2 ULONG CurrNetworkInfo::RATMask

#### 8.148.2.3 ULONG CurrNetworkInfo::SOMask

## 8.149 currNetworkInfo Struct Reference

### Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

### 8.149.1 Detailed Description

Network information structure

**Parameters**

| | |
|---|---|
| *NetworkType* | • Values:<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *RATMask* | • Radio Access Technology (RAT) mask to indicate the type of technology.<br>• Values:<br>– 0 - Don't Care<br>– 0x8000 - NULL Bearer<br>• CDMA RAT mask values:<br>– 0x01 - CDMA_1x<br>– 0x02 - EVDO_REV0<br>– 0x04 - EVDO_REVA<br>– 0x08 - EVDO_REVB<br>– 0x10 - EHRPD<br>– 0x20 - FMC<br>• UMTS RAT mask values:<br>– 0x01 - WCDMA<br>– 0x02 - GPRS<br>– 0x04 - HSDPA<br>– 0x08 - HSUPA<br>– 0x10 - EDGE<br>– 0x20 - LTE<br>– 0x40 - HSDPA+<br>– 0x80 - DC_HSDPA+<br>– 0x100 - 64_QAM<br>– 0x200 - TDSCDMA |

| | |
|---|---|
| *SOMask* | • Service Option (SO) mask to indicate the service option or type of application.<br>• Values:<br>– 0 - Don't Care<br>• CDMA 1x SO mask values:<br>– 0x01 - CDMA_1X_IS95<br>– 0x02 - CDMA_1X_IS2000<br>– 0x04 - CDMA_1X_IS2000_REL_A<br>• CDMA EV-DO Rev 0 SO mask values:<br>– 0x01 - DPA<br>• CDMA EV-DO Rev A SO mask values:<br>– 0x01 - DPA<br>– 0x02 - MFPA<br>– 0x04 - EMPA<br>– 0x08 - EMPA_EHRPD<br>• CDMA EV-DO Rev B SO mask values:<br>– 0x01 - DPA<br>– 0x02 - MFPA<br>– 0x04 - EMPA<br>– 0x08 - EMPA_EHRPD<br>– 0x10 - MMPA<br>– 0x20 - MMPA_EHRPD |

### 8.149.2 Field Documentation

#### 8.149.2.1 uint8_t currNetworkInfo::NetworkType

#### 8.149.2.2 uint32_t currNetworkInfo::RATMask

#### 8.149.2.3 uint32_t currNetworkInfo::SOMask

## 8.150 custFeaturesInfo Struct Reference

### Data Fields

- ULONG GpsEnable
- BYTE * pDisableIMSI
- WORD * pIPFamSupport
- BYTE * pRMAutoConnect
- BYTE * pGPSSel
- BYTE * pSMSSupport
- BYTE * pIsVoiceEnabled
- BYTE * pDHCPRelayEnabled
- BYTE * pGPSLPM

### 8.150.1 Detailed Description

This structure contains current settings of custom features

**Parameters**

| | |
|---|---|
| *GpsEnable[OUT]* | • describes if GPS is enabled or disabled<br>• values:<br>– 0x00 - GPS is disabled<br>– 0x01 - GPS is enabled<br>• function SLQSGetCustFeatures() returns a default value FFFFFFFF if no value is returned by the modem |
| *pDisableIMSI[OUT]* | • 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>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem |
| *pIPFamSupport[OUT]* | • optional 2 byte BitMask<br>• bitmask representing the IP families supported<br>• values:<br>– 0x01 - IPv4<br>– 0x02 - IPv6<br>– 0x04 - IPv4v6<br>• function SLQSGetCustFeatures() returns a default value FFFF if no value is returned by the modem |
| *pRMAutoConnect[OUT]* | • optional 1 byte parameter<br>• QMI Mode RM Net Auto Connect Support<br>• values:<br>– 0x00 - Not Supported<br>– 0x01 - Supported<br>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem |
| *pGPSSel[OUT]* | • 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>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem |

| | |
|---|---|
| *pSMSSupport[-OUT]* | • 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<br>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem. In this case assume, SMS is supported. |
| *pIsVoice-Enabled[OUT]* | • 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 |
| *pDHCPRelay-Enabled[OUT]* | • optional 1 byte parameter<br>• DHCP Relay support<br>• values:<br>– 0x00 - Disable DHCP relay<br>– 0x01 - Enable DHCP relay |
| *pGPSLPM[OUT]* | • 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 |

### 8.150.2 Field Documentation

#### 8.150.2.1 ULONG custFeaturesInfo::GpsEnable

#### 8.150.2.2 BYTE* custFeaturesInfo::pDHCPRelayEnabled

#### 8.150.2.3 BYTE* custFeaturesInfo::pDisableIMSI

#### 8.150.2.4 BYTE* custFeaturesInfo::pGPSLPM

#### 8.150.2.5 BYTE* custFeaturesInfo::pGPSSel

#### 8.150.2.6 WORD* custFeaturesInfo::pIPFamSupport

#### 8.150.2.7 BYTE* custFeaturesInfo::pIsVoiceEnabled

#### 8.150.2.8 BYTE* custFeaturesInfo::pRMAutoConnect

#### 8.150.2.9 BYTE* custFeaturesInfo::pSMSSupport

## 8.151 custFeaturesSetting Struct Reference

### Data Fields

- BYTE * pGPSSel
- ULONG * pGPSEnable
- BYTE * pIsVoiceEnabled
- BYTE * pDHCPRelayEnabled
- BYTE * pGPSLPM

### 8.151.1 Detailed Description

This structure contains settings to be used for custom features

**Parameters**

| | |
|---|---|
| *pGPSSel* | • 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 |
| *pGPSEnable* | • optional 4 byte parameter<br>• GPS Enable/Disable<br>• values: The value of 7 least significant bits:<br>– 0 - Disabled<br>– 1 - MT & MO enabled<br>– 2 - MO enabled<br>– 3 - MT enabled<br>– 4 - MT & MO enabled if GPS_DISABLE pin is not asserted<br>– 5 - MO GPS enabled if GPS_DISABLE pin is not asserted<br>– 6 - MT GPS enabled if GPS_DISABLE pin is not asserted |

**Note**

Only MC7750 3.5.x firmware supports above 0x04, 0x05 and 0x06 settings. To disable GLONASS, set the most significant bit - 0x80. This setting is only applicable if GPS is not Disabled.

**Parameters**

| | |
|---|---|
| *pIsVoiceEnabled* | • optional 1 byte parameter<br>• voice enabled/disabled<br>• values:<br>– 0 - Enable voice on both AT and QMI interface (default)<br>– 1 - Reserved<br>– 2 - Disable voice on both AT and QMI interface |

| | |
|---|---|
| *pDHCPRelay-Enabled* | • optional 1 byte parameter<br>• DHCPRELAYENABLE support<br>• values:<br>– 0 - Disable DHCP relay<br>– 1 - Enable DHCP relay |
| *pGPSLPM* | • optional 1 byte parameter<br>• GPSLPM support<br>• values:<br>– 0 - Enable GPS in Low Power Mode<br>– 1 - Disable GPS in Low Power Mode |

### 8.151.2 Field Documentation

#### 8.151.2.1 BYTE* custFeaturesSetting::pDHCPRelayEnabled

#### 8.151.2.2 ULONG* custFeaturesSetting::pGPSEnable

#### 8.151.2.3 BYTE* custFeaturesSetting::pGPSLPM

#### 8.151.2.4 BYTE* custFeaturesSetting::pGPSSel

#### 8.151.2.5 BYTE* custFeaturesSetting::pIsVoiceEnabled

## 8.152 custSettingInfo Struct Reference

### Data Fields

- WORD id_length
- CHAR cust_id [64+1]
- WORD value_length
- BYTE cust_value [8+1]
- WORD cust_attr

### 8.152.1 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) |
| *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 |

### 8.152.2 Field Documentation

#### 8.152.2.1 WORD custSettingInfo::cust_attr

#### 8.152.2.2 CHAR custSettingInfo::cust_id[64+1]

#### 8.152.2.3 BYTE custSettingInfo::cust_value[8+1]

#### 8.152.2.4 WORD custSettingInfo::id_length

#### 8.152.2.5 WORD custSettingInfo::value_length

## 8.153 custSettingList Struct Reference

### Data Fields

- BYTE list_type
- WORD num_instances
- custSettingInfo custSetting [256]

### 8.153.1 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 custSettingInfo for more information |

### 8.153.2 Field Documentation

#### 8.153.2.1 custSettingInfo custSettingList::custSetting[256]

#### 8.153.2.2 BYTE custSettingList::list_type

#### 8.153.2.3 WORD custSettingList::num_instances

## 8.154 dataBearers Struct Reference

### Data Fields

- BYTE dataBearerMask
- QmiWDSDataBearerTechnology * pCurDataBearerTechnology
- QmiWDSDataBearerTechnology * pLastCallDataBearerTechnology

### 8.154.1 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 qmiDataBearerMasks for bit-mask positions. |
| *pCurDataBearerTechnology[OUT]* | • current data bearer technology value.<br>– NULL if the parameter is not required |
| *pLastCallDataBearerTechnology[OUT]* | • last call data bearer technology value.<br>– NULL if the parameter is not required |

### 8.154.2 Field Documentation

#### 8.154.2.1 BYTE dataBearers::dataBearerMask

#### 8.154.2.2 QmiWDSDataBearerTechnology* dataBearers::pCurDataBearerTechnology

#### 8.154.2.3 QmiWDSDataBearerTechnology* dataBearers::pLastCallDataBearerTechnology

## 8.155 DataBearerTech Struct Reference

### Data Fields

- ULONG techType
- ULONG ratValue
- ULONGLONG soMask

### 8.155.1 Detailed Description

Network information structure

**Parameters**

| | |
|---|---|
| *TechType* | • Technology type<br>• Values:<br>– 0 - WDS_BEARER_TECH_NETWORK_3GPP - 3GPP<br>– 1 - WDS_BEARER_TECH_NETWORK_3GPP2 - 3GPP2 |
| *ratValue* | • Radio Access Technology (RAT) value<br>• Values:<br>– 0x00 - WDS_BEARER_TECH_RAT_EX_NULL_BEARER - NULL bearer<br>– 0x01 - WDS_BEARER_TECH_RAT_EX_3GPP_WCDMA - 3GPP WCDMA<br>– 0x02 - WDS_BEARER_TECH_RAT_EX_3GPP_GERAN - 3GPP GERAN<br>– 0x03 - WDS_BEARER_TECH_RAT_EX_3GPP_LTE - 3GPP LTE<br>– 0x04 - WDS_BEARER_TECH_RAT_EX_3GPP_TDSCDMA - 3GPP TDSCDMA<br>– 0x05 - WDS_BEARER_TECH_RAT_EX_3GPP_WLAN - 3GPP WLAN<br>– 0x64 - WDS_BEARER_TECH_RAT_EX_3GPP_MAX - 3GPP maximum<br>– 0x65 - WDS_BEARER_TECH_RAT_EX_3GPP2_1X - 3GPP2 1X<br>– 0x66 - WDS_BEARER_TECH_RAT_EX_3GPP2_HRPD - 3GPP2 HRPD<br>– 0x67 - WDS_BEARER_TECH_RAT_EX_3GPP2_EHRPD - 3GPP2 EHRPD<br>– 0x68 - WDS_BEARER_TECH_RAT_EX_3GPP2_WLAN - 3GPP2 WLAN<br>– 0xC8 - WDS_BEARER_TECH_RAT_EX_3GPP2_MAX - 3GPP2 maximum |

| | |
|---|---|
| *SOMask* | • 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 |

### 8.155.2 Field Documentation

#### 8.155.2.1 ULONG DataBearerTech::ratValue

#### 8.155.2.2 ULONGLONG DataBearerTech::soMask

#### 8.155.2.3 ULONG DataBearerTech::techType

## 8.156 DataBearerTechExt Struct Reference

### Data Fields

- DataBearerTech * pBearerTech
- DataBearerTech * pLastBearerTech

### 8.156.1 Detailed Description

Data Bearer Technology Ext

**Parameters**

| | |
|---|---|
| *pBearerTech* | [OUT]<br>• See DataBearerTech for more information |
| *pLastBearerTech* | [OUT]<br>• See DataBearerTech for more information |

### 8.156.2 Field Documentation

#### 8.156.2.1 DataBearerTech* DataBearerTechExt::pBearerTech

#### 8.156.2.2 DataBearerTech* DataBearerTechExt::pLastBearerTech

## 8.157 dataBearerTechnology Struct Reference

### Data Fields

- BYTE currentNetwork
- ULONG ratMask
- ULONG soMask

### 8.157.1 Detailed Description

Structure to hold the current data bearer technology values

**Parameters**

| | |
|---|---|
| *pCurrent-Network[OUT]* | • current selected network<br>– 0 - UNKNOWN<br>– 1 - 3GPP2<br>– 2 - 3GPP |

| | |
|---|---|
| *pRatMask[OUT]* | • Radio Access Technology (RAT) mask to indicate the type of technology (RAT mask value of zero indicates that this field is ignored)<br>– 0x8000 - NULL Bearer<br>– 0x0000 - DO_NOT_CARE CDMA RAT mask<br>– 0x01 - CDMA_1X<br>– 0x02 - EVDO_REV0<br>– 0x04 - EVDO_REVA UMTS RAT mask<br>– 0x01 - WCDMA<br>– 0x02 - GPRS<br>– 0x04 - HSDPA<br>– 0x08 - HSUPA<br>– 0x10 - EDGE<br>– 0x20 - LTE<br>– 0x40 - HSDPA+<br>– 0x80 - DC_HSDPA+ |
| *pSoMask[OUT]* | • Service Option (SO) mask to indicate the SO or type of application (SO mask value of zero indicates that this field is ignored)<br>– 0x00 - DO_NOT_CARE CDMA 1X SO mask<br>– 0x01 - CDMA_1X_IS95<br>– 0x02 - CDMA_1X_IS2000<br>– 0x04 - CDMA_1X_IS2000_REL_A CDMA EV-DO Rev A SO mask<br>– 0x01 - EVDO_REVA_DPA<br>– 0x02 - EVDO_REVA_MFPA<br>– 0x04 - EVDO_REVA_EMPA<br>– 0x08 - EVDO_REVA_EMPA_EHRPD |

### 8.157.2 Field Documentation

#### 8.157.2.1 BYTE dataBearerTechnology::currentNetwork

#### 8.157.2.2 ULONG dataBearerTechnology::ratMask

#### 8.157.2.3 ULONG dataBearerTechnology::soMask

## 8.158 dataRate Struct Reference

### Data Fields

- ULONG dataRateMax
- ULONG guaranteedRate

### 8.158.1 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) |

### 8.158.2 Field Documentation

#### 8.158.2.1 ULONG dataRate::dataRateMax

#### 8.158.2.2 ULONG dataRate::guaranteedRate

## 8.159 dataSrvCapabilities Struct Reference

### Data Fields

- BYTE dataCapabilitiesLen
- BYTE dataCapabilities [0x20]

### 8.159.1 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+ |

### 8.159.2 Field Documentation

#### 8.159.2.1 BYTE dataSrvCapabilities::dataCapabilities[0x20]

#### 8.159.2.2 BYTE dataSrvCapabilities::dataCapabilitiesLen

## 8.160 DataStatusDetail Struct Reference

### Data Fields

- ULONG IPAddress
- BYTE LastErrCode

### 8.160.1 Detailed Description

This structure contains Data Status Details

**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 |

### 8.160.2 Field Documentation

#### 8.160.2.1 ULONG DataStatusDetail::IPAddress

#### 8.160.2.2 BYTE DataStatusDetail::LastErrCode

## 8.161 DataULongLongTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONGLONG ullData

### 8.161.1 Field Documentation

#### 8.161.1.1 BYTE DataULongLongTlv::TlvPresent

#### 8.161.1.2 ULONGLONG DataULongLongTlv::ullData

## 8.162 DataULongTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG ulData

### 8.162.1 Field Documentation

#### 8.162.1.1 BYTE DataULongTlv::TlvPresent

#### 8.162.1.2 ULONG DataULongTlv::ulData

## 8.163 DcsUsbPortNames Struct Reference

### Data Fields

- CHAR AtCmdPort [32]
- CHAR NmeaPort [32]
- CHAR DmPort [32]

### 8.163.1 Field Documentation

#### 8.163.1.1 CHAR DcsUsbPortNames::AtCmdPort[32]

#### 8.163.1.2 CHAR DcsUsbPortNames::DmPort[32]

#### 8.163.1.3 CHAR DcsUsbPortNames::NmeaPort[32]

## 8.164 delAssistDataStatus Struct Reference

### Data Fields

- ULONG status

### 8.164.1 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 |

**Note**

None

### 8.164.2 Field Documentation

#### 8.164.2.1 ULONG delAssistDataStatus::status

## 8.165 depersonalizationInformation Struct Reference

### Data Fields

- BYTE feature
- BYTE operation
- BYTE ckLen
- BYTE ckVal [255]

### 8.165.1 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. |

### 8.165.2 Field Documentation

#### 8.165.2.1 BYTE depersonalizationInformation::ckLen

#### 8.165.2.2 BYTE depersonalizationInformation::ckVal[255]

#### 8.165.2.3 BYTE depersonalizationInformation::feature

#### 8.165.2.4 BYTE depersonalizationInformation::operation

## 8.166 detailSvcInfo Struct Reference

### Data Fields

- BYTE srvStatus
- BYTE srvCapability
- BYTE hdrSrvStatus
- BYTE hdrHybrid
- BYTE isSysForbidden

### 8.166.1 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 |

### 8.166.2 Field Documentation

#### 8.166.2.1 BYTE detailSvcInfo::hdrHybrid

#### 8.166.2.2 BYTE detailSvcInfo::hdrSrvStatus

#### 8.166.2.3 BYTE detailSvcInfo::isSysForbidden

#### 8.166.2.4 BYTE detailSvcInfo::srvCapability

#### 8.166.2.5 BYTE detailSvcInfo::srvStatus

## 8.167 DeviceConfigDetail Struct Reference

### Data Fields

- BYTE Technology
- BYTE QLIC
- BYTE Chipset
- BYTE HWVersion

### 8.167.1 Detailed Description

This structure contains Device Configuration Details

**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 |

### 8.167.2 Field Documentation

#### 8.167.2.1 BYTE DeviceConfigDetail::Chipset

#### 8.167.2.2 BYTE DeviceConfigDetail::HWVersion

#### 8.167.2.3 BYTE DeviceConfigDetail::QLIC

#### 8.167.2.4 BYTE DeviceConfigDetail::Technology

## 8.168 DHCPOption Struct Reference

### Data Fields

- BYTE optCode
- BYTE optValLen
- BYTE * pOptVal

### 8.168.1 Detailed Description

This structure contains DHCPv4 lease option values

**Parameters**

| | |
|---|---|
| *optCode* | • Values<br>– Option code |
| *optValLen* | • Values<br>– Option value length |
| *pOptValue* | • Val<br>– Option value |

### 8.168.2 Field Documentation

#### 8.168.2.1 BYTE DHCPOption::optCode

#### 8.168.2.2 BYTE DHCPOption::optValLen

#### 8.168.2.3 BYTE* DHCPOption::pOptVal

## 8.169 DHCPOptionList Struct Reference

### Data Fields

- BYTE numOpt
- DHCPOption * pOptions

### 8.169.1 Detailed Description

This structure contains DHCPv4 lease option list

**Parameters**

| | |
|---|---|
| *optListSize* | • Values<br>– Size of Option List |
| *pOptions* | • Values<br>– Options |

### 8.169.2 Field Documentation

#### 8.169.2.1 BYTE DHCPOptionList::numOpt

#### 8.169.2.2 DHCPOption* DHCPOptionList::pOptions

## 8.170 diagInfo Struct Reference

### Data Fields

- BYTE diagInfoLen
- BYTE diagnosticInfo [255]

### 8.170.1 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_DESCRIPTION_LENGTH]* | • Diagnostic information. |

### 8.170.2 Field Documentation

#### 8.170.2.1 BYTE diagInfo::diagInfoLen

#### 8.170.2.2 BYTE diagInfo::diagnosticInfo[255]

## 8.171 dirNum Struct Reference

### Data Fields

- BYTE dirNumLen
- BYTE dirNum [255]

### 8.171.1 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. |

### 8.171.2 Field Documentation

#### 8.171.2.1 BYTE dirNum::dirNum[255]

#### 8.171.2.2 BYTE dirNum::dirNumLen

## 8.172 dms_ActivationStatusTlv Struct Reference

### Data Fields

- uint16_t TlvPresent
- uint32_t activationStatus

### 8.172.1 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 |

### 8.172.2 Field Documentation

#### 8.172.2.1 uint32_t dms_ActivationStatusTlv::activationStatus

#### 8.172.2.2 uint16_t dms_ActivationStatusTlv::TlvPresent

## 8.173 dms_OperatingModeTlv Struct Reference

### Data Fields

- uint16_t TlvPresent
- uint32_t operatingMode

### 8.173.1 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 |

### 8.173.2 Field Documentation

#### 8.173.2.1 uint32_t dms_OperatingModeTlv::operatingMode

#### 8.173.2.2 uint16_t dms_OperatingModeTlv::TlvPresent

## 8.174 dms_PSMActiveTimerIndTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t ActiveTimerInd

### 8.174.1 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). |

### 8.174.2 Field Documentation

#### 8.174.2.1 uint32_t dms_PSMActiveTimerIndTlv::ActiveTimerInd

#### 8.174.2.2 uint8_t dms_PSMActiveTimerIndTlv::TlvPresent

## 8.175 dms_PSMActiveTimerTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t activeTimer

### 8.175.1 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). |

### 8.175.2 Field Documentation

#### 8.175.2.1 uint32_t dms_PSMActiveTimerTlv::activeTimer

#### 8.175.2.2 uint8_t dms_PSMActiveTimerTlv::TlvPresent

## 8.176 dms_PSMDurationDueToOOSTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t durationDueToOOS

### 8.176.1 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). |

### 8.176.2 Field Documentation

#### 8.176.2.1 uint32_t dms_PSMDurationDueToOOSTlv::durationDueToOOS

#### 8.176.2.2 uint8_t dms_PSMDurationDueToOOSTlv::TlvPresent

## 8.177 dms_PSMDurationThresholdTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t durationThreshold

### 8.177.1 Detailed Description

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

**Parameters**

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

### 8.177.2 Field Documentation

#### 8.177.2.1 uint32_t dms_PSMDurationThresholdTlv::durationThreshold

#### 8.177.2.2 uint8_t dms_PSMDurationThresholdTlv::TlvPresent

## 8.178 dms_PSMEarlyWakeupTimeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t earlyWakeupTime

### 8.178.1 Detailed Description

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

**Parameters**

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

### 8.178.2 Field Documentation

#### 8.178.2.1 uint32_t dms_PSMEarlyWakeupTimeTlv::earlyWakeupTime

#### 8.178.2.2 uint8_t dms_PSMEarlyWakeupTimeTlv::TlvPresent

## 8.179 dms_PSMEnableStateIndTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t EnableStateInd

### 8.179.1 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 |

### 8.179.2 Field Documentation

#### 8.179.2.1 uint8_t dms_PSMEnableStateIndTlv::EnableStateInd

#### 8.179.2.2 uint8_t dms_PSMEnableStateIndTlv::TlvPresent

## 8.180 dms_PSMEnableStateTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t enableState

### 8.180.1 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 |

### 8.180.2 Field Documentation

#### 8.180.2.1 uint8_t dms_PSMEnableStateTlv::enableState

#### 8.180.2.2 uint8_t dms_PSMEnableStateTlv::TlvPresent

## 8.181 dms_PSMPeriodicUpdateTimerIndTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t PeriodicUpdateTimerInd

### 8.181.1 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). |

### 8.181.2 Field Documentation

#### 8.181.2.1 uint32_t dms_PSMPeriodicUpdateTimerIndTlv::PeriodicUpdateTimerInd

#### 8.181.2.2 uint8_t dms_PSMPeriodicUpdateTimerIndTlv::TlvPresent

## 8.182 dms_PSMPeriodicUpdateTimerTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t periodicUpdateTimer

### 8.182.1 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). |

### 8.182.2 Field Documentation

#### 8.182.2.1 uint32_t dms_PSMPeriodicUpdateTimerTlv::periodicUpdateTimer

#### 8.182.2.2 uint8_t dms_PSMPeriodicUpdateTimerTlv::TlvPresent

## 8.183 dms_PSMRandomizationWindowTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t randomizationWindow

### 8.183.1 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) |

### 8.183.2 Field Documentation

#### 8.183.2.1 uint32_t dms_PSMRandomizationWindowTlv::randomizationWindow

#### 8.183.2.2 uint8_t dms_PSMRandomizationWindowTlv::TlvPresent

## 8.184 dms_TemperatureTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t TempStat
- uint16_t Temperature

### 8.184.1 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 |

### 8.184.2 Field Documentation

#### 8.184.2.1 uint16_t dms_TemperatureTlv::Temperature

#### 8.184.2.2 uint8_t dms_TemperatureTlv::TempStat

#### 8.184.2.3 uint8_t dms_TemperatureTlv::TlvPresent

## 8.185 dms_VoltageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t VoltStat
- uint16_t Voltage

### 8.185.1 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 |

### 8.185.2 Field Documentation

#### 8.185.2.1 uint8_t dms_VoltageTlv::TlvPresent

#### 8.185.2.2 uint16_t dms_VoltageTlv::Voltage

#### 8.185.2.3 uint8_t dms_VoltageTlv::VoltStat

## 8.186 dmsCurrentPRLInfo Struct Reference

### Data Fields

- WORD * pPRLVersion
- BYTE * pPRLPreference

### 8.186.1 Detailed Description

This structure contains GetCurrentPRLInfo response parameter

**Parameters**

| | |
|---|---|
| *pPRLVersion[OUT]* | - Optional<br>• PRL version of device. |
| *pPRLPreference* | [OUT]- Optional<br>• PRL Preference<br>– 0 - Unset<br>– 1 - Set |

### 8.186.2 Field Documentation

#### 8.186.2.1 BYTE* dmsCurrentPRLInfo::pPRLPreference

#### 8.186.2.2 WORD* dmsCurrentPRLInfo::pPRLVersion

## 8.187 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

### 8.187.1 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 |

### 8.187.2 Field Documentation

#### 8.187.2.1 uint16_t DMScustSettingInfo::cust_attr

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

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

#### 8.187.2.4 uint16_t DMScustSettingInfo::id_length

#### 8.187.2.5 uint16_t DMScustSettingInfo::value_length

## 8.188 DMScustSettingList Struct Reference

### Data Fields

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

### 8.188.1 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 |

### 8.188.2 Field Documentation

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

#### 8.188.2.2 uint8_t DMScustSettingList::list_type

#### 8.188.2.3 uint16_t DMScustSettingList::num_instances

## 8.189 DMSgetCustomFeatureV2 Struct Reference

### Data Fields

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

### 8.189.1 Detailed Description

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

**Parameters**

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

### 8.189.2 Field Documentation

#### 8.189.2.1 DMScustSettingInfo* DMSgetCustomFeatureV2::pCustSettingInfo

#### 8.189.2.2 DMScustSettingList* DMSgetCustomFeatureV2::pCustSettingList

#### 8.189.2.3 DMSgetCustomInput* DMSgetCustomFeatureV2::pGetCustomInput

## 8.190 DMSgetCustomInput Struct Reference

### Data Fields

- uint8_t cust_id [64+1]
- uint8_t list_type

### 8.190.1 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 |

### 8.190.2 Field Documentation

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

#### 8.190.2.2 uint8_t DMSgetCustomInput::list_type

## 8.191 dmsIndicationRegisterReq Struct Reference

### Data Fields

- BYTE * pSwiGetResetInd

### 8.191.1 Detailed Description

This structure contains the SLQSDmsSwiIndicationRegister request parameters.

**Parameters**

| | |
|---|---|
| *pGetResetInd* | [IN,Optional]<br>• Get Reset Info indication registration. The following callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |

**Note**

'NULL' value confirms that the indication value is not sent.

### 8.191.2 Field Documentation

#### 8.191.2.1 BYTE* dmsIndicationRegisterReq::pSwiGetResetInd

## 8.192 dmsPsmCfgParamChange Struct Reference

### Data Fields

- BYTE * pEnableState
- ULONG * pActiveTimer
- ULONG * pPeriodicUpdateTimer

### 8.192.1 Detailed Description

Contains the parameters passed for SLQSSetIMSARegStatusCallback by the device.

**Parameters**

| | |
|---|---|
| *pEnableState* | • PSM Enable State<br>• Values<br>– 0 - PSM is not enabled<br>– 1 - PSM is enabled |
| *pActiveTimer* | • Active Timer<br>• PSM active timer value (in seconds). |
| *pPeriodic-UpdateTimer* | • Periodic Update Timer<br>• PSM periodic update timer value (in seconds). |

### 8.192.2 Field Documentation

#### 8.192.2.1 ULONG* dmsPsmCfgParamChange::pActiveTimer

#### 8.192.2.2 BYTE* dmsPsmCfgParamChange::pEnableState

#### 8.192.2.3 ULONG* dmsPsmCfgParamChange::pPeriodicUpdateTimer

## 8.193 dmsSwiGetPCInfo Struct Reference

### Data Fields

- BYTE opMode
- ULONG * pLpmFlag
- BYTE * pWdisable
- BYTE * pPowerOffMode
- BYTE * pPersistentLpm

### 8.193.1 Detailed Description

This structure contains the TLV required to Get device power control status information.

**Parameters**

| | |
|---|---|
| *[OUT]* | opMode<br>• 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 |
| *[OUT]* | pLpmFlag[Optional]<br>• 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. |
| *[OUT]* | pWdisable[Optional]<br>• W_DISABLE<br>• Values<br>– 0 - Switch set to ON position<br>– 1 - Switch set to OFF position |
| *[OUT]* | pPowerOffMode[Optional]<br>• Power-off Mode<br>• Action taken when W_DISABLE is switched to the OFF position<br>– 0 - LPM<br>– 1 - Shutdown<br>– 2 - Ignore |
| *[OUT]* | pPersistentLpm[Optional]<br>• LPM Persistence<br>• Values<br>– 0 - Non-persistent LPM<br>– 1 - Persistent LPM |

### 8.193.2 Field Documentation

#### 8.193.2.1 BYTE dmsSwiGetPCInfo::opMode

#### 8.193.2.2 ULONG* dmsSwiGetPCInfo::pLpmFlag

#### 8.193.2.3 BYTE* dmsSwiGetPCInfo::pPersistentLpm

#### 8.193.2.4 BYTE* dmsSwiGetPCInfo::pPowerOffMode

#### 8.193.2.5 BYTE* dmsSwiGetPCInfo::pWdisable

## 8.194 dmsSwiGetResetInfo Struct Reference

### Data Fields

- BYTE type
- BYTE source

### 8.194.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

**Parameters**

| | |
|---|---|
| *OUT]* | type[OUT]<br>• 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 |
| *OUT]* | source[OUT]<br>• entity which initiated the reset or power down, possible values listed below:<br>– 0 - unknown<br>– 1 - user requested<br>– 2 - hardware switch<br>– 3 - temperature critical<br>– 4 - voltage critical<br>– 5 - configuration update<br>– 6 - LWM2M<br>– 7 - OMA-DM<br>– 8 - FOTA |

### 8.194.2 Field Documentation

#### 8.194.2.1 BYTE dmsSwiGetResetInfo::source

#### 8.194.2.2 BYTE dmsSwiGetResetInfo::type

## 8.195 Domain Struct Reference

### Data Fields

- WORD domainLen
- CHAR domainName [256]

### 8.195.1 Detailed Description

This structure contains the DomainName Information

**Parameters**

| | |
|---|---|
| *domainLen* | • length of the received Domain name |
| *domainName* | • Domain name(Max 256 characters) |

### 8.195.2 Field Documentation

#### 8.195.2.1 WORD Domain::domainLen

#### 8.195.2.2 CHAR Domain::domainName[256]

## 8.196 DomainNameList Struct Reference

### Data Fields

- BYTE numInstances
- struct Domain domain [10]

### 8.196.1 Detailed Description

This structure contains the DomainNameList Information

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of Domain name received |
| *domain* | • Domain name information(Max 10 Domain names) |

### 8.196.2 Field Documentation

#### 8.196.2.1 struct Domain DomainNameList::domain[10]

#### 8.196.2.2 BYTE DomainNameList::numInstances

## 8.197 DRCParams Struct Reference

### Data Fields

- BYTE DRCValue
- BYTE DRCCover

### 8.197.1 Detailed Description

This structure contains Data Rate Channel parameters

**Parameters**

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

### 8.197.2 Field Documentation

#### 8.197.2.1 BYTE DRCParams::DRCCover

#### 8.197.2.2 BYTE DRCParams::DRCValue

## 8.198 DTMFInfo Struct Reference

### Data Fields

- BYTE callID
- BYTE DTMFEvent
- BYTE digitCnt
- BYTE digitBuff [255]

### 8.198.1 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_DESCRIPTION_LENGTH]* | • DTMF digit buffer in ASCII string which is NULL terminated |

### 8.198.2 Field Documentation

#### 8.198.2.1 BYTE DTMFInfo::callID

#### 8.198.2.2 BYTE DTMFInfo::digitBuff[255]

#### 8.198.2.3 BYTE DTMFInfo::digitCnt

#### 8.198.2.4 BYTE DTMFInfo::DTMFEvent

## 8.199 DTMFLengths Struct Reference

### Data Fields

- BYTE DTMFPulseWidth
- BYTE DTMFInterdigitInterval

### 8.199.1 Detailed Description

This structure contains Voice Burst DTMF pulse length information

**Parameters**

| | |
|---|---|
| *DTMFPulseWidth* | • 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 |
| *DTMFInterdigitInterval* | • 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 |

### 8.199.2 Field Documentation

#### 8.199.2.1 BYTE DTMFLengths::DTMFInterdigitInterval

#### 8.199.2.2 BYTE DTMFLengths::DTMFPulseWidth

## 8.200 DUNCallInfoInd Struct Reference

### Data Fields

- BYTE MdmConnStatus
- WORD CallEndReason
- ULONGLONG TXOKBytesCount
- ULONGLONG RXOKBytesCount
- BYTE DormancyStatus
- BYTE DataBearerTech
- channelRate ChannelRate

### 8.200.1 Field Documentation

#### 8.200.1.1 WORD DUNCallInfoInd::CallEndReason

#### 8.200.1.2 channelRate DUNCallInfoInd::ChannelRate

#### 8.200.1.3 BYTE DUNCallInfoInd::DataBearerTech

#### 8.200.1.4 BYTE DUNCallInfoInd::DormancyStatus

#### 8.200.1.5 BYTE DUNCallInfoInd::MdmConnStatus

#### 8.200.1.6 ULONGLONG DUNCallInfoInd::RXOKBytesCount

#### 8.200.1.7 ULONGLONG DUNCallInfoInd::TXOKBytesCount

## 8.201 dunchannelRate Struct Reference

### Data Fields

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

### 8.201.1 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. |

### 8.201.2 Field Documentation

#### 8.201.2.1 uint32_t dunchannelRate::CurrChanRxRate

#### 8.201.2.2 uint32_t dunchannelRate::CurrChanTxRate

#### 8.201.2.3 uint32_t dunchannelRate::MaxChanRxRate

#### 8.201.2.4 uint32_t dunchannelRate::MaxChanTxRate

## 8.202 ecioListElement Struct Reference

### Data Fields

- SHORT ecio
- BYTE radioIf

### 8.202.1 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 - cdma2000Â® 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 |

### 8.202.2 Field Documentation

#### 8.202.2.1 SHORT ecioListElement::ecio

#### 8.202.2.2 BYTE ecioListElement::radioIf

## 8.203 ECIOThresh Struct Reference

### Data Fields

- BYTE ECIOThresListLen
- SHORT * pECIOThresList

### 8.203.1 Detailed Description

This structure contains ECIO threshold related parameters.

**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 |

### 8.203.2 Field Documentation

#### 8.203.2.1 BYTE ECIOThresh::ECIOThresListLen

#### 8.203.2.2 SHORT* ECIOThresh::pECIOThresList

## 8.204 ECTNum Struct Reference

### Data Fields

- BYTE ECTCallState
- BYTE presentationInd
- BYTE number [81]

### 8.204.1 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 |

### 8.204.2 Field Documentation

#### 8.204.2.1 BYTE ECTNum::ECTCallState

#### 8.204.2.2 BYTE ECTNum::number[81]

#### 8.204.2.3 BYTE ECTNum::presentationInd

## 8.205 ehrpdMTUSizeTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- WORD ehrpdMTUSize

### 8.205.1 Detailed Description

This structure contains the parameters for EHRPD MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *ehrpdMTUSize* | EHRPD size<br>• Size of EHRPD MTU |

### 8.205.2 Field Documentation

#### 8.205.2.1 WORD ehrpdMTUSizeTlv::ehrpdMTUSize

#### 8.205.2.2 BYTE ehrpdMTUSizeTlv::TlvPresent

## 8.206 encryptedPIN1 Struct Reference

### Data Fields

- BYTE pin1Len
- BYTE pin1Val [255]

### 8.206.1 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.

### 8.206.2 Field Documentation

#### 8.206.2.1 BYTE encryptedPIN1::pin1Len

#### 8.206.2.2 BYTE encryptedPIN1::pin1Val[255]

## 8.207 eriDataparams Struct Reference

### Data Fields

- uint16_t eriDataLen
- uint8_t eriData [1024]

### 8.207.1 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 ) |

### 8.207.2 Field Documentation

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

#### 8.207.2.2 uint16_t eriDataparams::eriDataLen

## 8.208 ERIFileparams Struct Reference

### Data Fields

- WORD * pFileSize
- BYTE * pFile

### 8.208.1 Detailed Description

This structure contains Extended Roaming Indicator(ERI) file parameters

**Parameters**

| | |
|---|---|
| *pFileSize[IN/OUT]* | • 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 |
| *pFile[OUT]* | • ERI data read from persistent storage( Max size is 1024 ) |

### 8.208.2 Field Documentation

#### 8.208.2.1 BYTE* ERIFileparams::pFile

#### 8.208.2.2 WORD* ERIFileparams::pFileSize

## 8.209 errorRateListElement Struct Reference

### Data Fields

- USHORT errorRate
- BYTE radioIf

### 8.209.1 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 - cdma2000Â® 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 |

### 8.209.2 Field Documentation

#### 8.209.2.1 USHORT errorRateListElement::errorRate

#### 8.209.2.2 BYTE errorRateListElement::radioIf

## 8.210 eTWSPLMNInfoTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSEtwsPlmnInfo ETWSPLMNInfo

### 8.210.1 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 |

### 8.210.2 Field Documentation

#### 8.210.2.1 sMSEtwsPlmnInfo eTWSPLMNInfoTlv::ETWSPLMNInfo

#### 8.210.2.2 uint8_t eTWSPLMNInfoTlv::TlvPresent

## 8.211 extDispRecInfo Struct Reference

### Data Fields

- BYTE dispType
- BYTE extDispInfoLen
- BYTE extDispInfo [255]

### 8.211.1 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. |

### 8.211.2 Field Documentation

#### 8.211.2.1 BYTE extDispRecInfo::dispType

#### 8.211.2.2 BYTE extDispRecInfo::extDispInfo[255]

#### 8.211.2.3 BYTE extDispRecInfo::extDispInfoLen

## 8.212 FactorySequenceNumber Struct Reference

### Data Fields

- BYTE FSNumber [255]

#### 8.212.1 Detailed Description

This structure used to store Factory Sequence Number parameter

**Parameters**

| | |
|---|---|
| *FSNumber[OUT]* | • Facorty Sequence Number<br>• Maximum Length is 255 Bytes |

#### 8.212.2 Field Documentation

##### 8.212.2.1 BYTE FactorySequenceNumber::FSNumber[255]

## 8.213 fileAttributes Struct Reference

### Data Fields

- WORD fileSize
- WORD fileID
- BYTE fileType
- WORD recordSize
- WORD recordCount
- BYTE secRead
- WORD secReadMask
- BYTE secWrite
- WORD secWriteMask
- BYTE secIncrease
- WORD secIncreaseMask
- BYTE secDeactivate
- WORD secDeactivateMask
- BYTE secActivate
- WORD secActivateMask
- WORD rawLen
- BYTE rawValue [255]

#### 8.213.1 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. |

### 8.213.2 Field Documentation

#### 8.213.2.1 WORD fileAttributes::fileID

#### 8.213.2.2 WORD fileAttributes::fileSize

#### 8.213.2.3 BYTE fileAttributes::fileType

#### 8.213.2.4 WORD fileAttributes::rawLen

#### 8.213.2.5 BYTE fileAttributes::rawValue[255]

#### 8.213.2.6 WORD fileAttributes::recordCount

#### 8.213.2.7 WORD fileAttributes::recordSize

#### 8.213.2.8 BYTE fileAttributes::secActivate

#### 8.213.2.9 WORD fileAttributes::secActivateMask

#### 8.213.2.10 BYTE fileAttributes::secDeactivate

#### 8.213.2.11 WORD fileAttributes::secDeactivateMask

#### 8.213.2.12 BYTE fileAttributes::secIncrease

#### 8.213.2.13 WORD fileAttributes::secIncreaseMask

#### 8.213.2.14 BYTE fileAttributes::secRead

#### 8.213.2.15 WORD fileAttributes::secReadMask

#### 8.213.2.16 BYTE fileAttributes::secWrite

#### 8.213.2.17 WORD fileAttributes::secWriteMask

## 8.214 fileInfo Struct Reference

### Data Fields

- WORD fileID
- BYTE pathLen
- WORD path [255]

### 8.214.1 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 and (0x3F00 ,0x7F20) is for GSM. |

### 8.214.2 Field Documentation

#### 8.214.2.1 WORD fileInfo::fileID

#### 8.214.2.2 WORD fileInfo::path[255]

#### 8.214.2.3 BYTE fileInfo::pathLen

## 8.215 FirmwareUpdatStat Struct Reference

### Data Fields

- ULONG ResCode
- BYTE * pImgType
- ULONG * pRefData
- BYTE * pRefStringLen
- BYTE * pRefString
- BYTE * pLogStringLen
- BYTE * pLogString

### 8.215.1 Detailed Description

This structure is used to store Firmware Update Status

**Parameters**

| | |
|---|---|
| *ResCode[OUT]* | • 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 $\sim$ 0FF - Insignia defined error code<br>* 100 $\sim$ 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 |
| *pImgType[OUT]* | • Optional parameter<br>• Firmware image type that failed the update |
| *pRefData[OUT]* | • Optional parameter<br>• Failed image reference data<br>• This is normally the offset of the image that caused the failure |
| *pRefStringLen[IN/OUT]* | • Length of Reference String parameter to follow<br>• As input parameter specifies length assigned to pRefString parameter<br>• As output parameter specifies length of actual value retrieved from the device |
| *pRefString[OUT]* | • Optional parameter<br>• Failed image reference string. This is normally the partition name of the image that caused the failure if applicable. |
| *pLogStringLen[IN/OUT]* | • Length of Reference String parameter to follow<br>• As input parameter specifies length assigned to pRefString parameter<br>• As output parameter specifies length of actual value retrieved from the device |
| *pLogString[OUT]* | • Optional parameter<br>• Failed image reference string. This is normally the partition name of the image that caused the failure if applicable. |

## 8.215.2 Field Documentation

### 8.215.2.1 BYTE* FirmwareUpdatStat::pImgType

#### 8.215.2.2 BYTE* FirmwareUpdatStat::pLogString

#### 8.215.2.3 BYTE* FirmwareUpdatStat::pLogStringLen

#### 8.215.2.4 ULONG* FirmwareUpdatStat::pRefData

#### 8.215.2.5 BYTE* FirmwareUpdatStat::pRefString

#### 8.215.2.6 BYTE* FirmwareUpdatStat::pRefStringLen

#### 8.215.2.7 ULONG FirmwareUpdatStat::ResCode

## 8.216 FMSImageElement Struct Reference

### Data Fields

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

### 8.216.1 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 |

### 8.216.2 Field Documentation

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

#### 8.216.2.2 uint8_t FMSImageElement::buildIdLength

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

#### 8.216.2.4 uint8_t FMSImageElement::imageType

## 8.217 FMSImageIdElement Struct Reference

### Data Fields

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

### 8.217.1 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 ) |

### 8.217.2 Field Documentation

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

#### 8.217.2.2 uint8_t FMSImageIdElement::buildIDLength

#### 8.217.2.3 uint8_t FMSImageIdElement::failureCount

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

#### 8.217.2.5 uint8_t FMSImageIdElement::storageIndex

## 8.218 FMSImageIDEntries Struct Reference

### Data Fields

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

### 8.218.1 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 |

### 8.218.2 Field Documentation

#### 8.218.2.1 uint8_t FMSImageIDEntries::executingImage

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

#### 8.218.2.3 uint8_t FMSImageIDEntries::imageIDSize

#### 8.218.2.4 uint8_t FMSImageIDEntries::imageType

#### 8.218.2.5 uint8_t FMSImageIDEntries::maxImages

## 8.219 FMSImageList Struct Reference

### Data Fields

- uint8_t listSize
- FMSImageIDEntries imageIDEntries [2]

### 8.219.1 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 ) |

### 8.219.2 Field Documentation

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

#### 8.219.2.2 uint8_t FMSImageList::listSize

## 8.220 FMSPrefImageList Struct Reference

### Data Fields

- uint8_t listSize
- FMSImageElement listEntries [2]

### 8.220.1 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 |

### 8.220.2 Field Documentation

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

#### 8.220.2.2 uint8_t FMSPrefImageList::listSize

## 8.221 fwinfo_s Struct Reference

### Data Fields

- ULONG FirmwareID
- ULONG Technology
- ULONG Carrier
- ULONG Region
- ULONG GPSCapability

### 8.221.1 Detailed Description

Gobi firmware image info structure

**Parameters**

| | |
|---|---|
| *FirmwareID* | • Firmware ID obtained from the firmware image |
| *Technology* | • Technology (0xFFFFFFFF if unknown) |
| *Carrier* | • Carrier (0xFFFFFFFF if unknown) |
| *Region* | • Region (0xFFFFFFFF if unknown) |
| *GPSCapability* | • GPS capability (0xFFFFFFFF if unknown) |

### 8.221.2 Field Documentation

#### 8.221.2.1 ULONG fwinfo_s::Carrier

#### 8.221.2.2 ULONG fwinfo_s::FirmwareID

#### 8.221.2.3 ULONG fwinfo_s::GPSCapability

#### 8.221.2.4 ULONG fwinfo_s::Region

#### 8.221.2.5 ULONG fwinfo_s::Technology

## 8.222 GERANInfo Struct Reference

### Data Fields

- ULONG cellID
- BYTE plmn [3]
- WORD lac
- WORD arfcn
- BYTE bsic
- ULONG timingAdvance
- WORD rxLev
- BYTE nmrInst
- nmrCellInfo insNmrCellInfo [255]

### 8.222.1 Detailed Description

This structure contains information about the GERAN Network.

**Parameters**

| | |
|---|---|
| *cellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present. |

| | |
|---|---|
| *plmn[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[MAX_DESCRIPTION_LENGTH]* | • See nmrCellInfo for more information. |

## 8.222.2 Field Documentation

### 8.222.2.1 WORD GERANInfo::arfcn

### 8.222.2.2 BYTE GERANInfo::bsic

### 8.222.2.3 ULONG GERANInfo::cellID

#### 8.222.2.4 nmrCellInfo GERANInfo::insNmrCellInfo[255]

#### 8.222.2.5 WORD GERANInfo::lac

#### 8.222.2.6 BYTE GERANInfo::nmrInst

#### 8.222.2.7 BYTE GERANInfo::plmn[3]

#### 8.222.2.8 WORD GERANInfo::rxLev

#### 8.222.2.9 ULONG GERANInfo::timingAdvance

## 8.223 geranInstInfo Struct Reference

### Data Fields

- WORD geranArfcn
- BYTE geranBsicNcc
- BYTE geranBsicBcc
- SHORT geranRssi

### 8.223.1 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. |

### 8.223.2 Field Documentation

#### 8.223.2.1 WORD geranInstInfo::geranArfcn

#### 8.223.2.2 BYTE geranInstInfo::geranBsicBcc

#### 8.223.2.3 BYTE geranInstInfo::geranBsicNcc

#### 8.223.2.4 SHORT geranInstInfo::geranRssi

## 8.224 getAllCallInformation Struct Reference

### Data Fields

- callInfo Callinfo
- BYTE isEmpty
- BYTE ALS

### 8.224.1 Detailed Description

This structure contains information related to call state change.

**Parameters**

| | |
|---|---|
| *Callinfo* | • See 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 |

### 8.224.2 Field Documentation

#### 8.224.2.1 BYTE getAllCallInformation::ALS

#### 8.224.2.2 callInfo getAllCallInformation::Callinfo

#### 8.224.2.3 BYTE getAllCallInformation::isEmpty

## 8.225 getAllCallRmtPtyName Struct Reference

### Data Fields

- BYTE callID
- remotePartyName RemotePartyName

### 8.225.1 Detailed Description

This structure contains information for All Call Remote Party Names

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |

| | |
|---|---|
| *RemoteParty-Name* | • See remotePartyName for more information. |

### 8.225.2 Field Documentation

#### 8.225.2.1 BYTE getAllCallRmtPtyName::callID

#### 8.225.2.2 remotePartyName getAllCallRmtPtyName::RemotePartyName

## 8.226 getAllCallRmtPtyNum Struct Reference

### Data Fields

- BYTE callID
- remotePartyNum RemotePartyNum

### 8.226.1 Detailed Description

This structure contains information for All Call Remote Party Numbers

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *RemoteParty-Num* | • See remotePartyNum for more information. |

### 8.226.2 Field Documentation

#### 8.226.2.1 BYTE getAllCallRmtPtyNum::callID

#### 8.226.2.2 remotePartyNum getAllCallRmtPtyNum::RemotePartyNum

## 8.227 GetAudioPathConfigReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE Item

### 8.227.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig request parameters

**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 |

### 8.227.2 Field Documentation

#### 8.227.2.1 BYTE GetAudioPathConfigReq::Item

#### 8.227.2.2 BYTE GetAudioPathConfigReq::Profile

## 8.228 GetAudioPathConfigResp Struct Reference

### Data Fields

- BYTE * pECMode
- BYTE * pNSEnable
- WORD * pTXGain
- WORD * pDTMFTXGain
- WORD * pCodecSTGain
- TXPCMIIRFltr * pTXPCMIIRFltr
- RXPCMIIRFltr * pRXPCMIIRFltr
- BYTE * pMICGainSelect
- BYTE * pRXAVCAGCSwitch
- BYTE * pTXAVCSwitch
- RXAGCList * pRXAGCList
- RXAVCList * pRXAVCList
- TXAGCList * pTXAGCList

### 8.228.1 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 |
| *pNSEnable* | [Optional]<br>• AV_NS<br>– 0 - Noise suppression off<br>– 1 - Noise suppression on |
| *pTXGain* | [Optional]<br>• AV_TXVOL<br>– 0x0000 - 0xffff |
| *pDTMFTXGain* | [Optional]<br>• AV_DTMFTXG<br>– 0x0000 - 0xffff |
| *pCodecSTGain* | [Optional]<br>• AV_CODECSTG<br>– 0x0000 - 0xffff |
| *pTXPCMIIRFltr* | [Optional]<br>• See TXPCMIIRFltr for more information |
| *pRXPCMIIRFltr* | [Optional]<br>• See RXPCMIIRFltr for more information |
| *pMICGainSelect* | [Optional]<br>• AV_MICGAIN |
| *pRXAVCAGC-Switch* | [Optional]<br>• RX AVC/AGC Switch |
| *pTXAVCSwitch* | [Optional]<br>• TX AVC Switch |
| *pRXAGCList* | [Optional]<br>• See RXAGCList for more information |
| *pRXAVCList* | [Optional]<br>• See RXAVCList for more information |
| *pTXAGCList* | [Optional]<br>• See TXAGCList for more information |

### 8.228.2 Field Documentation

#### 8.228.2.1 WORD* GetAudioPathConfigResp::pCodecSTGain

#### 8.228.2.2 WORD* GetAudioPathConfigResp::pDTMFTXGain

#### 8.228.2.3 BYTE* GetAudioPathConfigResp::pECMode

#### 8.228.2.4 BYTE* GetAudioPathConfigResp::pMICGainSelect

#### 8.228.2.5 BYTE* GetAudioPathConfigResp::pNSEnable

#### 8.228.2.6 RXAGCList* GetAudioPathConfigResp::pRXAGCList

#### 8.228.2.7 BYTE* GetAudioPathConfigResp::pRXAVCAGCSwitch

#### 8.228.2.8 RXAVCList* GetAudioPathConfigResp::pRXAVCList

#### 8.228.2.9 RXPCMIIRFltr* GetAudioPathConfigResp::pRXPCMIIRFltr

#### 8.228.2.10 TXAGCList* GetAudioPathConfigResp::pTXAGCList

#### 8.228.2.11 BYTE* GetAudioPathConfigResp::pTXAVCSwitch

#### 8.228.2.12 WORD* GetAudioPathConfigResp::pTXGain

#### 8.228.2.13 TXPCMIIRFltr* GetAudioPathConfigResp::pTXPCMIIRFltr

## 8.229 GetAudioProfileReq Struct Reference

### Data Fields

- BYTE Generator

### 8.229.1 Detailed Description

This structure contains the SLQSGetAudioProfile request parameters

**Parameters**

| | |
|---|---|
| *Generator* | • Audio Generator<br>– 0 - Voice<br>– 1 - Key Beep<br>– 2 - MIDI |

### 8.229.2 Field Documentation

#### 8.229.2.1 BYTE GetAudioProfileReq::Generator

## 8.230 GetAudioProfileResp Struct Reference

### Data Fields

- BYTE Profile
- BYTE EarMute
- BYTE MicMute
- BYTE Volume

### 8.230.1 Detailed Description

This structure contains the SLQSGetAudioProfile response parameters.

**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 |
| *Volume* | • Audio Volume Level<br>– 0 to 7 |

### 8.230.2 Field Documentation

#### 8.230.2.1 BYTE GetAudioProfileResp::EarMute

#### 8.230.2.2 BYTE GetAudioProfileResp::MicMute

#### 8.230.2.3 BYTE GetAudioProfileResp::Profile

#### 8.230.2.4 BYTE GetAudioProfileResp::Volume

## 8.231 GetAudioVolTLBConfigReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Volume
- BYTE Item

### 8.231.1 Detailed Description

This structure contains the SLQSGetAudioVolTLBConfig request parameters

**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 |

### 8.231.2 Field Documentation

#### 8.231.2.1 BYTE GetAudioVolTLBConfigReq::Generator

#### 8.231.2.2 BYTE GetAudioVolTLBConfigReq::Item

#### 8.231.2.3 BYTE GetAudioVolTLBConfigReq::Profile

#### 8.231.2.4 BYTE GetAudioVolTLBConfigReq::Volume

## 8.232 GetAudioVolTLBConfigResp Struct Reference

### Data Fields

- WORD ResCode

### 8.232.1 Detailed Description

This structure contains the SLQSGetAudioVolTLBConfig response parameters.

**Parameters**

| | |
|---|---|
| *ResCode* | • Result of requested item |

### 8.232.2 Field Documentation

#### 8.232.2.1 WORD GetAudioVolTLBConfigResp::ResCode

## 8.233 getCallFWExtInfo Struct Reference

### Data Fields

- BYTE numInstances
- callFWExtInfo CallFWExtInfo [20]

### 8.233.1 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[-MAX_NO_OF_-CALLS]* | • Array of CallFWExtInfo.<br>– See CallFWExtInfo for more information. |

### 8.233.2 Field Documentation

#### 8.233.2.1 callFWExtInfo getCallFWExtInfo::CallFWExtInfo[20]

#### 8.233.2.2 BYTE getCallFWExtInfo::numInstances

## 8.234 getCallFWInfo Struct Reference

### Data Fields

- BYTE numInstances
- callFWInfo CallFWInfo [20]

### 8.234.1 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[MAX_NO_OF_CALLS]* | • Array of callFWInfo.<br>– See callFWInfo for more information. |

### 8.234.2 Field Documentation

#### 8.234.2.1 callFWInfo getCallFWInfo::CallFWInfo[20]

#### 8.234.2.2 BYTE getCallFWInfo::numInstances

## 8.235 getCustomFeatureV2 Struct Reference

### Data Fields

- getCustomInput * pGetCustomInput
- custSettingInfo * pCustSettingInfo
- custSettingList * pCustSettingList

### 8.235.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pGetCustomInput[IN]* | • Optional parameter<br>• See getCustomInput for more information |
| *pCustSettingInfo[OUT]* | • Optional parameter<br>• See custSettingInfo for more information |
| *pCustSettingList[OUT]* | • Optional parameter<br>• See custSettingList for more information |

### 8.235.2 Field Documentation

#### 8.235.2.1 custSettingInfo* getCustomFeatureV2::pCustSettingInfo

#### 8.235.2.2 custSettingList* getCustomFeatureV2::pCustSettingList

#### 8.235.2.3 getCustomInput* getCustomFeatureV2::pGetCustomInput

## 8.236 getCustomInput Struct Reference

### Data Fields

- CHAR cust_id [64+1]
- BYTE list_type

### 8.236.1 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) |
| *list_type* | • list type requested |

### 8.236.2 Field Documentation

#### 8.236.2.1 CHAR getCustomInput::cust_id[64+1]

#### 8.236.2.2 BYTE getCustomInput::list_type

## 8.237 getDUNCallInfoReq Struct Reference

### Data Fields

- ULONG Mask
- BYTE * pReportConnStatus
- TransferStatInd * pTransferStatInd
- BYTE * pReportDormStatus
- BYTE * pReportDataBearerTech
- BYTE * pReportChannelRate

### 8.237.1 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 |
| *pReportConn-Status* | • Connect Status Indicator<br>– 0 - Do not report<br>– 1 - Report connection status and call end reason |
| *pTransferStatInd* | • See 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 |

## 8.237.2 Field Documentation

### 8.237.2.1 ULONG getDUNCallInfoReq::Mask

### 8.237.2.2 BYTE* getDUNCallInfoReq::pReportChannelRate

### 8.237.2.3 BYTE* getDUNCallInfoReq::pReportConnStatus

### 8.237.2.4 BYTE* getDUNCallInfoReq::pReportDataBearerTech

### 8.237.2.5 BYTE* getDUNCallInfoReq::pReportDormStatus

### 8.237.2.6 TransferStatInd* getDUNCallInfoReq::pTransferStatInd

## 8.238 getDUNCallInfoResp Struct Reference

### Data Fields

- ConnectionStatus * pConnectionStatus
- WORD * pCallEndReason
- ULONGLONG * pTXOKBytesCount
- ULONGLONG * pRXOKBytesCount
- BYTE * pDormancyStatus
- BYTE * pDataBearerTech
- ChannelRate * pChannelRate
- ULONGLONG * pLastCallTXOKBytesCnt
- ULONGLONG * pLastCallRXOKBytesCnt
- ULONGLONG * pMdmCallDurationActive
- BYTE * pLastCallDataBearerTech

### 8.238.1 Detailed Description

This structure contains the DUN Call Info response parameters

**Parameters**

| | |
|---|---|
| *pConnection-Status* | • See ConnectionStatus for more information |
| *pCallEndReason* | • 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 |
| *pTXOKBytes-Count* | • Number of bytes transmitted without error<br>• Returned only if a data call is up |
| *pRXOKBytes-Count* | • Number of bytes received without error<br>• Returned only if a data call is up |
| *pDormancy-Status* | • Current traffic channel status<br>• Returned if a data call is up<br>– 0x01 - Traffic channel dormant<br>– 0x02 - Traffic channel active |

| | |
|---|---|
| *pDataBearer-Tech* | • 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 |
| *pChannelRate* | • See ChannelRate for more information |
| *pLastCallTXOK-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. |
| *pLastCallRXOK-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. |
| *pMdmCall-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) |

| | |
|---|---|
| *pLastCallData-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 |

### 8.238.2 Field Documentation

#### 8.238.2.1 WORD* getDUNCallInfoResp::pCallEndReason

#### 8.238.2.2 ChannelRate* getDUNCallInfoResp::pChannelRate

#### 8.238.2.3 ConnectionStatus* getDUNCallInfoResp::pConnectionStatus

#### 8.238.2.4 BYTE* getDUNCallInfoResp::pDataBearerTech

#### 8.238.2.5 BYTE* getDUNCallInfoResp::pDormancyStatus

#### 8.238.2.6 BYTE* getDUNCallInfoResp::pLastCallDataBearerTech

#### 8.238.2.7 ULONGLONG* getDUNCallInfoResp::pLastCallRXOKBytesCnt

#### 8.238.2.8 ULONGLONG* getDUNCallInfoResp::pLastCallTXOKBytesCnt

#### 8.238.2.9 ULONGLONG* getDUNCallInfoResp::pMdmCallDurationActive

#### 8.238.2.10 ULONGLONG* getDUNCallInfoResp::pRXOKBytesCount

#### 8.238.2.11 ULONGLONG* getDUNCallInfoResp::pTXOKBytesCount

## 8.239 getDyingGaspCfg Struct Reference

### Data Fields

- BYTE * pDestSMSNum
- BYTE * pDestSMSContent

### 8.239.1 Detailed Description

This struture contains the TLV required to get the Dysing GASP Config.

**Parameters**

| | |
|---|---|
| *pDestSMSNum[-OUT]* | • SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.<br>• Optional parameter. |
| *pDestSMS-Content[OUT]* | • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars.<br>• Optional parameter. |

### 8.239.2 Field Documentation

#### 8.239.2.1 BYTE* getDyingGaspCfg::pDestSMSContent

#### 8.239.2.2 BYTE* getDyingGaspCfg::pDestSMSNum

## 8.240 getDyingGaspStatistics Struct Reference

### Data Fields

- ULONG * pTimeStamp
- BYTE * pSMSAttemptedFlag

### 8.240.1 Detailed Description

This struture contains the TLV required to get the Dysing GASP Statistics.

**Parameters**

| | |
|---|---|
| *pTimeStamp[O-UT]* | • 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). |
| *pSMS-AttemptedFlag[-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. |

### 8.240.2 Field Documentation

#### 8.240.2.1 BYTE* getDyingGaspStatistics::pSMSAttemptedFlag

#### 8.240.2.2 ULONG* getDyingGaspStatistics::pTimeStamp

## 8.241 GetErrRateResp Struct Reference

### Data Fields

- WORD * pCDMAFrameErrRate
- WORD * pHDRPackErrRate
- BYTE * pGSMBER
- BYTE * pWCDMABER

### 8.241.1 Detailed Description

This structure contains information about the SLQSGetErrorRate response parameters.

**Parameters**

| | |
|---|---|
| *pCDMAFrameErrRate[Out]* | • 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. |
| *pHDRPackErrRate[Out]* | • 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. |
| *pGSMBER[Out]* | • 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. |
| *pWCDMABER[Out]* | • 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. |

### 8.241.2 Field Documentation

#### 8.241.2.1 WORD* GetErrRateResp::pCDMAFrameErrRate

#### 8.241.2.2 BYTE* GetErrRateResp::pGSMBER

#### 8.241.2.3 WORD* GetErrRateResp::pHDRPackErrRate

#### 8.241.2.4 BYTE* GetErrRateResp::pWCDMABER

## 8.242 GetHRPDStatsResp Struct Reference

### Data Fields

- DRCParams * pDRCParams
- BYTE * pUATI
- PilotSetData * pPilotSetData

### 8.242.1 Detailed Description

This structure contains information about the SLQSSwiGetHRPDStats response parameters.

**Parameters**

| | |
|---|---|
| *pDRCParams[-Out]* | • See DRCParams for more information. |
| *pUATI[Out]* | • A 128-bit address that includes the access terminal identifier and subnet ID<br>• Size must be 16 bytes |
| *pPilotSetData[-Out]* | • See PilotSetData for more information. |

### 8.242.2 Field Documentation

#### 8.242.2.1 DRCParams* GetHRPDStatsResp::pDRCParams

#### 8.242.2.2 PilotSetData* GetHRPDStatsResp::pPilotSetData

#### 8.242.2.3 BYTE* GetHRPDStatsResp::pUATI

## 8.243 GetIMSSMSConfigParams Struct Reference

### Data Fields

- BYTE * pSettingResp
- BYTE * pSMSFormat
- BYTE * pSMSOverIPNwInd
- BYTE * pPhoneCtxtURILen
- BYTE * pPhoneCtxtURI

### 8.243.1 Detailed Description

This structure contains the SLQSGetIMSSMSConfig response parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response |

| | |
|---|---|
| *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[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 |

### 8.243.2 Field Documentation

#### 8.243.2.1 BYTE* GetIMSSMSConfigParams::pPhoneCtxtURI

#### 8.243.2.2 BYTE* GetIMSSMSConfigParams::pPhoneCtxtURILen

#### 8.243.2.3 BYTE* GetIMSSMSConfigParams::pSettingResp

#### 8.243.2.4 BYTE* GetIMSSMSConfigParams::pSMSFormat

#### 8.243.2.5 BYTE* GetIMSSMSConfigParams::pSMSOverIPNwInd

## 8.244 GetIMSUserConfigParams Struct Reference

### Data Fields

- BYTE * pSettingResp
- BYTE * pIMSDomainLen
- BYTE * pIMSDomain

### 8.244.1 Detailed Description

This structure contains the SLQSGetIMSUserConfig response parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response |
| *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 |

### 8.244.2 Field Documentation

#### 8.244.2.1 BYTE* GetIMSUserConfigParams::pIMSDomain

#### 8.244.2.2 BYTE* GetIMSUserConfigParams::pIMSDomainLen

#### 8.244.2.3 BYTE* GetIMSUserConfigParams::pSettingResp

## 8.245 GetIMSVoIPConfigResp Struct Reference

### Data Fields

- BYTE * pSettingResp
- WORD * pSessionExpiryTimer
- WORD * pMinSessionExpiryTimer
- BYTE * pAmrWbEnable
- BYTE * pScrAmrEnable
- BYTE * pScrAmrWbEnable
- BYTE * pAmrMode
- WORD * pAmrWBMode
- BYTE * pAmrOctetAligned
- BYTE * pAmrWBOctetAligned
- WORD * pRingingTimer
- WORD * pRingBackTimer
- WORD * pRTPRTCPInactTimer

### 8.245.1 Detailed Description

This structure contains the SLQSGetIMSVoIPConfig request parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE |
| *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. |

### 8.245.2 Field Documentation

#### 8.245.2.1 BYTE* GetIMSVoIPConfigResp::pAmrMode

#### 8.245.2.2 BYTE* GetIMSVoIPConfigResp::pAmrOctetAligned

#### 8.245.2.3 BYTE* GetIMSVoIPConfigResp::pAmrWbEnable

#### 8.245.2.4 WORD* GetIMSVoIPConfigResp::pAmrWBMode

#### 8.245.2.5 BYTE* GetIMSVoIPConfigResp::pAmrWBOctetAligned

#### 8.245.2.6 WORD* GetIMSVoIPConfigResp::pMinSessionExpiryTimer

#### 8.245.2.7 WORD* GetIMSVoIPConfigResp::pRingBackTimer

#### 8.245.2.8 WORD* GetIMSVoIPConfigResp::pRingingTimer

#### 8.245.2.9 WORD* GetIMSVoIPConfigResp::pRTPRTCPInactTimer

#### 8.245.2.10 BYTE* GetIMSVoIPConfigResp::pScrAmrEnable

#### 8.245.2.11 BYTE* GetIMSVoIPConfigResp::pScrAmrWbEnable

#### 8.245.2.12 WORD* GetIMSVoIPConfigResp::pSessionExpiryTimer

#### 8.245.2.13 BYTE* GetIMSVoIPConfigResp::pSettingResp

## 8.246 GetInstIDResp Struct Reference

### Data Fields

- BYTE * pInstanceID
- BYTE * pIPFamily

### 8.246.1 Field Documentation

#### 8.246.1.1 BYTE* GetInstIDResp::pInstanceID

#### 8.246.1.2 BYTE* GetInstIDResp::pIPFamily

## 8.247 GetM2MAudioProfileReq Struct Reference

### Data Fields

- BYTE * pGenerator

### 8.247.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile request parameters.

**Parameters**

| | |
|---|---|
| *pGenerator[optional]* | • Generator<br>– 0 - Voice |

### 8.247.2 Field Documentation

#### 8.247.2.1 BYTE* GetM2MAudioProfileReq::pGenerator

## 8.248 GetM2MAudioProfileResp Struct Reference

### Data Fields

- BYTE Profile
- BYTE EarMute
- BYTE MicMute
- BYTE Generator
- BYTE Volume
- BYTE CwtMute

### 8.248.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile response parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0-5 |
| *EarMute* | • Ear Mute<br>– 0 - Mute<br>– 1 - Unmute |

| | |
|---|---|
| *MicMute* | • MIC Mute<br>– 0 - Mute<br>– 1 - Unmute |
| *Generator* | • Generator<br>– 0 - Voice |
| *Volume* | • RX volume level<br>– 0-5 |
| *CwtMute* | • Call waiting tone Mute<br>– 0 - Mute<br>– 1 - Unmute |

### 8.248.2 Field Documentation

#### 8.248.2.1 BYTE GetM2MAudioProfileResp::CwtMute

#### 8.248.2.2 BYTE GetM2MAudioProfileResp::EarMute

#### 8.248.2.3 BYTE GetM2MAudioProfileResp::Generator

#### 8.248.2.4 BYTE GetM2MAudioProfileResp::MicMute

#### 8.248.2.5 BYTE GetM2MAudioProfileResp::Profile

#### 8.248.2.6 BYTE GetM2MAudioProfileResp::Volume

## 8.249 GetM2MAudioVolumeReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE Generator

### 8.249.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume request parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0-5 |

| | |
|---|---|
| *Generator* | • Generator<br>– 0 - Voice |

### 8.249.2 Field Documentation

#### 8.249.2.1 BYTE GetM2MAudioVolumeReq::Generator

#### 8.249.2.2 BYTE GetM2MAudioVolumeReq::Profile

## 8.250 GetM2MAudioVolumeResp Struct Reference

### Data Fields

- BYTE Level

### 8.250.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume response parameters.

**Parameters**

| | |
|---|---|
| *Level* | • The RX Volume Level<br>– 0-5 |

### 8.250.2 Field Documentation

#### 8.250.2.1 BYTE GetM2MAudioVolumeResp::Level

## 8.251 GetM2MAVMuteReq Struct Reference

### Data Fields

- BYTE Profile

### 8.251.1 Detailed Description

This structure contains the SLQSGetM2MAVMute request parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |

### 8.251.2 Field Documentation

#### 8.251.2.1 BYTE GetM2MAVMuteReq::Profile

## 8.252 GetM2MAVMuteResp Struct Reference

### Data Fields

- BYTE EarMute
- BYTE MicMute
- BYTE CwtMute

### 8.252.1 Detailed Description

This structure contains the SLQSGetM2MAVMute response parameters.

**Parameters**

| | |
|---|---|
| *pEarMute* | • Ear Mute<br>– 0-Mute<br>– 1-UnMute |
| *pMicMute* | • Mic Mute<br>– 0-Mute<br>– 1-unmute |
| *CwtMute* | • Waiting tone Mute<br>– 0-5 |

### 8.252.2 Field Documentation

#### 8.252.2.1 BYTE GetM2MAVMuteResp::CwtMute

#### 8.252.2.2 BYTE GetM2MAVMuteResp::EarMute

#### 8.252.2.3 BYTE GetM2MAVMuteResp::MicMute

## 8.253 GetM2MSpkrGainReq Struct Reference

### Data Fields

- BYTE Profile

### 8.253.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain request parameters.

**Parameters**

| | |
|---|---|
| *pProfile* | • Audio Profile Number<br>– 0-5 |

### 8.253.2 Field Documentation

#### 8.253.2.1 BYTE GetM2MSpkrGainReq::Profile

## 8.254 GetM2MSpkrGainResp Struct Reference

### Data Fields

- WORD Value

### 8.254.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain response parameters.

**Parameters**

| | |
|---|---|
| *Value* | • RX speakerphone gain<br>– 0x0 - 0x7fff |

### 8.254.2 Field Documentation

#### 8.254.2.1 WORD GetM2MSpkrGainResp::Value

## 8.255 getMsgWaitingInfo Struct Reference

### Data Fields

- BYTE numInstances
- messageWaitingInfoContent msgWaitInfo [0xFF]

### 8.255.1 Detailed Description

This structure contains Get Message Waiting Info Response parameters

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of sets of the elements in structure messageWaitingInfoContent |
| *pMsgWaitInfo* | • Pointer to structure of messageWaitingInfoContent.<br>– See messageWaitingInfoContent for more information. |

### 8.255.2 Field Documentation

#### 8.255.2.1 messageWaitingInfoContent getMsgWaitingInfo::msgWaitInfo[0xFF]

#### 8.255.2.2 BYTE getMsgWaitingInfo::numInstances

## 8.256 GetNetworkTimeResp Struct Reference

### Data Fields

- timeInfo * p3GPP2TimeInfo
- timeInfo * p3GPPTimeInfo

### 8.256.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

**Parameters**

| | |
|---|---|
| *p3GPP2TimeInfo* | [Optional]<br>• See timeInfo for more information |
| *p3GPPTimeInfo* | [Optional]<br>• See timeInfo for more information |

### 8.256.2 Field Documentation

#### 8.256.2.1 timeInfo* GetNetworkTimeResp::p3GPP2TimeInfo

#### 8.256.2.2 timeInfo* GetNetworkTimeResp::p3GPPTimeInfo

## 8.257 GetRegMgrConfigParams Struct Reference

### Data Fields

- BYTE * pSettingResp
- WORD * pPCSCFPort
- BYTE * pPriCSCFPortNameLen
- BYTE * pPriCSCFPortName
- BYTE * pIMSTestMode

### 8.257.1 Detailed Description

This structure contains the SLQSGetRegMgrConfig response parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response |
| *pPCSCFPort* | • Proxy call session control function port |

| | |
|---|---|
| *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 |
| *pIMSTestMode* | • IMS Test mode Enabled.<br>– TRUE - Enabled<br>– FALSE - Disabled |

**Note**

pPriCSCFPortNameLen must be set to a valid value during API call to retrieve pPriCSCFPortName.

### 8.257.2 Field Documentation

#### 8.257.2.1 BYTE* GetRegMgrConfigParams::pIMSTestMode

#### 8.257.2.2 WORD* GetRegMgrConfigParams::pPCSCFPort

#### 8.257.2.3 BYTE* GetRegMgrConfigParams::pPriCSCFPortName

#### 8.257.2.4 BYTE* GetRegMgrConfigParams::pPriCSCFPortNameLen

#### 8.257.2.5 BYTE* GetRegMgrConfigParams::pSettingResp

## 8.258 GetSessionIDResp Struct Reference

### Data Fields

- ULONG * pSessionIDv4
- ULONG * pSessionIDv6

### 8.258.1 Field Documentation

#### 8.258.1.1 ULONG* GetSessionIDResp::pSessionIDv4

#### 8.258.1.2 ULONG* GetSessionIDResp::pSessionIDv6

## 8.259 GetSIPConfigResp Struct Reference

### Data Fields

- BYTE * pSettingResp
- WORD * pSIPLocalPort
- ULONG * pTimerSIPReg
- ULONG * pSubscribeTimer
- ULONG * pTimerT1

- ULONG * pTimerT2
- ULONG * pTimerTf
- BYTE * pSigCompEnabled

### 8.259.1 Detailed Description

This structure contains the SLQSGetSIPConfig response parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response |
| *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 - Sig Comp Enabled<br>– FALSE - Sig Comp Disabled |

### 8.259.2 Field Documentation

#### 8.259.2.1 BYTE* GetSIPConfigResp::pSettingResp

#### 8.259.2.2 BYTE* GetSIPConfigResp::pSigCompEnabled

#### 8.259.2.3 WORD* GetSIPConfigResp::pSIPLocalPort

#### 8.259.2.4 ULONG* GetSIPConfigResp::pSubscribeTimer

#### 8.259.2.5 ULONG* GetSIPConfigResp::pTimerSIPReg

#### 8.259.2.6 ULONG* GetSIPConfigResp::pTimerT1

#### 8.259.2.7 ULONG* GetSIPConfigResp::pTimerT2

#### 8.259.2.8 ULONG* GetSIPConfigResp::pTimerTf

## 8.260 GnssData Struct Reference

### Data Fields

- ULONGLONG mask

### 8.260.1 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 |

### 8.260.2 Field Documentation

#### 8.260.2.1 ULONGLONG GnssData::mask

## 8.261 gnssSvInfoNotification Struct Reference

### Data Fields

- BYTE bAltitudeAssumed
- satelliteInfo * pSatelliteInfo

### 8.261.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

**Parameters**

| | |
|---|---|
| *bAltitude-Assumed* | • 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 |
| *pSatelliteInfo* | • See satelliteInfo for more information. |

**Note**

None

### 8.261.2 Field Documentation

#### 8.261.2.1 BYTE gnssSvInfoNotification::bAltitudeAssumed

#### 8.261.2.2 satelliteInfo* gnssSvInfoNotification::pSatelliteInfo

## 8.262 GPRSQoS Struct Reference

### Data Fields

- ULONG precedenceClass
- ULONG delayClass
- ULONG reliabilityClass
- ULONG peakThroughputClass
- ULONG meanThroughputClass

### 8.262.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

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

**Parameters**

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

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

### 8.262.2 Field Documentation

#### 8.262.2.1 ULONG GPRSQoS::delayClass

#### 8.262.2.2 ULONG GPRSQoS::meanThroughputClass

#### 8.262.2.3 ULONG GPRSQoS::peakThroughputClass

#### 8.262.2.4 ULONG GPRSQoS::precedenceClass

#### 8.262.2.5 ULONG GPRSQoS::reliabilityClass

## 8.263 GPRSRequestedQoS Struct Reference

### Data Fields

- ULONG precedenceClass
- ULONG delayClass
- ULONG reliabilityClass
- ULONG peakThroughputClass
- ULONG meanThroughputClass

### 8.263.1 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 |

| | |
|---|---|
| *meanThroughputClass* | • Mean throughput class |

### 8.263.2 Field Documentation

#### 8.263.2.1 ULONG GPRSRequestedQoS::delayClass

#### 8.263.2.2 ULONG GPRSRequestedQoS::meanThroughputClass

#### 8.263.2.3 ULONG GPRSRequestedQoS::peakThroughputClass

#### 8.263.2.4 ULONG GPRSRequestedQoS::precedenceClass

#### 8.263.2.5 ULONG GPRSRequestedQoS::reliabilityClass

## 8.264 GPSStateInfo Struct Reference

### Data Fields

- BYTE EngineState
- ULONG ValidMask
- ULONGLONG Latitude
- ULONGLONG Longitude
- ULONG HorizontalUncertainty
- ULONG Altitude
- ULONG VerticalUncertainty
- ULONG TimeStmp_tow_ms
- WORD TimeStmp_gps_week
- ULONG Time_uncert_ms
- BYTE Iono_valid
- ULONG gps_ephemeris_sv_msk
- ULONG gps_almanac_sv_msk
- ULONG gps_health_sv_msk
- ULONG gps_visible_sv_msk
- ULONG glo_ephemeris_sv_msk
- ULONG glo_almanac_sv_msk
- ULONG glo_health_sv_msk
- ULONG glo_visible_sv_msk
- ULONG sbas_ephemeris_sv_msk
- ULONG sbas_almanac_sv_msk
- ULONG sbas_health_sv_msk
- ULONG sbas_visible_sv_msk
- WORD xtra_start_gps_week
- WORD xtra_start_gps_minutes
- WORD xtra_valid_duration_hours

### 8.264.1 Detailed Description

GPS state Info.

**Parameters**

| | |
|---|---|
| *EngineState* | • Values:<br>– 0 - OFF<br>– 1 - ON<br>• This field is always valid |
| *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 |
| *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) |
| *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) |
| *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) |

| | |
|---|---|
| *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) |
| *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) |
| *TimeStmp_tow_-ms* | • Time stamp in GPS time of week( in milliseconds) |
| *TimeStmp_gps-_week* | • GPS week number |
| *Time_uncert_ms* | • Time uncertainty (in milliseconds). The uncertainty is provided at 99 percent confidence. |
| *Iono_valid* | • Iono validity.<br>• Values:<br>– 0 - Invalid<br>– 1 - Valid |
| *gps_ephemeris-_sv_msk* | • GPS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. |
| *gps_almanac_-sv_msk* | • GPS SV mask for almanac; if the bit is set, almanac for that SV is available. |
| *gps_health_sv_-msk* | • GPS SV mask for health; if the bit is set, health for that SV is available. |
| *gps_visible_sv_-msk* | • GPS SV mask for visible Svs; if the bit is set, the SV is available. |
| *glo_ephemeris_-sv_msk* | • GLONASS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. |
| *glo_almanac_sv-_msk* | • GLONASS SV mask for almanac; if the bit is set, almanac for that SV is available. |
| *glo_health_sv_-msk* | • GLONASS SV mask for health; if the bit is set, health for that SV is available. |
| *glo_visible_sv_-msk* | • GLONASS SV mask for visible SVs; if the bit is set, the SV is available. |
| *sbas_-ephemeris_sv_-msk* | • SBAS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. |

| | |
|---|---|
| *sbas_almanac_sv_msk* | • SBAS SV mask for almanac; if the bit is set, almanac for that SV is available. |
| *sbas_health_sv_msk* | • SBAS SV mask for health; if the bit is set, health for that SV is available. |
| *sbas_visible_sv_msk* | • SBAS SV mask for visible SVs; if the bit is set, the SV is available. |
| *xtra_start_gps_week* | • Current XTRA information is valid starting from this GPS week number |
| *xtra_start_gps_minutes* | • Current XTRA information is valid starting from the GPS minutes with the GPS week |
| *xtra_valid_duration_hours* | • XTRA information is valid for this many hours starting from the specified GPS week/minutes |

## 8.264.2 Field Documentation

### 8.264.2.1 ULONG GPSStateInfo::Altitude

### 8.264.2.2 BYTE GPSStateInfo::EngineState

### 8.264.2.3 ULONG GPSStateInfo::glo_almanac_sv_msk

### 8.264.2.4 ULONG GPSStateInfo::glo_ephemeris_sv_msk

### 8.264.2.5 ULONG GPSStateInfo::glo_health_sv_msk

### 8.264.2.6 ULONG GPSStateInfo::glo_visible_sv_msk

### 8.264.2.7 ULONG GPSStateInfo::gps_almanac_sv_msk

### 8.264.2.8 ULONG GPSStateInfo::gps_ephemeris_sv_msk

### 8.264.2.9 ULONG GPSStateInfo::gps_health_sv_msk

### 8.264.2.10 ULONG GPSStateInfo::gps_visible_sv_msk

### 8.264.2.11 ULONG GPSStateInfo::HorizontalUncertainty

### 8.264.2.12 BYTE GPSStateInfo::Iono_valid

### 8.264.2.13 ULONGLONG GPSStateInfo::Latitude

### 8.264.2.14 ULONGLONG GPSStateInfo::Longitude

### 8.264.2.15 ULONG GPSStateInfo::sbas_almanac_sv_msk

### 8.264.2.16 ULONG GPSStateInfo::sbas_ephemeris_sv_msk

#### 8.264.2.17 ULONG GPSStateInfo::sbas_health_sv_msk

#### 8.264.2.18 ULONG GPSStateInfo::sbas_visible_sv_msk

#### 8.264.2.19 ULONG GPSStateInfo::Time_uncert_ms

#### 8.264.2.20 WORD GPSStateInfo::TimeStmp_gps_week

#### 8.264.2.21 ULONG GPSStateInfo::TimeStmp_tow_ms

#### 8.264.2.22 ULONG GPSStateInfo::ValidMask

#### 8.264.2.23 ULONG GPSStateInfo::VerticalUncertainty

#### 8.264.2.24 WORD GPSStateInfo::xtra_start_gps_minutes

#### 8.264.2.25 WORD GPSStateInfo::xtra_start_gps_week

#### 8.264.2.26 WORD GPSStateInfo::xtra_valid_duration_hours

## 8.265 gpsTime_s Struct Reference

### Data Fields

- WORD gpsWeek
- ULONG gpsTimeOfWeekMs

### 8.265.1 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 |

### 8.265.2 Field Documentation

#### 8.265.2.1 ULONG gpsTime_s::gpsTimeOfWeekMs

#### 8.265.2.2 WORD gpsTime_s::gpsWeek

## 8.266 gsmCellInfo Struct Reference

### Data Fields

- WORD arfcn
- BYTE band1900

- BYTE cellIdValid
- BYTE bsicId
- SHORT rssi
- SHORT srxlev

### 8.266.1 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. |

### 8.266.2 Field Documentation

#### 8.266.2.1 WORD gsmCellInfo::arfcn

#### 8.266.2.2 BYTE gsmCellInfo::band1900

#### 8.266.2.3 BYTE gsmCellInfo::bsicId

#### 8.266.2.4 BYTE gsmCellInfo::cellIdValid

#### 8.266.2.5 SHORT gsmCellInfo::rssi

#### 8.266.2.6 SHORT gsmCellInfo::srxlev

## 8.267 GSMRSSIThresh Struct Reference

### Data Fields

- BYTE GSMRSSIThreshListLen
- WORD * pGSMRSSIThreshList

### 8.267.1 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) |

### 8.267.2 Field Documentation

#### 8.267.2.1 BYTE GSMRSSIThresh::GSMRSSIThreshListLen

#### 8.267.2.2 WORD* GSMRSSIThresh::pGSMRSSIThreshList

## 8.268 GSMSrvStatusInfo Struct Reference

### Data Fields

- BYTE srvStatus
- BYTE trueSrvStatus
- BYTE isPrefDataPath

### 8.268.1 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 |

### 8.268.2 Field Documentation

#### 8.268.2.1 BYTE GSMSrvStatusInfo::isPrefDataPath

#### 8.268.2.2 BYTE GSMSrvStatusInfo::srvStatus

#### 8.268.2.3 BYTE GSMSrvStatusInfo::trueSrvStatus

## 8.269 GSMSysInfo Struct Reference

### Data Fields

- sysInfoCommon sysInfoGSM
- BYTE lacValid
- WORD lac
- BYTE cellIdValid
- ULONG cellId
- BYTE regRejectInfoValid
- BYTE rejectSrvDomain
- BYTE rejCause
- BYTE networkIdValid
- BYTE MCC [3]
- BYTE MNC [3]
- BYTE egprsSuppValid
- BYTE egprsSupp
- BYTE dtmSuppValid
- BYTE dtmSupp

### 8.269.1 Detailed Description

Structure for storing the GSM System Information.

**Parameters**

| | |
|---|---|
| *sysInfoGSM* | • See 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[PLMN_LENGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters |
| *MNC[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. |
| *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 |

## 8.269.2 Field Documentation

### 8.269.2.1 ULONG GSMSysInfo::cellId

### 8.269.2.2 BYTE GSMSysInfo::cellIdValid

### 8.269.2.3 BYTE GSMSysInfo::dtmSupp

### 8.269.2.4 BYTE GSMSysInfo::dtmSuppValid

### 8.269.2.5 BYTE GSMSysInfo::egprsSupp

### 8.269.2.6 BYTE GSMSysInfo::egprsSuppValid

#### 8.269.2.7 WORD GSMSysInfo::lac

#### 8.269.2.8 BYTE GSMSysInfo::lacValid

#### 8.269.2.9 BYTE GSMSysInfo::MCC[3]

#### 8.269.2.10 BYTE GSMSysInfo::MNC[3]

#### 8.269.2.11 BYTE GSMSysInfo::networkIdValid

#### 8.269.2.12 BYTE GSMSysInfo::regRejectInfoValid

#### 8.269.2.13 BYTE GSMSysInfo::rejCause

#### 8.269.2.14 BYTE GSMSysInfo::rejectSrvDomain

#### 8.269.2.15 sysInfoCommon GSMSysInfo::sysInfoGSM

## 8.270 gyroAcceptReady_s Struct Reference

### Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

### 8.270.1 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. |

### 8.270.2 Field Documentation

#### 8.270.2.1 WORD gyroAcceptReady_s::batchPerSec

#### 8.270.2.2 BYTE gyroAcceptReady_s::injectEnable

#### 8.270.2.3 WORD gyroAcceptReady_s::samplesPerBatch

## 8.271 gyroTempAcceptReady_s Struct Reference

### Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

### 8.271.1 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. |

### 8.271.2 Field Documentation

#### 8.271.2.1 WORD gyroTempAcceptReady_s::batchPerSec

#### 8.271.2.2 BYTE gyroTempAcceptReady_s::injectEnable

#### 8.271.2.3 WORD gyroTempAcceptReady_s::samplesPerBatch

## 8.272 HDRECIOThresh Struct Reference

### Data Fields

- BYTE HDRECIOThreshListLen
- WORD ∗ pHDRECIOThreshList

### 8.272.1 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). |

### 8.272.2 Field Documentation

#### 8.272.2.1 BYTE HDRECIOThresh::HDRECIOThreshListLen

#### 8.272.2.2 WORD* HDRECIOThresh::pHDRECIOThreshList

## 8.273 HDRIOThresh Struct Reference

### Data Fields

- BYTE HDRIOThreshListLen
- WORD * pHDRIOThreshList

### 8.273.1 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). |

### 8.273.2 Field Documentation

#### 8.273.2.1 BYTE HDRIOThresh::HDRIOThreshListLen

#### 8.273.2.2 WORD* HDRIOThresh::pHDRIOThreshList

## 8.274 HDRPersonalityInd Struct Reference

### Data Fields

- WORD * pCurrentPersonality
- BYTE * pPersonalityListLength
- protocolSubtypeElement * pProtocolSubtypeElement

### 8.274.1 Field Documentation

#### 8.274.1.1 WORD* HDRPersonalityInd::pCurrentPersonality

#### 8.274.1.2 BYTE* HDRPersonalityInd::pPersonalityListLength

#### 8.274.1.3 protocolSubtypeElement* HDRPersonalityInd::pProtocolSubtypeElement

## 8.275 HDRPersonalityResp Struct Reference

### Data Fields

- WORD * pCurrentPersonality
- BYTE * pPersonalityListLength
- protocolSubtypeElement * pProtocolSubtypeElement

### 8.275.1 Detailed Description

This structure contains information about the SLQSSwiGetHDRPersonality response parameters.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality[Out]* | • Current active personality index. |
| *pPersonalityList-Length[In/Out]* | • Number of Personality Protocol Subtype contains in this response.<br>• maximum input value is 3 |
| *pProtocol-Subtype-Element[Out]* | • See protocolSubtypeElement for more information. |

### 8.275.2 Field Documentation

#### 8.275.2.1 WORD* HDRPersonalityResp::pCurrentPersonality

#### 8.275.2.2 BYTE* HDRPersonalityResp::pPersonalityListLength

#### 8.275.2.3 protocolSubtypeElement* HDRPersonalityResp::pProtocolSubtypeElement

## 8.276 HDRProtSubtypResp Struct Reference

### Data Fields

- WORD * pCurrentPrsnlty
- BYTE * pPersonalityListLength

- protocolSubtypeElement * pProtoSubTypElmnt
- ULONGLONG * pAppSubType

### 8.276.1 Detailed Description

This structure contains information about the SLQSSwiGetHDRProtSubtype response parameters.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality[Out]* | • Current active personality index. |
| *pPersonalityList-Length[In/Out]* | • Number of Personality Protocol Subtype contains in this response.<br>• maximum input value is 4 |
| *pProtocol-Subtype-Element[Out]* | • See protocolSubtypeElement for more information. |
| *pAppSubType[-Out]* | • Stream application subtype<br>• Application subtype for each stream, |

### 8.276.2 Field Documentation

#### 8.276.2.1 ULONGLONG* HDRProtSubtypResp::pAppSubType

#### 8.276.2.2 WORD* HDRProtSubtypResp::pCurrentPrsnlty

#### 8.276.2.3 BYTE* HDRProtSubtypResp::pPersonalityListLength

#### 8.276.2.4 protocolSubtypeElement* HDRProtSubtypResp::pProtoSubTypElmnt

## 8.277 HDRRSSIThresh Struct Reference

### Data Fields

- BYTE HDRRSSIThreshListLen
- WORD * pHDRRSSIThreshList

### 8.277.1 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). |

### 8.277.2 Field Documentation

#### 8.277.2.1 BYTE HDRRSSIThresh::HDRRSSIThreshListLen

#### 8.277.2.2 WORD* HDRRSSIThresh::pHDRRSSIThreshList

## 8.278 HDRSINRThresh Struct Reference

### Data Fields

- BYTE HDRSINRThresListLen
- BYTE * pHDRSINRThresList

### 8.278.1 Detailed Description

This structure contains HDR SINR threshold related parameters.

**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 |

### 8.278.2 Field Documentation

#### 8.278.2.1 BYTE HDRSINRThresh::HDRSINRThresListLen

#### 8.278.2.2 BYTE* HDRSINRThresh::pHDRSINRThresList

## 8.279 HDRSINRThreshold Struct Reference

### Data Fields

- BYTE HDRSINRThreshListLen
- WORD * pHDRSINRThreshList

### 8.279.1 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 |

### 8.279.2 Field Documentation

#### 8.279.2.1 BYTE HDRSINRThreshold::HDRSINRThreshListLen

#### 8.279.2.2 WORD* HDRSINRThreshold::pHDRSINRThreshList

## 8.280 HDRSSInfo Struct Reference

### Data Fields

- INT8 rssi
- SHORT ecio
- BYTE sinr
- INT32 io

### 8.280.1 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. |

### 8.280.2 Field Documentation

#### 8.280.2.1 SHORT HDRSSInfo::ecio

#### 8.280.2.2 INT32 HDRSSInfo::io

#### 8.280.2.3 INT8 HDRSSInfo::rssi

#### 8.280.2.4 BYTE HDRSSInfo::sinr

## 8.281 hdrSSInfo Struct Reference

### Data Fields

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

### 8.281.1 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. |

### 8.281.2 Field Documentation

#### 8.281.2.1 int16_t hdrSSInfo::ecio

#### 8.281.2.2 int32_t hdrSSInfo::io

#### 8.281.2.3 int8_t hdrSSInfo::rssi

#### 8.281.2.4 uint8_t hdrSSInfo::sinr

## 8.282 HDRSysInfo Struct Reference

### Data Fields

- sysInfoCommon sysInfoHDR
- BYTE isSysPrlMatchValid
- BYTE isSysPrlMatch
- BYTE hdrPersonalityValid
- BYTE hdrPersonality
- BYTE hdrActiveProtValid
- BYTE hdrActiveProt
- BYTE is856SysIdValid
- BYTE is856SysId [16]

### 8.282.1 Detailed Description

Structure for storing the HDR System Information.

**Parameters**

| | |
|---|---|
| *sysInfoHDR* | • See 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[SLQS_SYSTEM_ID_SIZE]* | • IS-856 system ID.<br>• Only applicable for HDR. |

### 8.282.2 Field Documentation

#### 8.282.2.1 BYTE HDRSysInfo::hdrActiveProt

#### 8.282.2.2 BYTE HDRSysInfo::hdrActiveProtValid

#### 8.282.2.3 BYTE HDRSysInfo::hdrPersonality

#### 8.282.2.4 BYTE HDRSysInfo::hdrPersonalityValid

#### 8.282.2.5 BYTE HDRSysInfo::is856SysId[16]

#### 8.282.2.6 BYTE HDRSysInfo::is856SysIdValid

#### 8.282.2.7 BYTE HDRSysInfo::isSysPrlMatch

#### 8.282.2.8 BYTE HDRSysInfo::isSysPrlMatchValid

#### 8.282.2.9 sysInfoCommon HDRSysInfo::sysInfoHDR

## 8.283 homeSIDNID Struct Reference

### Data Fields

- BYTE numInstances
- sidNid SidNid [255]

### 8.283.1 Detailed Description

This structure contains the parameters for Home SID/NID Information

**Parameters**

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

### 8.283.2 Field Documentation

#### 8.283.2.1 BYTE homeSIDNID::numInstances

#### 8.283.2.2 sidNid homeSIDNID::SidNid[255]

## 8.284 hotSwapStatus Struct Reference

### Data Fields

- BYTE hotSwapLength
- BYTE hotSwap [255]

### 8.284.1 Detailed Description

This structure contains Hot Swap Status Information.

**Parameters**

| | |
|---|---|
| *hotSwapLength* | • Number of sets of the following elements. i.e. hot_swap |
| *hotSwap[MAX_-DESCRIPTION-_LENGTH]* | • 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 |

### 8.284.2 Field Documentation

#### 8.284.2.1 BYTE hotSwapStatus::hotSwap[255]

#### 8.284.2.2 BYTE hotSwapStatus::hotSwapLength

## 8.285 hrpdMTUSizeTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- WORD hrpdMTUSize

### 8.285.1 Detailed Description

This structure contains the parameters for HRPD MTU Size.

**Parameters**

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

| | |
|---|---|
| *hrpdMTUSize* | HRPD size<br>• Size of HRPD MTU |

### 8.285.2 Field Documentation

#### 8.285.2.1 WORD hrpdMTUSizeTlv::hrpdMTUSize

#### 8.285.2.2 BYTE hrpdMTUSizeTlv::TlvPresent

## 8.286 image_info_t Struct Reference

### Data Fields

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

### 8.286.1 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 |

### 8.286.2 Field Documentation

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

#### 8.286.2.2 uint8_t image_info_t::buildIDLen

#### 8.286.2.3 uint8_t image_info_t::imageType

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

## 8.287 ImageElement Struct Reference

### Data Fields

- BYTE imageType
- BYTE imageId [16]
- BYTE buildIdLength
- CHAR buildId [100]

### 8.287.1 Detailed Description

push current alignment to stack set alignment to 1 byte boundary 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) |
| *buildId* | • Build ID ANSI string( Max 100 characters ) |

### 8.287.2 Field Documentation

#### 8.287.2.1 CHAR ImageElement::buildId[100]

#### 8.287.2.2 BYTE ImageElement::buildIdLength

#### 8.287.2.3 BYTE ImageElement::imageId[16]

#### 8.287.2.4 BYTE ImageElement::imageType

## 8.288 ImageIdElement Struct Reference

### Data Fields

- BYTE storageIndex
- BYTE failureCount
- BYTE imageID [16]
- BYTE buildIDLength
- CHAR buildID [100]

### 8.288.1 Detailed Description

push current alignment to stack set alignment to 1 byte boundary 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 ) |

### 8.288.2 Field Documentation

#### 8.288.2.1 CHAR ImageIdElement::buildID[100]

#### 8.288.2.2 BYTE ImageIdElement::buildIDLength

#### 8.288.2.3 BYTE ImageIdElement::failureCount

#### 8.288.2.4 BYTE ImageIdElement::imageID[16]

#### 8.288.2.5 BYTE ImageIdElement::storageIndex

## 8.289 ImageIDEntries Struct Reference

### Data Fields

- BYTE imageType
- BYTE maxImages
- BYTE executingImage
- BYTE imageIDSize
- struct ImageIdElement imageIDElement [50]

### 8.289.1 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 ) |

### 8.289.2 Field Documentation

#### 8.289.2.1 BYTE ImageIDEntries::executingImage

#### 8.289.2.2 struct ImageIdElement ImageIDEntries::imageIDElement[50]

#### 8.289.2.3 BYTE ImageIDEntries::imageIDSize

#### 8.289.2.4 BYTE ImageIDEntries::imageType

#### 8.289.2.5 BYTE ImageIDEntries::maxImages

## 8.290 ImageList Struct Reference

### Data Fields

- BYTE listSize
- struct ImageIDEntries imageIDEntries [2]

### 8.290.1 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 ) |

### 8.290.2 Field Documentation

#### 8.290.2.1 struct ImageIDEntries ImageList::imageIDEntries[2]

#### 8.290.2.2 BYTE ImageList::listSize

## 8.291 IMSAIndRegisterInfo Struct Reference

### Data Fields

- BYTE * pRegStatusConfig
- BYTE * pServiceStatusConfig
- BYTE * pRatHandoverStatusConfig
- BYTE * pPdpStatusConfig

### 8.291.1 Detailed Description

This structure contains parameters of IMSA Config Indication Register

**Parameters**

| | |
|---|---|
| *pRegStatusConfig(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 |
| *pServiceStatusConfig(optional)* | • Register Indication For Service status Events.<br>• When this registration is enabled, the device learns of Service status via the QMI_IMSA_SERVICE_STATUS_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pRatHandoverStatusConfig(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 |
| *pPdpStatusConfig(optional)* | • PDP Status Configuration.<br>– 0x00 - Disable<br>– 0x01 - Enable |

**Note**

One of the optional parameter is mandatory to be present in the request.

### 8.291.2 Field Documentation

#### 8.291.2.1 BYTE* IMSAIndRegisterInfo::pPdpStatusConfig

#### 8.291.2.2 BYTE* IMSAIndRegisterInfo::pRatHandoverStatusConfig

#### 8.291.2.3 BYTE* IMSAIndRegisterInfo::pRegStatusConfig

#### 8.291.2.4 BYTE* IMSAIndRegisterInfo::pServiceStatusConfig

## 8.292 imsaPdpStatusInfo Struct Reference

### Data Fields

- BYTE connetionState
- ULONG * pFailErrorCode

### 8.292.1 Detailed Description

Contains the parameters passed for SLQSSetIMSAPdpStatusCallback by the device.

**Parameters**

| | |
|---|---|
| *connetionState* | • IMS PDP connection state information.<br>• Values<br>– TRUE – IMS PDP is connected<br>– FALSE – IMS PDP is not connected |
| *pFailErrorCode* | • IMS PDP connection failure error reason code when the IMS PDP Connection State TLV is FALSE.<br>• Values<br>– 0 - Generic failure reason for other than specified<br>– 1 - Option is unsubscribed.<br>– 2 - PDP status was unknown. |

### 8.292.2 Field Documentation

#### 8.292.2.1 BYTE imsaPdpStatusInfo::connetionState

#### 8.292.2.2 ULONG* imsaPdpStatusInfo::pFailErrorCode

## 8.293 imsaRatStatusInfo Struct Reference

### Data Fields

- ULONG * pRATStatus
- ULONG * pSrcRAT
- ULONG * pTgtRAT
- BYTE * pErrorCodeStr

### 8.293.1 Detailed Description

Contains the parameters passed for SLQSSetIMSARatStatusCallback by the device.

**Parameters**

| | |
|---|---|
| *pRATStatus* | • RAT handover Status |
| *pSrcRAT* | • Source RAT |
| *pTgtRAT* | • Target RAT |
| *pErrorCodeStr* | • Error Code String |

### 8.293.2 Field Documentation

#### 8.293.2.1 BYTE* imsaRatStatusInfo::pErrorCodeStr

#### 8.293.2.2 ULONG* imsaRatStatusInfo::pRATStatus

#### 8.293.2.3 ULONG* imsaRatStatusInfo::pSrcRAT

#### 8.293.2.4 ULONG* imsaRatStatusInfo::pTgtRAT

## 8.294 IMSARegistrationStatus Struct Reference

### Data Fields

- BYTE * pImsRegStatus
- WORD * pImsRegErrCode
- ULONG * pNewImsRegStatus

### 8.294.1 Detailed Description

This structure contains response parameters of registration status.

**Parameters**

| | |
|---|---|
| *pImsRegStatus* | • 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 |
| *pImsRegErr-Code* | • 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 |

| | |
|---|---|
| *pNewImsRegStatus* | • New IMS Registration Status<br>• Values<br>– 0 - Not registered for IMS<br>– 1 - Registering for IMS<br>– 2 - Registered for IMS |

### 8.294.2 Field Documentation

#### 8.294.2.1 WORD* IMSARegistrationStatus::pImsRegErrCode

#### 8.294.2.2 BYTE* IMSARegistrationStatus::pImsRegStatus

#### 8.294.2.3 ULONG* IMSARegistrationStatus::pNewImsRegStatus

## 8.295 imsaRegStatusInfo Struct Reference

### Data Fields

- BYTE * pbIMSRegistered
- WORD * pRegStatusErrorCode
- ULONG * pImsRegStatus

### 8.295.1 Detailed Description

Contains the parameters passed for SLQSSetIMSARegStatusCallback by the device.

**Parameters**

| | |
|---|---|
| *pbIMSRegistered* | • TRUE/FALSE |
| *pRegStatusErrorCode* | • if IMSA_STATUS_NOT_REGISTERED. Values: 3xx – Redirection responses 4xx – Client failure responses 5xx – Server failure responses 6xx – Global failure responses |
| *pImsRegStatus* | IMS registration status. Values: IMSA_STATUS_NOT_REGISTERED - 0 IMSA_STATUS_REGISTERING - 1 IMSA_STATUS_REGISTERED -2 |

### 8.295.2 Field Documentation

#### 8.295.2.1 BYTE* imsaRegStatusInfo::pbIMSRegistered

#### 8.295.2.2 ULONG* imsaRegStatusInfo::pImsRegStatus

#### 8.295.2.3 WORD* imsaRegStatusInfo::pRegStatusErrorCode

## 8.296 IMSAServiceStatus Struct Reference

### Data Fields

- ULONG * pSmsServiceStatus
- ULONG * pVoipServiceStatus
- ULONG * pVtServiceStatus
- ULONG * pSmsServiceRat
- ULONG * pVoipServiceRat
- ULONG * pVtServiceRat
- ULONG * pUtServiceStatus
- ULONG * pUtServiceRat
- ULONG * pVsServiceStatus
- ULONG * pVsServiceRat

### 8.296.1 Detailed Description

This structure contains response parameters of service status for various IMS services.

**Parameters**

| | |
|---|---|
| *pSmsService-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 |
| *pVoipSerive-Status* | • VoIP Service Status. -Values<br>– 0 - IMS VoIP service is not available<br>– 2 - IMS VoIP is in full service |
| *pVtService-Status* | • VT Service Status<br>• Values<br>– 0 - IMS VT service is not available<br>– 2 - IMS VT is in full service |
| *pSmsServiceRat* | • 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 |
| *pVoipServiceRat* | • 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 |

| | |
|---|---|
| *pVtServiceRat* | • 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 |
| *pUtService-Status* | • UT service Status.<br>• Values<br>– 0 - IMS UT service is not available<br>– 2 - IMS UT is in full service |
| *pUtServiceRat* | • 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 |
| *pVsService-Status* | • VS service Status.<br>• Values<br>– 0 - IMS UT service is not available<br>– 2 - IMS UT is in full service |
| *pVsServiceRat* | • 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 |

## 8.296.2 Field Documentation

### 8.296.2.1 ULONG* IMSAServiceStatus::pSmsServiceRat

### 8.296.2.2 ULONG* IMSAServiceStatus::pSmsServiceStatus

### 8.296.2.3 ULONG* IMSAServiceStatus::pUtServiceRat

### 8.296.2.4 ULONG* IMSAServiceStatus::pUtServiceStatus

### 8.296.2.5 ULONG* IMSAServiceStatus::pVoipServiceRat

### 8.296.2.6 ULONG* IMSAServiceStatus::pVoipServiceStatus

### 8.296.2.7 ULONG* IMSAServiceStatus::pVsServiceRat

### 8.296.2.8 ULONG* IMSAServiceStatus::pVsServiceStatus

#### 8.296.2.9 ULONG* IMSAServiceStatus::pVtServiceRat

#### 8.296.2.10 ULONG* IMSAServiceStatus::pVtServiceStatus

## 8.297 IMSASupportedFieldsResp Struct Reference

### Data Fields

- struct ReqFieldsList * pReqFieldsList
- struct RespFieldsList * pRespFieldsList
- struct IndFieldsList * pIndFieldsList

### 8.297.1 Detailed Description

This structure contains response of supported fields by the currently running software.

**Parameters**

| | |
|---|---|
| *pReqFieldsList* | • List of Supported Request Fields.<br>• See ReqFieldsList for more information |
| *pRespFieldsList* | • List of Supported Request Fields.<br>• See RespFieldsList for more information |
| *pIndFieldsList* | • List of Supported Request Fields.<br>• See IndFieldsList for more information |

### 8.297.2 Field Documentation

#### 8.297.2.1 struct IndFieldsList* IMSASupportedFieldsResp::pIndFieldsList

#### 8.297.2.2 struct ReqFieldsList* IMSASupportedFieldsResp::pReqFieldsList

#### 8.297.2.3 struct RespFieldsList* IMSASupportedFieldsResp::pRespFieldsList

## 8.298 IMSASupportedMsgInfo Struct Reference

### Data Fields

- struct SupportedMsgList * pSupportedMsgList

### 8.298.1 Detailed Description

This structure contains Queries the set of messages implemented by the currently running software.

**Parameters**

| | |
|---|---|
| *pSupportedMsg-List* | • List of Supported Messages.<br>• See SupportedMsgList for more information |

### 8.298.2 Field Documentation

#### 8.298.2.1 struct SupportedMsgList* IMSASupportedMsgInfo::pSupportedMsgList

## 8.299 imsaSvcStatusInfo Struct Reference

### Data Fields

- ULONG * pSMSSvcStatus
- ULONG * pVOIPSvcStatus
- ULONG * pVTSvcStatus
- ULONG * pSMSSvcRAT
- ULONG * pVOIPSvcRAT
- ULONG * pVTSvcRAT
- ULONG * pUTSvcStatus
- ULONG * pUTSvcRAT

### 8.299.1 Detailed Description

Contains the parameters passed for SLQSSetIMSASvcStatusCallback by the device.

**Parameters**

| | |
|---|---|
| *pSMSSvcStatus* | IMS SMS NOT AVAILABLE - 0 IMS SMS LIMITED SERVICE - 1 IMS SMS FULL SERVICE - 2 |
| *pVOIPSvcStatus* | IMS VOIP NOT AVAILABLE - 0 IMS SMS FULL SERVICE -2 |
| *pVTSvcStatus* | IMS VT NOT AVAILABLE - 0 IMS VT FULL SERVICE - 2 |
| *pSMSSvcRAT* | IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN -2 |
| *pVOIPSvcRAT* | IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN -2 * |
| *pVTSvcRAT* | IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN -2 |
| *pUTSvcStatus* | IMS VOIP NOT AVAILABLE - 0 IMS SMS FULL SERVICE -2 |
| *pUTSvcRAT* | IMS service is registered on WLAN - 0 IMS service is registered on WWAN - 1 IMS service is registered on interworking WLAN -2 |

### 8.299.2 Field Documentation

#### 8.299.2.1 ULONG* imsaSvcStatusInfo::pSMSSvcRAT

#### 8.299.2.2 ULONG* imsaSvcStatusInfo::pSMSSvcStatus

#### 8.299.2.3 ULONG* imsaSvcStatusInfo::pUTSvcRAT

#### 8.299.2.4 ULONG* imsaSvcStatusInfo::pUTSvcStatus

#### 8.299.2.5 ULONG* imsaSvcStatusInfo::pVOIPSvcRAT

#### 8.299.2.6 ULONG* imsaSvcStatusInfo::pVOIPSvcStatus

#### 8.299.2.7 ULONG* imsaSvcStatusInfo::pVTSvcRAT

#### 8.299.2.8 ULONG* imsaSvcStatusInfo::pVTSvcStatus

## 8.300 imsCfgIndRegisterInfo Struct Reference

### Data Fields

- BYTE * pSIPConfigEvents
- BYTE * pRegMgrConfigEvents
- BYTE * pSMSConfigEvents
- BYTE * pUserConfigEvents
- BYTE * pVoIPConfigEvents

### 8.300.1 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.

### 8.300.2 Field Documentation

#### 8.300.2.1 BYTE* imsCfgIndRegisterInfo::pRegMgrConfigEvents

#### 8.300.2.2 BYTE* imsCfgIndRegisterInfo::pSIPConfigEvents

#### 8.300.2.3 BYTE* imsCfgIndRegisterInfo::pSMSConfigEvents

#### 8.300.2.4 BYTE* imsCfgIndRegisterInfo::pUserConfigEvents

#### 8.300.2.5 BYTE* imsCfgIndRegisterInfo::pVoIPConfigEvents

## 8.301 imsRegMgrConfigInfo Struct Reference

### Data Fields

- WORD * pPriCSCFPort
- BYTE * pCSCFPortName
- BYTE * pIMSTestMode

### 8.301.1 Detailed Description

Contains the parameters passed for SLQSSetRegMgrConfigCallback by the device.

**Parameters**

| | |
|---|---|
| *pPriCSCFPort* | • Primary call session control function port |
| *pCSCFPort-Name* | • Call Session control port, fully qualified domain name<br>• Length of this string can be of maximum 255 bytes |
| *pIMSTestMode* | • IMS Test mode Enabled.<br>– TRUE - Enable, no IMS registration<br>– FALSE - Disable, IMS registration is initiated |

### 8.301.2 Field Documentation

#### 8.301.2.1 BYTE* imsRegMgrConfigInfo::pCSCFPortName

#### 8.301.2.2 BYTE* imsRegMgrConfigInfo::pIMSTestMode

#### 8.301.2.3 WORD* imsRegMgrConfigInfo::pPriCSCFPort

## 8.302 imsSIPConfigInfo Struct Reference

### Data Fields

- WORD * pSIPLocalPort
- ULONG * pTimerSIPReg
- ULONG * pSubscribeTimer
- ULONG * pTimerT1
- ULONG * pTimerT2
- ULONG * pTimerTf
- BYTE * pSigCompEnabled

### 8.302.1 Detailed Description

Contains the parameters passed for SLQSSetSIPConfigCallback by the device.

**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 |

**Note**

None

### 8.302.2 Field Documentation

#### 8.302.2.1 BYTE* imsSIPConfigInfo::pSigCompEnabled

#### 8.302.2.2 WORD* imsSIPConfigInfo::pSIPLocalPort

#### 8.302.2.3 ULONG* imsSIPConfigInfo::pSubscribeTimer

#### 8.302.2.4 ULONG* imsSIPConfigInfo::pTimerSIPReg

#### 8.302.2.5 ULONG* imsSIPConfigInfo::pTimerT1

#### 8.302.2.6 ULONG* imsSIPConfigInfo::pTimerT2

#### 8.302.2.7 ULONG* imsSIPConfigInfo::pTimerTf

## 8.303 imsSMSConfigInfo Struct Reference

### Data Fields

- BYTE * pSMSFormat
- BYTE * pSMSOverIPNwInd
- BYTE * pPhoneCtxtURI

### 8.303.1 Detailed Description

Contains the parameters passed for SLQSSetIMSSMSConfigCallback by the device.

**Parameters**

| | |
|---|---|
| *pSMSFormat* | • SMS format<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *pSMSOverIPNw-Ind* | • SMS over IP Network Indication Flag<br>– TRUE - Mobile-Originated(MO) SMS turned on<br>– FALSE - MO SMS turned off |
| *pPhoneCtxtURI* | • Phone context universal resource identifier<br>• Length of this string can be of maximum 255 bytes |

### 8.303.2 Field Documentation

#### 8.303.2.1 BYTE* imsSMSConfigInfo::pPhoneCtxtURI

#### 8.303.2.2 BYTE* imsSMSConfigInfo::pSMSFormat

#### 8.303.2.3 BYTE* imsSMSConfigInfo::pSMSOverIPNwInd

## 8.304 imsUserConfigInfo Struct Reference

### Data Fields

- BYTE * pIMSDomain

### 8.304.1 Detailed Description

Contains the parameters passed for SLQSSetIMSUserConfigCallback by the device.

**Parameters**

| | |
|---|---|
| *pIMSDomain* | • IMS domain name<br>• Length of this string can be of maximum 255 bytes |

### 8.304.2 Field Documentation

#### 8.304.2.1 BYTE* imsUserConfigInfo::pIMSDomain

## 8.305 imsVoIPConfigInfo Struct Reference

### Data Fields

- WORD * pSessionExpiryTimer
- WORD * pMinSessionExpiryTimer
- BYTE * pAmrWbEnable
- BYTE * pScrAmrEnable
- BYTE * pScrAmrWbEnable
- BYTE * pAmrMode
- WORD * pAmrWBMode
- BYTE * pAmrOctetAligned
- BYTE * pAmrWBOctetAligned
- WORD * pRingingTimer
- WORD * pRingBackTimer
- WORD * pRTPRTCPInactTimer

### 8.305.1 Detailed Description

Contains the parameters passed for SLQSSetIMSVoIPConfigCallback by the device.

**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 - Enabled<br>– False - Disabled |
| *pScrAmrEnable* | • Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB)<br>• Values:<br>– True - Enabled<br>– False - Disabled |
| *pScrAmrWb-Enable* | • Flag to enable/disable SCR for AMR WB Audio<br>• Values:<br>– True - Enabled<br>– False - Disabled |
| *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. |

### 8.305.2 Field Documentation

#### 8.305.2.1 BYTE* imsVoIPConfigInfo::pAmrMode

#### 8.305.2.2 BYTE* imsVoIPConfigInfo::pAmrOctetAligned

#### 8.305.2.3 BYTE* imsVoIPConfigInfo::pAmrWbEnable

#### 8.305.2.4 WORD* imsVoIPConfigInfo::pAmrWBMode

#### 8.305.2.5 BYTE* imsVoIPConfigInfo::pAmrWBOctetAligned

#### 8.305.2.6 WORD* imsVoIPConfigInfo::pMinSessionExpiryTimer

#### 8.305.2.7 WORD* imsVoIPConfigInfo::pRingBackTimer

#### 8.305.2.8 WORD* imsVoIPConfigInfo::pRingingTimer

#### 8.305.2.9 WORD* imsVoIPConfigInfo::pRTPRTCPInactTimer

#### 8.305.2.10 BYTE* imsVoIPConfigInfo::pScrAmrEnable

#### 8.305.2.11 BYTE* imsVoIPConfigInfo::pScrAmrWbEnable

#### 8.305.2.12 WORD* imsVoIPConfigInfo::pSessionExpiryTimer

## 8.306 IndFieldsList Struct Reference

### Data Fields

- BYTE indicationFieldsLen
- BYTE indicationFields [256]

### 8.306.1 Detailed Description

This structure contains the Supported Indication Fields List Information

**Parameters**

| | |
|---|---|
| *indicationFields-Len* | • Number of sets of the indication fields. |
| *indicationFields* | • Describes which optional field IDs are supported in QMI indication.<br>• Format is same as request field. |

### 8.306.2 Field Documentation

#### 8.306.2.1 BYTE IndFieldsList::indicationFields[256]

#### 8.306.2.2 BYTE IndFieldsList::indicationFieldsLen

## 8.307 infoInterFreq Struct Reference

### Data Fields

- WORD earfcn
- BYTE threshXLow
- BYTE threshXHigh
- BYTE cell_resel_priority
- BYTE cells_len
- cellParams cellInterFreqParams [255]

### 8.307.1 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[MAX_DESCRIPTION_LENGTH]* | • See cellParams for more information. |

### 8.307.2 Field Documentation

#### 8.307.2.1 BYTE infoInterFreq::cell_resel_priority

#### 8.307.2.2 cellParams infoInterFreq::cellInterFreqParams[255]

#### 8.307.2.3 BYTE infoInterFreq::cells_len

#### 8.307.2.4 WORD infoInterFreq::earfcn

#### 8.307.2.5 BYTE infoInterFreq::threshXHigh

#### 8.307.2.6 BYTE infoInterFreq::threshXLow

## 8.308 IOThresh Struct Reference

### Data Fields

- BYTE IOThresListLen
- INT32 * pIOThresList

### 8.308.1 Detailed Description

This structure contains IO threshold related parameters.

**Parameters**

| | |
|---|---|
| *IOThresListLen* | • Length of the LTE SNR 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 |

### 8.308.2 Field Documentation

#### 8.308.2.1 BYTE IOThresh::IOThresListLen

#### 8.308.2.2 INT32* IOThresh::pIOThresList

## 8.309 IPv4Addr Struct Reference

### Data Fields

- ULONG addr
- ULONG subnetMask

### 8.309.1 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 |

### 8.309.2 Field Documentation

#### 8.309.2.1 ULONG IPv4Addr::addr

#### 8.309.2.2 ULONG IPv4Addr::subnetMask

## 8.310 IPv4Config Struct Reference

### Data Fields

- ULONG IPv4Addr
- WORD IPv4Port

### 8.310.1 Detailed Description

This structure specifies information regarding the IPv4 address and port.

**Parameters**

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

### 8.310.2 Field Documentation

#### 8.310.2.1 ULONG IPv4Config::IPv4Addr

#### 8.310.2.2 WORD IPv4Config::IPv4Port

## 8.311 IPv4Info Struct Reference

### Data Fields

- ULONG address
- WORD port
- BYTE TlvPresent

### 8.311.1 Detailed Description

This structure contains LOC Server IPV4 info field.

**Parameters**

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

### 8.311.2 Field Documentation

#### 8.311.2.1 ULONG IPv4Info::address

#### 8.311.2.2 WORD IPv4Info::port

#### 8.311.2.3 BYTE IPv4Info::TlvPresent

## 8.312 IPv6Addr Struct Reference

### Data Fields

- BYTE addr [16]
- BYTE prefixLen

### 8.312.1 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. |

### 8.312.2 Field Documentation

#### 8.312.2.1 BYTE IPv6Addr::addr[16]

#### 8.312.2.2 BYTE IPv6Addr::prefixLen

## 8.313 IPV6AddressInfo Struct Reference

### Data Fields

- BYTE IPV6PrefixLen
- USHORT IPAddressV6 [8]

### 8.313.1 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. |

### 8.313.2 Field Documentation

#### 8.313.2.1 USHORT IPV6AddressInfo::IPAddressV6[8]

#### 8.313.2.2 BYTE IPV6AddressInfo::IPV6PrefixLen

## 8.314 ipv6AddressInfo Struct Reference

### Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

### 8.314.1 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) |

### 8.314.2 Field Documentation

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

#### 8.314.2.2 uint8_t ipv6AddressInfo::IPV6PrefixLen

## 8.315 IPv6Config Struct Reference

### Data Fields

- WORD IPv6Addr [8]
- ULONG IPv6Port

### 8.315.1 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. |

### 8.315.2 Field Documentation

#### 8.315.2.1 WORD IPv6Config::IPv6Addr[8]

#### 8.315.2.2 ULONG IPv6Config::IPv6Port

## 8.316 IPV6GWAddressInfo Struct Reference

### Data Fields

- BYTE gwV6PrefixLen
- USHORT gwAddressV6 [8]

### 8.316.1 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. |

### 8.316.2 Field Documentation

#### 8.316.2.1 USHORT IPV6GWAddressInfo::gwAddressV6[8]

#### 8.316.2.2 BYTE IPV6GWAddressInfo::gwV6PrefixLen

## 8.317 IPv6Info Struct Reference

### Data Fields

- WORD address [8]
- ULONG port
- BYTE TlvPresent

### 8.317.1 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 |

### 8.317.2 Field Documentation

#### 8.317.2.1 WORD IPv6Info::address[8]

#### 8.317.2.2 ULONG IPv6Info::port

#### 8.317.2.3 BYTE IPv6Info::TlvPresent

## 8.318 IPv6TrafCls Struct Reference

### Data Fields

- BYTE val
- BYTE mask

#### 8.318.1 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 |

#### 8.318.2 Field Documentation

##### 8.318.2.1 BYTE IPv6TrafCls::mask

##### 8.318.2.2 BYTE IPv6TrafCls::val

## 8.319 LibPackGPRSRequestedQoS Struct Reference

### Data Fields

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

#### 8.319.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

**Parameters**

| | |
|---|---|
| *precedenceClass* | • Precedence class |
| *delayClass* | • Delay class |
| *reliabilityClass* | • Reliability class |
| *peakThroughputClass* | • Peak throughput class |

| | |
|---|---|
| *meanThroughputClass* | • Mean throughput class |

### 8.319.2 Field Documentation

#### 8.319.2.1 uint32_t LibPackGPRSRequestedQoS::delayClass

#### 8.319.2.2 uint32_t LibPackGPRSRequestedQoS::meanThroughputClass

#### 8.319.2.3 uint32_t LibPackGPRSRequestedQoS::peakThroughputClass

#### 8.319.2.4 uint32_t LibPackGPRSRequestedQoS::precedenceClass

#### 8.319.2.5 uint32_t LibPackGPRSRequestedQoS::reliabilityClass

## 8.320 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

### 8.320.1 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. |

### 8.320.2 Field Documentation

#### 8.320.2.1 uint8_t* LibpackProfile3GPP::pAddrAllocPref

#### 8.320.2.2 uint8_t* LibpackProfile3GPP::pAPNClass

#### 8.320.2.3 uint8_t* LibpackProfile3GPP::pAPNDisabledFlag

#### 8.320.2.4 uint8_t* LibpackProfile3GPP::pAPNName

#### 8.320.2.5 uint16_t* LibpackProfile3GPP::pAPNnameSize

#### 8.320.2.6 uint8_t* LibpackProfile3GPP::pAuthenticationPref

#### 8.320.2.7 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSMinimumQoS

8.320.2.8 **LibPackGPRSRequestedQoS*** LibpackProfile3GPP::pGPRSRequestedQos

8.320.2.9 uint8_t* LibpackProfile3GPP::pImCnFlag

8.320.2.10 uint32_t* LibpackProfile3GPP::pIPv4AddrPref

8.320.2.11 uint16_t* LibpackProfile3GPP::pIPv6AddPref

8.320.2.12 uint8_t* LibpackProfile3GPP::pPassword

8.320.2.13 uint16_t* LibpackProfile3GPP::pPasswordSize

8.320.2.14 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingDhcp

8.320.2.15 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingPCO

8.320.2.16 uint32_t* LibpackProfile3GPP::pPDNInactivTimeout

8.320.2.17 uint8_t* LibpackProfile3GPP::pPdpAccessConFlag

8.320.2.18 uint8_t* LibpackProfile3GPP::pPdpContext

8.320.2.19 uint8_t* LibpackProfile3GPP::pPdpDataCompType

8.320.2.20 uint8_t* LibpackProfile3GPP::pPdpHdrCompType

8.320.2.21 uint8_t* LibpackProfile3GPP::pPDPtype

8.320.2.22 uint32_t* LibpackProfile3GPP::pPriDNSIPv4AddPref

8.320.2.23 uint16_t* LibpackProfile3GPP::pPriDNSIPv6addpref

8.320.2.24 uint8_t* LibpackProfile3GPP::pPrimaryID

8.320.2.25 uint8_t* LibpackProfile3GPP::pProfilename

8.320.2.26 uint16_t* LibpackProfile3GPP::pProfilenameSize

8.320.2.27 **LibPackQosClassID*** LibpackProfile3GPP::pQosClassID

8.320.2.28 uint32_t* LibpackProfile3GPP::pSecDNSIPv4AddPref

8.320.2.29 uint16_t* LibpackProfile3GPP::pSecDNSIPv6addpref

8.320.2.30 uint8_t* LibpackProfile3GPP::pSecondaryFlag

8.320.2.31 uint8_t* LibpackProfile3GPP::pSupportEmergencyCalls

8.320.2.32 **LibPackTFTIDParams*** LibpackProfile3GPP::pTFTID1Params

8.320.2.33 **LibPackTFTIDParams*** LibpackProfile3GPP::pTFTID2Params

8.320.2.34 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSMinQoS

8.320.2.35 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSMinQosSigInd

#### 8.320.2.36 LibPackUMTSQoS∗ LibpackProfile3GPP::pUMTSReqQoS

#### 8.320.2.37 LibPackUMTSReqQoSSigInd∗ LibpackProfile3GPP::pUMTSReqQoSSigInd

#### 8.320.2.38 uint8_t∗ LibpackProfile3GPP::pUsername

#### 8.320.2.39 uint16_t∗ LibpackProfile3GPP::pUsernameSize

## 8.321 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

### 8.321.1 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(). |

| | |
|---|---|
| *pPDNInactiv-Timeout3GPP2* | • 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(). |
| *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(). |

### 8.321.2 Field Documentation

#### 8.321.2.1 uint8_t* LibpackProfile3GPP2::pAllowLinger

#### 8.321.2.2 uint8_t* LibpackProfile3GPP2::pAPNClass3GPP2

#### 8.321.2.3 uint8_t* LibpackProfile3GPP2::pAPNEnabled3GPP2

#### 8.321.2.4 uint8_t* LibpackProfile3GPP2::pApnString

#### 8.321.2.5 uint16_t* LibpackProfile3GPP2::pApnStringSize

#### 8.321.2.6 uint8_t* LibpackProfile3GPP2::pAppPriority

#### 8.321.2.7 uint32_t* LibpackProfile3GPP2::pAppType

#### 8.321.2.8 uint8_t* LibpackProfile3GPP2::pAuthPassword

#### 8.321.2.9 uint16_t* LibpackProfile3GPP2::pAuthPasswordSize

#### 8.321.2.10 uint8_t* LibpackProfile3GPP2::pAuthProtocol

#### 8.321.2.11 uint8_t* LibpackProfile3GPP2::pAuthRetryCount

#### 8.321.2.12 uint16_t* LibpackProfile3GPP2::pAuthTimeout

#### 8.321.2.13 uint8_t* LibpackProfile3GPP2::pDataMode

#### 8.321.2.14 uint8_t* LibpackProfile3GPP2::pDataRate

#### 8.321.2.15 uint16_t* LibpackProfile3GPP2::pIpcpAckTimeout

#### 8.321.2.16 uint8_t* LibpackProfile3GPP2::pIpcpCreqRetryCount

#### 8.321.2.17 uint8_t* LibpackProfile3GPP2::pIsPcscfAddressNedded

#### 8.321.2.18 uint16_t* LibpackProfile3GPP2::pLcpAckTimeout

8.321.2.19 uint8_t* LibpackProfile3GPP2::pLcpCreqRetryCount

8.321.2.20 uint8_t* LibpackProfile3GPP2::pNegoDnsSrvrPref

8.321.2.21 uint32_t* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2

8.321.2.22 uint8_t* LibpackProfile3GPP2::pPdnType

8.321.2.23 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimer1x

8.321.2.24 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimerDO

8.321.2.25 uint32_t* LibpackProfile3GPP2::pPrimaryV4DnsAddress

8.321.2.26 uint16_t* LibpackProfile3GPP2::pPriV6DnsAddress

8.321.2.27 uint8_t* LibpackProfile3GPP2::pRATType

8.321.2.28 uint32_t* LibpackProfile3GPP2::pSecondaryV4DnsAddress

8.321.2.29 uint16_t* LibpackProfile3GPP2::pSecV6DnsAddress

8.321.2.30 uint8_t* LibpackProfile3GPP2::pUserId

8.321.2.31 uint16_t* LibpackProfile3GPP2::pUserIdSize

## 8.322 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

### 8.322.1 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 |
| *pProfilename-Size;* | • 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 |
| *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 |
| *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 |

| | |
|---|---|
| *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 |

### 8.322.2 Field Documentation

8.322.2.1 uint8_t* LibPackprofile_3GPP::pAddrAllocPref

8.322.2.2 uint8_t* LibPackprofile_3GPP::pAPNClass

8.322.2.3 uint8_t* LibPackprofile_3GPP::pAPNDisabledFlag

8.322.2.4 uint8_t* LibPackprofile_3GPP::pAPNName

8.322.2.5 uint16_t* LibPackprofile_3GPP::pAPNnameSize

8.322.2.6 uint8_t* LibPackprofile_3GPP::pAuthenticationPref

8.322.2.7 **LibPackGPRSRequestedQoS*** LibPackprofile_3GPP::pGPRSMinimumQoS

8.322.2.8 **LibPackGPRSRequestedQoS*** LibPackprofile_3GPP::pGPRSRequestedQos

8.322.2.9 uint8_t* LibPackprofile_3GPP::pImCnFlag

8.322.2.10 uint32_t* LibPackprofile_3GPP::pIPv4AddrPref

8.322.2.11 uint16_t* LibPackprofile_3GPP::pIPv6AddPref

8.322.2.12 uint8_t* LibPackprofile_3GPP::pPassword

8.322.2.13 uint16_t* LibPackprofile_3GPP::pPasswordSize

8.322.2.14 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingDhcp

8.322.2.15 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingPCO

8.322.2.16 uint32_t* LibPackprofile_3GPP::pPDNInactivTimeout

8.322.2.17 uint8_t* LibPackprofile_3GPP::pPdpAccessConFlag

8.322.2.18 uint8_t* LibPackprofile_3GPP::pPdpContext

8.322.2.19 uint8_t* LibPackprofile_3GPP::pPdpDataCompType

8.322.2.20 uint8_t* LibPackprofile_3GPP::pPdpHdrCompType

8.322.2.21 uint8_t* LibPackprofile_3GPP::pPDPtype

8.322.2.22 uint32_t* LibPackprofile_3GPP::pPriDNSIPv4AddPref

8.322.2.23 uint16_t* LibPackprofile_3GPP::pPriDNSIPv6addpref

8.322.2.24 uint8_t* LibPackprofile_3GPP::pPrimaryID

8.322.2.25 uint8_t* LibPackprofile_3GPP::pProfilename

8.322.2.26 uint16_t* LibPackprofile_3GPP::pProfilenameSize

8.322.2.27 **LibPackQosClassID*** LibPackprofile_3GPP::pQosClassID

8.322.2.28 uint32_t* LibPackprofile_3GPP::pSecDNSIPv4AddPref

8.322.2.29 uint16_t* LibPackprofile_3GPP::pSecDNSIPv6addpref

8.322.2.30 uint8_t* LibPackprofile_3GPP::pSecondaryFlag

8.322.2.31 uint8_t* LibPackprofile_3GPP::pSupportEmergencyCalls

8.322.2.32 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID1Params

8.322.2.33 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID2Params

8.322.2.34 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSMinQoS

8.322.2.35 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSMinQosSigInd

8.322.2.36 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSReqQoS

8.322.2.37 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSReqQoSSigInd

8.322.2.38 uint8_t* LibPackprofile_3GPP::pUsername

8.322.2.39 uint16_t* LibPackprofile_3GPP::pUsernameSize

## 8.323 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

### 8.323.1 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(). |

### 8.323.2 Field Documentation

#### 8.323.2.1 uint8_t* LibPackprofile_3GPP2::pAllowLinger

#### 8.323.2.2 uint8_t* LibPackprofile_3GPP2::pAPNClass3GPP2

#### 8.323.2.3 uint8_t* LibPackprofile_3GPP2::pAPNEnabled3GPP2

#### 8.323.2.4 uint8_t* LibPackprofile_3GPP2::pApnString

#### 8.323.2.5 uint16_t* LibPackprofile_3GPP2::pApnStringSize

#### 8.323.2.6 uint8_t* LibPackprofile_3GPP2::pAppPriority

#### 8.323.2.7 uint32_t* LibPackprofile_3GPP2::pAppType

#### 8.323.2.8 uint8_t* LibPackprofile_3GPP2::pAuthPassword

#### 8.323.2.9 uint16_t* LibPackprofile_3GPP2::pAuthPassword_tSize

#### 8.323.2.10 uint8_t* LibPackprofile_3GPP2::pAuthProtocol

#### 8.323.2.11 uint8_t* LibPackprofile_3GPP2::pAuthRetryCount

#### 8.323.2.12 uint16_t* LibPackprofile_3GPP2::pAuthTimeout

#### 8.323.2.13 uint8_t* LibPackprofile_3GPP2::pDataMode

#### 8.323.2.14 uint8_t* LibPackprofile_3GPP2::pDataRate

#### 8.323.2.15 uint16_t* LibPackprofile_3GPP2::pIpcpAckTimeout

#### 8.323.2.16 uint8_t* LibPackprofile_3GPP2::pIpcpCreqRetryCount

#### 8.323.2.17 uint8_t* LibPackprofile_3GPP2::pIsPcscfAddressNedded

#### 8.323.2.18 uint16_t* LibPackprofile_3GPP2::pLcpAckTimeout

#### 8.323.2.19 uint8_t* LibPackprofile_3GPP2::pLcpCreqRetryCount

#### 8.323.2.20 uint8_t* LibPackprofile_3GPP2::pNegoDnsSrvrPref

#### 8.323.2.21 uint32_t* LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2

#### 8.323.2.22 uint8_t* LibPackprofile_3GPP2::pPdnType

#### 8.323.2.23 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimer1x

#### 8.323.2.24 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimerDO

#### 8.323.2.25 uint32_t* LibPackprofile_3GPP2::pPrimaryV4DnsAddress

#### 8.323.2.26 uint16_t* LibPackprofile_3GPP2::pPriV6DnsAddress

#### 8.323.2.27 uint8_t* LibPackprofile_3GPP2::pRATType

#### 8.323.2.28 uint32_t* LibPackprofile_3GPP2::pSecondaryV4DnsAddress

#### 8.323.2.29 uint16_t* LibPackprofile_3GPP2::pSecV6DnsAddress

#### 8.323.2.30 uint8_t* LibPackprofile_3GPP2::pUserId

#### 8.323.2.31 uint16_t* LibPackprofile_3GPP2::pUserIdSize

## 8.324 LibPackQosClassID Struct Reference

### Data Fields

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

### 8.324.1 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 |

### 8.324.2 Field Documentation

#### 8.324.2.1 uint8_t LibPackQosClassID::gDlBitRate

#### 8.324.2.2 uint32_t LibPackQosClassID::gUlBitRate

#### 8.324.2.3 uint32_t LibPackQosClassID::maxDlBitRate

#### 8.324.2.4 uint32_t LibPackQosClassID::maxUlBitRate

#### 8.324.2.5 uint8_t LibPackQosClassID::QCI

## 8.325 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

### 8.325.1 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 |

## 8.325.2 Field Documentation

### 8.325.2.1 uint16_t LibPackTFTIDParams::destPortRangeEnd

### 8.325.2.2 uint32_t LibPackTFTIDParams::destPortRangeStart

#### 8.325.2.3 uint8_t LibPackTFTIDParams::eValid

#### 8.325.2.4 uint8_t LibPackTFTIDParams::filterId

#### 8.325.2.5 uint32_t LibPackTFTIDParams::flowLabel

#### 8.325.2.6 uint32_t LibPackTFTIDParams::IPSECSPI

#### 8.325.2.7 uint8_t LibPackTFTIDParams::ipVersion

#### 8.325.2.8 uint8_t LibPackTFTIDParams::nextHeader

#### 8.325.2.9 uint16_t* LibPackTFTIDParams::pSourceIP

#### 8.325.2.10 uint8_t LibPackTFTIDParams::sourceIPMask

#### 8.325.2.11 uint16_t LibPackTFTIDParams::srcPortRangeEnd

#### 8.325.2.12 uint16_t LibPackTFTIDParams::srcPortRangeStart

#### 8.325.2.13 uint16_t LibPackTFTIDParams::tosMask

## 8.326 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

### 8.326.1 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.

### 8.326.2 Field Documentation

#### 8.326.2.1 uint8_t LibPackUMTSQoS::deliveryErrSDU

#### 8.326.2.2 uint32_t LibPackUMTSQoS::grntDownlinkBitrate

#### 8.326.2.3 uint32_t LibPackUMTSQoS::grntUplinkBitrate

#### 8.326.2.4 uint32_t LibPackUMTSQoS::maxDownlinkBitrate

#### 8.326.2.5 uint32_t LibPackUMTSQoS::maxSDUSize

#### 8.326.2.6 uint32_t LibPackUMTSQoS::maxUplinkBitrate

#### 8.326.2.7 uint8_t LibPackUMTSQoS::qosDeliveryOrder

#### 8.326.2.8 uint8_t LibPackUMTSQoS::resBerRatio

#### 8.326.2.9 uint8_t LibPackUMTSQoS::sduErrorRatio

#### 8.326.2.10 uint8_t LibPackUMTSQoS::trafficClass

#### 8.326.2.11 uint32_t LibPackUMTSQoS::trafficPriority

#### 8.326.2.12 uint32_t LibPackUMTSQoS::transferDelay

## 8.327 LibPackUMTSReqQoSSigInd Struct Reference

### Data Fields

- LibPackUMTSQoS UMTSReqQoS
- uint8_t SigInd

### 8.327.1 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 |

### 8.327.2 Field Documentation

#### 8.327.2.1 uint8_t LibPackUMTSReqQoSSigInd::SigInd

#### 8.327.2.2 LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS

## 8.328 lineCtrlInfo Struct Reference

### Data Fields

- BYTE polarityIncluded
- BYTE toggleMode
- BYTE revPolarity
- BYTE pwrDenialTime

### 8.328.1 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 |

### 8.328.2 Field Documentation

#### 8.328.2.1 BYTE lineCtrlInfo::polarityIncluded

#### 8.328.2.2 BYTE lineCtrlInfo::pwrDenialTime

#### 8.328.2.3 BYTE lineCtrlInfo::revPolarity

#### 8.328.2.4 BYTE lineCtrlInfo::toggleMode

## 8.329 loc_accelAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.329.1 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. |

### 8.329.2 Field Documentation

#### 8.329.2.1 uint16_t loc_accelAcceptReady::batchPerSec

#### 8.329.2.2 uint8_t loc_accelAcceptReady::injectEnable

#### 8.329.2.3 uint16_t loc_accelAcceptReady::samplesPerBatch

## 8.330 loc_accelTempAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.330.1 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. |

### 8.330.2 Field Documentation

#### 8.330.2.1 uint16_t loc_accelTempAcceptReady::batchPerSec

#### 8.330.2.2 uint8_t loc_accelTempAcceptReady::injectEnable

#### 8.330.2.3 uint16_t loc_accelTempAcceptReady::samplesPerBatch

## 8.331 loc_BdsSV Struct Reference

### Data Fields

- uint16_t id
- uint8_t mask

### 8.331.1 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 |

### 8.331.2 Field Documentation

#### 8.331.2.1 uint16_t loc_BdsSV::id

#### 8.331.2.2 uint8_t loc_BdsSV::mask

## 8.332 loc_BdsSVInfo Struct Reference

### Data Fields

- uint8_t len
- loc_BdsSV * pSV

### 8.332.1 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 |

### 8.332.2 Field Documentation

#### 8.332.2.1 uint8_t loc_BdsSVInfo::len

#### 8.332.2.2 loc_BdsSV* loc_BdsSVInfo::pSV

## 8.333 loc_CellDb Struct Reference

### Data Fields

- uint32_t mask

#### 8.333.1 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 |

#### 8.333.2 Field Documentation

##### 8.333.2.1 uint32_t loc_CellDb::mask

## 8.334 loc_ClkInfo Struct Reference

### Data Fields

- uint32_t mask

#### 8.334.1 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 |

### 8.334.2 Field Documentation

#### 8.334.2.1 uint32_t loc_ClkInfo::mask

## 8.335 loc_GnssData Struct Reference

### Data Fields

- uint64_t mask

### 8.335.1 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 |

### 8.335.2 Field Documentation

#### 8.335.2.1 uint64_t loc_GnssData::mask

## 8.336 loc_gpsTime Struct Reference

### Data Fields

- uint16_t gpsWeek
- uint32_t gpsTimeOfWeekMs

### 8.336.1 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 |

### 8.336.2 Field Documentation

#### 8.336.2.1 uint32_t loc_gpsTime::gpsTimeOfWeekMs

#### 8.336.2.2 uint16_t loc_gpsTime::gpsWeek

## 8.337 loc_gyroAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.337.1 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. |

### 8.337.2 Field Documentation

#### 8.337.2.1 uint16_t loc_gyroAcceptReady::batchPerSec

#### 8.337.2.2 uint8_t loc_gyroAcceptReady::injectEnable

#### 8.337.2.3 uint16_t loc_gyroAcceptReady::samplesPerBatch

## 8.338 loc_gyroTempAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.338.1 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. |

### 8.338.2 Field Documentation

#### 8.338.2.1 uint16_t loc_gyroTempAcceptReady::batchPerSec

#### 8.338.2.2 uint8_t loc_gyroTempAcceptReady::injectEnable

#### 8.338.2.3 uint16_t loc_gyroTempAcceptReady::samplesPerBatch

## 8.339 loc_IPv4Config Struct Reference

### Data Fields

- uint32_t IPv4Addr
- uint16_t IPv4Port

### 8.339.1 Detailed Description

This structure specifies information regarding the IPv4 address and port.

**Parameters**

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

### 8.339.2 Field Documentation

#### 8.339.2.1 uint32_t loc_IPv4Config::IPv4Addr

#### 8.339.2.2 uint16_t loc_IPv4Config::IPv4Port

## 8.340 loc_IPv4Info Struct Reference

### Data Fields

- uint32_t address
- uint16_t port
- uint8_t TlvPresent

### 8.340.1 Detailed Description

This structure contains LOC Server IPV4 info field.

**Parameters**

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

| | |
|---|---|
| *TlvPresent* | • Tlv Present or not |

### 8.340.2 Field Documentation

#### 8.340.2.1 uint32_t loc_IPv4Info::address

#### 8.340.2.2 uint16_t loc_IPv4Info::port

#### 8.340.2.3 uint8_t loc_IPv4Info::TlvPresent

## 8.341 loc_IPv6Config Struct Reference

### Data Fields

- uint16_t IPv6Addr [8]
- uint32_t IPv6Port

### 8.341.1 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. |

### 8.341.2 Field Documentation

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

#### 8.341.2.2 uint32_t loc_IPv6Config::IPv6Port

## 8.342 loc_IPv6Info Struct Reference

### Data Fields

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

### 8.342.1 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 |

### 8.342.2 Field Documentation

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

#### 8.342.2.2 uint32_t loc_IPv6Info::port

#### 8.342.2.3 uint8_t loc_IPv6Info::TlvPresent

## 8.343 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

### 8.343.1 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 |

### 8.343.2 Field Documentation

#### 8.343.2.1 uint8_t loc_LocApplicationInfo::appNameLength

#### 8.343.2.2 uint8_t loc_LocApplicationInfo::appProviderLength

#### 8.343.2.3 uint8_t loc_LocApplicationInfo::appVersionLength

#### 8.343.2.4 uint8_t loc_LocApplicationInfo::appVersionValid

#### 8.343.2.5 uint8_t* loc_LocApplicationInfo::pAppName

#### 8.343.2.6 uint8_t* loc_LocApplicationInfo::pAppProvider

#### 8.343.2.7 uint8_t* loc_LocApplicationInfo::pAppVersion

## 8.344 loc_precisionDilution Struct Reference

### Data Fields

- uint32_t PDOP
- uint32_t HDOP
- uint32_t VDOP

### 8.344.1 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) |

### 8.344.2 Field Documentation

#### 8.344.2.1 uint32_t loc_precisionDilution::HDOP

#### 8.344.2.2 uint32_t loc_precisionDilution::PDOP

#### 8.344.2.3 uint32_t loc_precisionDilution::VDOP

## 8.345 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

### 8.345.1 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 |

### 8.345.2 Field Documentation

#### 8.345.2.1 float loc_satelliteInfo::azimuth

#### 8.345.2.2 float loc_satelliteInfo::elevation

#### 8.345.2.3 uint16_t loc_satelliteInfo::gnssSvId

#### 8.345.2.4 uint8_t loc_satelliteInfo::healthStatus

#### 8.345.2.5 float loc_satelliteInfo::snr

#### 8.345.2.6 uint8_t loc_satelliteInfo::svInfoMask

#### 8.345.2.7 uint8_t loc_satelliteInfo::svListLen

#### 8.345.2.8 uint32_t loc_satelliteInfo::svStatus

#### 8.345.2.9 uint32_t loc_satelliteInfo::system

#### 8.345.2.10 uint32_t loc_satelliteInfo::validMask

## 8.346 loc_sensorDataUsage Struct Reference

### Data Fields

- uint32_t usageMask
- uint32_t aidingIndicatorMask

### 8.346.1 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 |

### 8.346.2 Field Documentation

#### 8.346.2.1 uint32_t loc_sensorDataUsage::aidingIndicatorMask

#### 8.346.2.2 uint32_t loc_sensorDataUsage::usageMask

## 8.347 loc_SV Struct Reference

### Data Fields

- uint16_t id
- uint32_t system
- uint8_t mask

### 8.347.1 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 |

### 8.347.2 Field Documentation

#### 8.347.2.1 uint16_t loc_SV::id

#### 8.347.2.2 uint8_t loc_SV::mask

#### 8.347.2.3 uint32_t loc_SV::system

## 8.348 loc_SVInfo Struct Reference

### Data Fields

- uint8_t len
- loc_SV * pSV

### 8.348.1 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 |

### 8.348.2 Field Documentation

#### 8.348.2.1 uint8_t loc_SVInfo::len

#### 8.348.2.2 loc_SV* loc_SVInfo::pSV

## 8.349 loc_svUsedforFix Struct Reference

### Data Fields

- uint8_t gnssSvUsedList_len
- uint16_t gnssSvUsedList [255]

### 8.349.1 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 |

### 8.349.2 Field Documentation

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

#### 8.349.2.2 uint8_t loc_svUsedforFix::gnssSvUsedList_len

## 8.350 loc_urlAddr Struct Reference

### Data Fields

- char address [256]
- uint8_t TlvPresent

### 8.350.1 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 |

### 8.350.2 Field Documentation

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

#### 8.350.2.2 uint8_t loc_urlAddr::TlvPresent

## 8.351 loc_URLAddrInfo Struct Reference

### Data Fields

- char urlAddr [256]

### 8.351.1 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 |

### 8.351.2 Field Documentation

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

## 8.352 LocApplicationInfo Struct Reference

### Data Fields

- BYTE appProviderLength
- CHAR * pAppProvider
- BYTE appNameLength
- CHAR * pAppName
- BYTE appVersionValid
- CHAR appVersionLength
- CHAR * pAppVersion

### 8.352.1 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 |

### 8.352.2 Field Documentation

#### 8.352.2.1 BYTE LocApplicationInfo::appNameLength

#### 8.352.2.2 BYTE LocApplicationInfo::appProviderLength

#### 8.352.2.3 CHAR LocApplicationInfo::appVersionLength

#### 8.352.2.4 BYTE LocApplicationInfo::appVersionValid

#### 8.352.2.5 CHAR* LocApplicationInfo::pAppName

#### 8.352.2.6 CHAR* LocApplicationInfo::pAppProvider

#### 8.352.2.7 CHAR* LocApplicationInfo::pAppVersion

## 8.353 LocDelAssDataReq Struct Reference

### Data Fields

- SVInfo * pSVInfo
- GnssData * pGnssData
- CellDb * pCellDb
- ClkInfo * pClkInfo
- BdsSVInfo * pBdsSVInfo

### 8.353.1 Detailed Description

This structure contains LOC delete assist data request

**Parameters**

| | |
|---|---|
| *pSVInfo[IN]* | • Optional parameter<br>• Pointer to struct SVInfo. See SVInfo for more information |
| *pGnssData[IN]* | • Optional parameter<br>• Pointer to struct GnssData. See GnssData for more information |
| *pCellDb[IN]* | • Optional parameter<br>• Pointer to struct CellDb. See CellDb for more information |

| | |
|---|---|
| *pClkInfo[IN]* | • Optional parameter<br>• Pointer to struct ClkInfo. See ClkInfo for more information |
| *pBdsSVInfo[IN]* | • Optional parameter<br>• Pointer to struct BdsSVInfo. See BdsSVInfo for more information |

### 8.353.2 Field Documentation

#### 8.353.2.1 BdsSVInfo* LocDelAssDataReq::pBdsSVInfo

#### 8.353.2.2 CellDb* LocDelAssDataReq::pCellDb

#### 8.353.2.3 ClkInfo* LocDelAssDataReq::pClkInfo

#### 8.353.2.4 GnssData* LocDelAssDataReq::pGnssData

#### 8.353.2.5 SVInfo* LocDelAssDataReq::pSVInfo

## 8.354 LOCEventRegisterReqResp Struct Reference

### Data Fields

- ULONGLONG eventRegister

### 8.354.1 Detailed Description

This structure contains the Parameter for RegisterEvents

**Parameters**

| | |
|---|---|
| *pEventRegMask* | • 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.

### 8.354.2 Field Documentation

#### 8.354.2.1 ULONGLONG LOCEventRegisterReqResp::eventRegister

## 8.355 LOCExtPowerStateReqResp Struct Reference

### Data Fields

- ULONG extPowerState

### 8.355.1 Detailed Description

This structure contains the Parameter External Power Source State.

**Parameters**

| | |
|---|---|
| *pLOCEvent-RegisterReq-Resp* | • 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 |

### 8.355.2 Field Documentation

#### 8.355.2.1 ULONG LOCExtPowerStateReqResp::extPowerState

## 8.356 LocGetServerReq Struct Reference

### Data Fields

- ULONG serverType
- BYTE * pServerAddrTypeMask

### 8.356.1 Detailed Description

This structure contains parameters to gets the location server from the location engine.

**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 |
| *pServerAddr-TypeMask* | • 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 |

### 8.356.2 Field Documentation

#### 8.356.2.1 BYTE* LocGetServerReq::pServerAddrTypeMask

#### 8.356.2.2 ULONG LocGetServerReq::serverType

## 8.357 LocInjectPositionReq Struct Reference

### Data Fields

- ULONGLONG * pLatitude
- ULONGLONG * pLongitude
- ULONG * pHorUncCircular
- BYTE * pHorConfidence
- ULONG * pHorReliability
- ULONG * pAltitudeWrtEllipsoid
- ULONG * pAltitudeWrtMeanSeaLevel
- ULONG * pVertUnc
- BYTE * pVertConfidence
- ULONG * pVertReliability
- altitudeSrcInfo * pAltitudeSrcInfo
- ULONGLONG * pTimestampUtc
- ULONG * pTimestampAge
- ULONG * pPositionSrc
- ULONG * pRawHorUncCircular
- BYTE * pRawHorConfidence

### 8.357.1 Detailed Description

This structure contains LOC Inject Position parameters

**Parameters**

| | |
|---|---|
| *pLatitude* | • 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. |
| *pLongitude* | • 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. |
| *pHorUncCircular* | • Optional parameter<br>• Horizontal position uncertainty.<br>• Units - Meters<br>• Note - This field must be specified together with pLatitude and pLongitude. |
| *pHorConfidence* | • 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. |
| *pHorReliability* | • Optional parameter |

- Values
  - 0 - Location reliability is not set.
  - 1 - Location reliability is very low; use it at your own risk
  - 2 - Location reliability is low; little or no cross-checking is possible.
  - 3 - Location reliability is medium; limited cross-check passed
  - 4 - Location reliability is high; strong cross-check passed

**Parameters**

| | |
|---|---|
| *pAltitudeWrt-Ellipsoid* | • Optional parameter<br>• Altitude With Respect to Ellipsoid.<br>• Units - Meters<br>• Values<br>– Positive - height<br>– Negative = depth |
| *pAltitudeWrt-MeanSeaLevel* | • Optional parameter<br>• Altitude With Respect to Sea Level.<br>• Units - Meters |
| *pVertUnc* | • Optional parameter<br>• Vertical uncertainty.<br>• Units - Meters<br>• Note - This is mandatory if either pAltitudeWrtEllipsoid or pAltitudeWrtMeanSeaLevel is specified. |
| *pVertConfidence* | • 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. |
| *pVertReliability* | • Optional parameter |

- Values
  - 0 - Location reliability is not set.
  - 1 - Location reliability is very low; use it at your own risk.
  - 2 - Location reliability is low; little or no cross-checking is possible
  - 3 - Location reliability is medium; limited cross-check passed
  - 4 - Location reliability is high; strong cross-check passed

**Parameters**

| | |
|---|---|
| *pAltitudeSrcInfo* | • Optional parameter |

- Pointer to struct altitudeSrcInfo. See altitudeSrcInfo for more information

**Parameters**

| | |
|---|---|
| *pTimestampUtc* | • Optional parameter<br>• UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970 |
| *pTimestampAge* | • Optional parameter<br>• Position age, which is an estimate of how long ago this fix was made.<br>• Units - Milliseconds |
| *pPositionSrc* | • 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. |
| *pRawHorUnc-Circular* | • Optional parameter<br>• Horizontal position uncertainty (circular) without any optimization.<br>• Units - Meters |
| *pRawHor-Confidence* | • 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. |

## 8.357.2 Field Documentation

### 8.357.2.1 altitudeSrcInfo∗ LocInjectPositionReq::pAltitudeSrcInfo

### 8.357.2.2 ULONG∗ LocInjectPositionReq::pAltitudeWrtEllipsoid

### 8.357.2.3 ULONG∗ LocInjectPositionReq::pAltitudeWrtMeanSeaLevel

#### 8.357.2.4 BYTE* LocInjectPositionReq::pHorConfidence

#### 8.357.2.5 ULONG* LocInjectPositionReq::pHorReliability

#### 8.357.2.6 ULONG* LocInjectPositionReq::pHorUncCircular

#### 8.357.2.7 ULONGLONG* LocInjectPositionReq::pLatitude

#### 8.357.2.8 ULONGLONG* LocInjectPositionReq::pLongitude

#### 8.357.2.9 ULONG* LocInjectPositionReq::pPositionSrc

#### 8.357.2.10 BYTE* LocInjectPositionReq::pRawHorConfidence

#### 8.357.2.11 ULONG* LocInjectPositionReq::pRawHorUncCircular

#### 8.357.2.12 ULONG* LocInjectPositionReq::pTimestampAge

#### 8.357.2.13 ULONGLONG* LocInjectPositionReq::pTimestampUtc

#### 8.357.2.14 BYTE* LocInjectPositionReq::pVertConfidence

#### 8.357.2.15 ULONG* LocInjectPositionReq::pVertReliability

#### 8.357.2.16 ULONG* LocInjectPositionReq::pVertUnc

## 8.358 LocInjectSensorDataReq Struct Reference

### Data Fields

- ULONG * pOpaqueIdentifier
- sensorData * pAcceleroData
- sensorData * pGyroData
- ULONG * pAcceleroTimeSrc
- ULONG * pGyroTimeSrc
- tempratureData * pAcceleroTempData
- tempratureData * pGyroTempData

### 8.358.1 Detailed Description

This structure contains parameters to inject sensor data into the GNSS location engine

**Parameters**

| | |
|---|---|
| *pOpaque-Identifier* | • 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. |
| *pAcceleroData* | • Optional parameter<br>• Pointer to struct sensorData. See sensorData for more information |

| | |
|---|---|
| *pGyroData* | • Optional parameter<br>• Pointer to struct sensorData. See sensorData for more information |
| *pAcceleroTime-Src* | • 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 |
| *pGyroTimeSrc* | • 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 |
| *pAcceleroTemp-Data* | • Optional parameter<br>• Pointer to struct tempratureData. See tempratureData for more information |
| *pGyroTempData* | • Optional parameter<br>• Pointer to struct tempratureData. See tempratureData for more information |

### 8.358.2 Field Documentation

#### 8.358.2.1 sensorData* LocInjectSensorDataReq::pAcceleroData

#### 8.358.2.2 tempratureData* LocInjectSensorDataReq::pAcceleroTempData

#### 8.358.2.3 ULONG* LocInjectSensorDataReq::pAcceleroTimeSrc

#### 8.358.2.4 sensorData* LocInjectSensorDataReq::pGyroData

#### 8.358.2.5 tempratureData* LocInjectSensorDataReq::pGyroTempData

#### 8.358.2.6 ULONG* LocInjectSensorDataReq::pGyroTimeSrc

#### 8.358.2.7 ULONG* LocInjectSensorDataReq::pOpaqueIdentifier

## 8.359 LocSetCradleMountReq Struct Reference

### Data Fields

- ULONG state
- BYTE * pConfidence

#### 8.359.1 Detailed Description

This structure contains parameters to set current cradle mount configuration

**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 |
| *pConfidence* | • Cradle Mount Confidence (Optional)<br>• Confidence in the Cradle Mount state expressed as a percentage.<br>• Range - 0 to 100 |

#### 8.359.2 Field Documentation

##### 8.359.2.1 BYTE* LocSetCradleMountReq::pConfidence

##### 8.359.2.2 ULONG LocSetCradleMountReq::state

## 8.360 LocSetServerReq Struct Reference

### Data Fields

- ULONG serverType
- IPv4Config * pIPv4Config
- IPv6Config * pIPv6Config
- URLAddrInfo * pURLAddr

#### 8.360.1 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 IPv4Config for more information |
| *pIPv6Config* | • IPv6 address and port (Optional)<br>• See IPv6Config for more information |
| *pURLAddr* | • Uniform Resource Locator (Optional)<br>• See URLAddrInfo for more information |

### 8.360.2 Field Documentation

#### 8.360.2.1 IPv4Config* LocSetServerReq::pIPv4Config

#### 8.360.2.2 IPv6Config* LocSetServerReq::pIPv6Config

#### 8.360.2.3 URLAddrInfo* LocSetServerReq::pURLAddr

#### 8.360.2.4 ULONG LocSetServerReq::serverType

## 8.361 LOCStartReq Struct Reference

### Data Fields

- BYTE SessionId
- ULONG * pRecurrenceType
- ULONG * pHorizontalAccuracyLvl
- ULONG * pIntermediateReportState
- ULONG * pMinIntervalTime
- struct LocApplicationInfo * pApplicationInfo
- ULONG * pConfigAltitudeAssumed

### 8.361.1 Detailed Description

This structure contains the LOC Start Request

**Parameters**

| | |
|---|---|
| *SessionId[IN]* | • ID of the session as identified by the control point.<br>• Range: 0 to 255 |
| *pRecurrence-Type[IN]* | • 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[IN]* | • 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[IN]* | • 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[IN]* | • 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[IN]* | • Optional Parameter<br>• LOC Application Parameters<br>• See LocApplicationInfo for more information |
| *pConfigAltitude-Assumed[IN]* | • 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 |

### 8.361.2 Field Documentation

#### 8.361.2.1 struct LocApplicationInfo* LOCStartReq::pApplicationInfo

#### 8.361.2.2 ULONG* LOCStartReq::pConfigAltitudeAssumed

#### 8.361.2.3 ULONG* LOCStartReq::pHorizontalAccuracyLvl

#### 8.361.2.4 ULONG* LOCStartReq::pIntermediateReportState

#### 8.361.2.5 ULONG* LOCStartReq::pMinIntervalTime

#### 8.361.2.6 ULONG* LOCStartReq::pRecurrenceType

#### 8.361.2.7 BYTE LOCStartReq::SessionId

## 8.362 LOCStopReq Struct Reference

### Data Fields

- BYTE sessionId

### 8.362.1 Detailed Description

This structure contains the LOC Stop Request

**Parameters**

| | |
|---|---|
| *sessionId* | • ID of the session as identified by the control point.<br>• Range: 0 to 255 |

### 8.362.2 Field Documentation

#### 8.362.2.1 BYTE LOCStopReq::sessionId

## 8.363 LteCQIParm Struct Reference

### Data Fields

- BYTE ValidityCW0
- BYTE CQIValueCW0
- BYTE ValidityCW1
- BYTE CQIValueCW1

### 8.363.1 Detailed Description

This structure contains information about the SLQSSwiGetLteCQI response parameters.

**Parameters**

| | |
|---|---|
| *pValidityCW0[OUT]* | • Values<br>– 0- Invalid.<br>– 1- Valid. |
| *pCQIValueCW0[OUT]* | • Values<br>– Range 0∼15 |
| *pValidityCW1[OUT]* | • Values<br>– 0- Invalid.<br>– 1- Valid. |

| | |
|---|---|
| *pCQIValueCW1[OUT]* | • Values<br>– Range 0∼15 |

### 8.363.2 Field Documentation

#### 8.363.2.1 BYTE LteCQIParm::CQIValueCW0

#### 8.363.2.2 BYTE LteCQIParm::CQIValueCW1

#### 8.363.2.3 BYTE LteCQIParm::ValidityCW0

#### 8.363.2.4 BYTE LteCQIParm::ValidityCW1

## 8.364 lteEARFCN Struct Reference

### Data Fields

- BYTE status
- ULONG earfcn0
- ULONG earfcn1

### 8.364.1 Detailed Description

This structure contains the parameters for WCDMA UARCFN.

**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. |

### 8.364.2 Field Documentation

#### 8.364.2.1 ULONG lteEARFCN::earfcn0

#### 8.364.2.2 ULONG lteEARFCN::earfcn1

#### 8.364.2.3 BYTE lteEARFCN::status

## 8.365 lteGsmCellInfo Struct Reference

### Data Fields

- BYTE cellReselPriority
- BYTE threshGsmHigh
- BYTE threshGsmLow
- BYTE nccPermitted
- BYTE cells_len
- gsmCellInfo GsmCellInfo [255]

## 8.365.1 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[MAX_DESCRIPTION_LENGTH]* | • See gsmCellInfo for more information. |

## 8.365.2 Field Documentation

### 8.365.2.1 BYTE lteGsmCellInfo::cellReselPriority

### 8.365.2.2 BYTE lteGsmCellInfo::cells_len

### 8.365.2.3 gsmCellInfo lteGsmCellInfo::GsmCellInfo[255]

### 8.365.2.4 BYTE lteGsmCellInfo::nccPermitted

#### 8.365.2.5 BYTE lteGsmCellInfo::threshGsmHigh

#### 8.365.2.6 BYTE lteGsmCellInfo::threshGsmLow

## 8.366 LTEInfo Struct Reference

### Data Fields

- BYTE band
- BYTE bandwidth
- WORD RXChan
- WORD TXChan
- BYTE emmState
- BYTE emmSubState
- BYTE emmConnState

### 8.366.1 Detailed Description

Structure for storing the LTE information for the device.

**Parameters**

| | |
|---|---|
| *band* | • LTE Band<br>– 1 ∼ 41 (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 |

## 8.366.2 Field Documentation

### 8.366.2.1 BYTE LTEInfo::band

#### 8.366.2.2 BYTE LTEInfo::bandwidth

#### 8.366.2.3 BYTE LTEInfo::emmConnState

#### 8.366.2.4 BYTE LTEInfo::emmState

#### 8.366.2.5 BYTE LTEInfo::emmSubState

#### 8.366.2.6 WORD LTEInfo::RXChan

#### 8.366.2.7 WORD LTEInfo::TXChan

## 8.367 LTEInfoInterfreq Struct Reference

### Data Fields

- BYTE ueInIdle
- BYTE freqsLen
- infoInterFreq InfoInterfreq [255]

### 8.367.1 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[MAX_DESCRIPTION_LENGTH]* | • See infoInterFreq for more information. |

### 8.367.2 Field Documentation

#### 8.367.2.1 BYTE LTEInfoInterfreq::freqsLen

#### 8.367.2.2 infoInterFreq LTEInfoInterfreq::InfoInterfreq[255]

#### 8.367.2.3 BYTE LTEInfoInterfreq::ueInIdle

## 8.368 LTEInfoIntrafreq Struct Reference

### Data Fields

- BYTE ueInIdle
- BYTE plmn [3]
- WORD tac

- ULONG globalCellId
- WORD earfcn
- WORD servingCellId
- BYTE cellReselPriority
- BYTE sNonIntraSearch
- BYTE threshServingLow
- BYTE sIntraSearch
- BYTE cellsLen
- cellParams CellParams [255]

### 8.368.1 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[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 |

| | |
|---|---|
| *threshServing-Low* | • 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[MAX_DESCRIPTION_LENGTH]* | • See cellParams for more information. |

### 8.368.2 Field Documentation

#### 8.368.2.1 cellParams LTEInfoIntrafreq::CellParams[255]

#### 8.368.2.2 BYTE LTEInfoIntrafreq::cellReselPriority

#### 8.368.2.3 BYTE LTEInfoIntrafreq::cellsLen

#### 8.368.2.4 WORD LTEInfoIntrafreq::earfcn

#### 8.368.2.5 ULONG LTEInfoIntrafreq::globalCellId

#### 8.368.2.6 BYTE LTEInfoIntrafreq::plmn[3]

#### 8.368.2.7 WORD LTEInfoIntrafreq::servingCellId

#### 8.368.2.8 BYTE LTEInfoIntrafreq::sIntraSearch

#### 8.368.2.9 BYTE LTEInfoIntrafreq::sNonIntraSearch

#### 8.368.2.10 WORD LTEInfoIntrafreq::tac

#### 8.368.2.11 BYTE LTEInfoIntrafreq::threshServingLow

#### 8.368.2.12 BYTE LTEInfoIntrafreq::ueInIdle

## 8.369 LTEInfoNeighboringGSM Struct Reference

### Data Fields

- BYTE ueInIdle

- BYTE freqsLen
- lteGsmCellInfo LteGsmCellInfo [255]

### 8.369.1 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[-MAX_DESCRIP-TION_LENGTH]* | • See lteGsmCellInfo for more information. |

### 8.369.2 Field Documentation

#### 8.369.2.1 BYTE LTEInfoNeighboringGSM::freqsLen

#### 8.369.2.2 lteGsmCellInfo LTEInfoNeighboringGSM::LteGsmCellInfo[255]

#### 8.369.2.3 BYTE LTEInfoNeighboringGSM::ueInIdle

## 8.370 LTEInfoNeighboringWCDMA Struct Reference

### Data Fields

- BYTE ueInIdle
- BYTE freqsLen
- lteWcdmaCellInfo LTEWCDMACellInfo [255]

### 8.370.1 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. |

| | |
|---|---|
| *LTEWCDMA-CellInfo[MAX_D-ESCRIPTION_L-ENGTH]* | • See lteWcdmaCellInfo for more information. |

### 8.370.2 Field Documentation

#### 8.370.2.1 BYTE LTEInfoNeighboringWCDMA::freqsLen

#### 8.370.2.2 lteWcdmaCellInfo LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

#### 8.370.2.3 BYTE LTEInfoNeighboringWCDMA::ueInIdle

## 8.371 LteNasReleaseInfo_s Struct Reference

### Data Fields

- BYTE nas_release
- BYTE nas_major
- BYTE nas_minor

### 8.371.1 Detailed Description

This structure contains LTE Nas Release Infomation

**Parameters**

| | |
|---|---|
| *nas_release* | • LTE NAS release |
| *nas_major* | • LTE NAS version major |
| *nas_minor* | • LTE NAS version minor |

### 8.371.2 Field Documentation

#### 8.371.2.1 BYTE LteNasReleaseInfo_s::nas_major

#### 8.371.2.2 BYTE LteNasReleaseInfo_s::nas_minor

#### 8.371.2.3 BYTE LteNasReleaseInfo_s::nas_release

## 8.372 LTEOperationMode Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG * pLTEOperationMode

### 8.372.1 Detailed Description

This structure contains the LTEOperationMode response parameters.

**Parameters**

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

### 8.372.2 Field Documentation

#### 8.372.2.1 ULONG* LTEOperationMode::pLTEOperationMode

#### 8.372.2.2 BYTE LTEOperationMode::TlvPresent

## 8.373 ltePCI Struct Reference

### Data Fields

- BYTE status
- ULONG earfcn
- ULONG pci

### 8.373.1 Detailed Description

This structure contains the parameters for WCDMA UARCFN.

**Parameters**

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

### 8.373.2 Field Documentation

#### 8.373.2.1 ULONG ltePCI::earfcn

#### 8.373.2.2 ULONG ltePCI::pci

#### 8.373.2.3 BYTE ltePCI::status

## 8.374 lteRsrpinformation Struct Reference

### Data Fields

- SHORT rsrplevel

### 8.374.1 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). |

### 8.374.2 Field Documentation

#### 8.374.2.1 SHORT lteRsrpinformation::rsrplevel

## 8.375 LTERSRPThresh Struct Reference

### Data Fields

- BYTE LTERSRPThreshListLen
- WORD * pLTERSRPThreshList

### 8.375.1 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). |

### 8.375.2 Field Documentation

#### 8.375.2.1 BYTE LTERSRPThresh::LTERSRPThreshListLen

#### 8.375.2.2 WORD* LTERSRPThresh::pLTERSRPThreshList

## 8.376 LTERSRQThresh Struct Reference

### Data Fields

- BYTE LTERSRQThreshListLen
- WORD * pLTERSRQThreshList

### 8.376.1 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) |

### 8.376.2 Field Documentation

#### 8.376.2.1 BYTE LTERSRQThresh::LTERSRQThreshListLen

#### 8.376.2.2 WORD* LTERSRQThresh::pLTERSRQThreshList

## 8.377 LTERSSIThresh Struct Reference

### Data Fields

- BYTE LTERSSIThreshListLen
- WORD * pLTERSSIThreshList

### 8.377.1 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) |

### 8.377.2 Field Documentation

#### 8.377.2.1 BYTE LTERSSIThresh::LTERSSIThreshListLen

#### 8.377.2.2 WORD* LTERSSIThresh::pLTERSSIThreshList

## 8.378 LteSccRxInfoResp Struct Reference

### Data Fields

- SccRxInfo * pSccRxInfo

### 8.378.1 Detailed Description

This structure contains information about the SLQSSwiGetLteSccRxInfo response parameters.

**Parameters**

| | |
|---|---|
| *pSccRxInfo* | [Optional]<br>• See SccRxInfo for more information |

### 8.378.2 Field Documentation

#### 8.378.2.1 SccRxInfo* LteSccRxInfoResp::pSccRxInfo

## 8.379 LTESigRptCfg Struct Reference

### Data Fields

- BYTE rptRate
- BYTE avgPeriod

### 8.379.1 Detailed Description

This structure contains LTE Signal Report Config parameters.

**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 |

### 8.379.2 Field Documentation

#### 8.379.2.1 BYTE LTESigRptCfg::avgPeriod

#### 8.379.2.2 BYTE LTESigRptCfg::rptRate

## 8.380 LTESigRptConfig Struct Reference

### Data Fields

- BYTE rptRate
- BYTE avgPeriod

### 8.380.1 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 |

### 8.380.2 Field Documentation

#### 8.380.2.1 BYTE LTESigRptConfig::avgPeriod

#### 8.380.2.2 BYTE LTESigRptConfig::rptRate

## 8.381 lteSnrinformation Struct Reference

### Data Fields

- SHORT snrlevel

### 8.381.1 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. |

### 8.381.2 Field Documentation

#### 8.381.2.1 SHORT lteSnrinformation::snrlevel

## 8.382 LTESNRThresh Struct Reference

### Data Fields

- BYTE LTESNRThresListLen
- SHORT * pLTESNRThresList

### 8.382.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

**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 |

### 8.382.2 Field Documentation

#### 8.382.2.1 BYTE LTESNRThresh::LTESNRThresListLen

#### 8.382.2.2 SHORT* LTESNRThresh::pLTESNRThresList

## 8.383 LTESNRThreshold Struct Reference

### Data Fields

- BYTE LTESNRThreshListLen
- WORD * pLTESNRThreshList

### 8.383.1 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). |

### 8.383.2 Field Documentation

#### 8.383.2.1 BYTE LTESNRThreshold::LTESNRThreshListLen

#### 8.383.2.2 WORD* LTESNRThreshold::pLTESNRThreshList

## 8.384 LTESSInfo Struct Reference

### Data Fields

- INT8 rssi
- INT8 rsrq
- SHORT rsrp
- SHORT snr

### 8.384.1 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, |

### 8.384.2 Field Documentation

#### 8.384.2.1 SHORT LTESSInfo::rsrp

#### 8.384.2.2 INT8 LTESSInfo::rsrq

#### 8.384.2.3 INT8 LTESSInfo::rssi

#### 8.384.2.4 SHORT LTESSInfo::snr

## 8.385 lteSSInfo Struct Reference

### Data Fields

- int8_t rssi
- int8_t rsrq
- int16_t rsrp

- int16_t snr

### 8.385.1 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, |

### 8.385.2 Field Documentation

#### 8.385.2.1 int16_t lteSSInfo::rsrp

#### 8.385.2.2 int8_t lteSSInfo::rsrq

#### 8.385.2.3 int8_t lteSSInfo::rssi

#### 8.385.2.4 int16_t lteSSInfo::snr

## 8.386 LTESysInfo Struct Reference

### Data Fields

- sysInfoCommon sysInfoLTE
- BYTE lacValid
- WORD lac
- BYTE cellIdValid
- ULONG cellId
- BYTE regRejectInfoValid
- BYTE rejectSrvDomain
- BYTE rejCause
- BYTE networkIdValid
- BYTE MCC [3]
- BYTE MNC [3]
- BYTE tacValid
- WORD tac

### 8.386.1 Detailed Description

Structure for storing the LTE System Information.

**Parameters**

| | |
|---|---|
| *sysInfoLTE* | • See 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[PLMN_LENGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters |
| *MNC[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 |

## 8.386.2 Field Documentation

### 8.386.2.1 ULONG LTESysInfo::cellId

### 8.386.2.2 BYTE LTESysInfo::cellIdValid

### 8.386.2.3 WORD LTESysInfo::lac

### 8.386.2.4 BYTE LTESysInfo::lacValid

### 8.386.2.5 BYTE LTESysInfo::MCC[3]

### 8.386.2.6 BYTE LTESysInfo::MNC[3]

### 8.386.2.7 BYTE LTESysInfo::networkIdValid

### 8.386.2.8 BYTE LTESysInfo::regRejectInfoValid

### 8.386.2.9 BYTE LTESysInfo::rejCause

### 8.386.2.10 BYTE LTESysInfo::rejectSrvDomain

### 8.386.2.11 sysInfoCommon LTESysInfo::sysInfoLTE

#### 8.386.2.12 WORD LTESysInfo::tac

#### 8.386.2.13 BYTE LTESysInfo::tacValid

## 8.387 lteWcdmaCellInfo Struct Reference

### Data Fields

- WORD uarfcn
- BYTE cellReselPriority
- WORD threshXhigh
- WORD threshXlow
- BYTE cellsLen
- wcdmaCellInfo WCDMACellInfo [255]

### 8.387.1 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[MAX_DESCRIPTION_LENGTH]* | • See wcdmaCellInfo for more information. |

### 8.387.2 Field Documentation

#### 8.387.2.1 BYTE lteWcdmaCellInfo::cellReselPriority

#### 8.387.2.2 BYTE lteWcdmaCellInfo::cellsLen

#### 8.387.2.3 WORD lteWcdmaCellInfo::threshXhigh

#### 8.387.2.4 WORD lteWcdmaCellInfo::threshXlow

#### 8.387.2.5 WORD lteWcdmaCellInfo::uarfcn

#### 8.387.2.6 wcdmaCellInfo lteWcdmaCellInfo::WCDMACellInfo[255]

## 8.388 messageModeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSMessageModeInfo MessageModeInfo

### 8.388.1 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 |

### 8.388.2 Field Documentation

#### 8.388.2.1 sMSMessageModeInfo messageModeTlv::MessageModeInfo

#### 8.388.2.2 uint8_t messageModeTlv::TlvPresent

## 8.389 messageWaitingInfoContent Struct Reference

### Data Fields

- BYTE msgType
- BYTE activeInd
- BYTE msgCount

### 8.389.1 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 |

### 8.389.2 Field Documentation

#### 8.389.2.1 BYTE messageWaitingInfoContent::activeInd

#### 8.389.2.2 BYTE messageWaitingInfoContent::msgCount

#### 8.389.2.3 BYTE messageWaitingInfoContent::msgType

## 8.390 minBasedIMSI Struct Reference

### Data Fields

- BYTE mccM [3]
- WORD imsiM1112
- BYTE imsiMS1 [7]
- BYTE imsiMS2 [3]

### 8.390.1 Detailed Description

This structure contains the parameters for Min based IMSI Information

**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 |

### 8.390.2 Field Documentation

#### 8.390.2.1 WORD minBasedIMSI::imsiM1112

#### 8.390.2.2 BYTE minBasedIMSI::imsiMS1[7]

#### 8.390.2.3 BYTE minBasedIMSI::imsiMS2[3]

#### 8.390.2.4 BYTE minBasedIMSI::mccM[3]

## 8.391 mitigationDevList Struct Reference

### Data Fields

- BYTE mitigationDevIdLen
- CHAR mitigationDevId [255]
- BYTE maxMitigationLevel

### 8.391.1 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. |

### 8.391.2 Field Documentation

#### 8.391.2.1 BYTE mitigationDevList::maxMitigationLevel

#### 8.391.2.2 CHAR mitigationDevList::mitigationDevId[255]

#### 8.391.2.3 BYTE mitigationDevList::mitigationDevIdLen

## 8.392 MNRInfo Struct Reference

### Data Fields

- WORD mcc
- WORD mnc
- ULONG rat

### 8.392.1 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 |

### 8.392.2 Field Documentation

#### 8.392.2.1 WORD MNRInfo::mcc

#### 8.392.2.2 WORD MNRInfo::mnc

#### 8.392.2.3 ULONG MNRInfo::rat

## 8.393 ModifyProfileIn Struct Reference

### Data Fields

- BYTE * pProfileID
- BYTE * pProfileType
- QmiProfileInfo curProfile

### 8.393.1 Detailed Description

This structure contains input parameters for SLQSModifyProfile

**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<br>– NULL is not allowed |

| | |
|---|---|
| *curProfile* | • Contains Union of profile(3GPP/3GPP2) structures |

### 8.393.2 Field Documentation

#### 8.393.2.1 QmiProfileInfo ModifyProfileIn::curProfile

#### 8.393.2.2 BYTE* ModifyProfileIn::pProfileID

#### 8.393.2.3 BYTE* ModifyProfileIn::pProfileType

## 8.394 ModifyProfileOut Struct Reference

### Data Fields

- USHORT * pExtErrorCode

### 8.394.1 Detailed Description

This structure contains out parameters for 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. |

### 8.394.2 Field Documentation

#### 8.394.2.1 USHORT* ModifyProfileOut::pExtErrorCode

## 8.395 msgWaitingInfo Struct Reference

### Data Fields

- BYTE numInstances
- messageWaitingInfoContent msgWaitInfo [0xFF]

### 8.395.1 Detailed Description

This structure holds information related to message waiting information

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of sets of the elements in structure messageWaitingInfoContent |

| | |
|---|---|
| *msgWaitInfo* | • Pointer to structure of messageWaitingInfoContent.<br>– See messageWaitingInfoContent for more information. |

### 8.395.2 Field Documentation

#### 8.395.2.1 messageWaitingInfoContent msgWaitingInfo::msgWaitInfo[0xFF]

#### 8.395.2.2 BYTE msgWaitingInfo::numInstances

## 8.396 mtuSize3gppTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- WORD MTUSize3gpp

### 8.396.1 Detailed Description

This structure contains the parameters for 3GPP MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *MTUSize3gpp* | 3GPP MTU size<br>• Size of 3GPP MTU |

### 8.396.2 Field Documentation

#### 8.396.2.1 WORD mtuSize3gppTlv::MTUSize3gpp

#### 8.396.2.2 BYTE mtuSize3gppTlv::TlvPresent

## 8.397 MTUSizeReq Struct Reference

### Data Fields

- WORD MTUSize

### 8.397.1 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. |

### 8.397.2 Field Documentation

#### 8.397.2.1 WORD MTUSizeReq::MTUSize

## 8.398 namName Struct Reference

### Data Fields

- BYTE namNameLen
- BYTE namName [12]

### 8.398.1 Detailed Description

This structure contains the parameters for NAM Name Information

**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. |

### 8.398.2 Field Documentation

#### 8.398.2.1 BYTE namName::namName[12]

#### 8.398.2.2 BYTE namName::namNameLen

## 8.399 nas_acqOrderPref Struct Reference

### Data Fields

- uint8_t acqOrdeLen
- uint8_t * pAcqOrder

### 8.399.1 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 |

### 8.399.2 Field Documentation

#### 8.399.2.1 uint8_t nas_acqOrderPref::acqOrdeLen

#### 8.399.2.2 uint8_t* nas_acqOrderPref::pAcqOrder

## 8.400 nas_ActPilotPNElement Struct Reference

### Data Fields

- uint16_t ActSetPilotPN
- uint8_t ActSetPilotPNStrength

### 8.400.1 Detailed Description

This structure contains pilot PN Element.

**Parameters**

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

### 8.400.2 Field Documentation

#### 8.400.2.1 uint16_t nas_ActPilotPNElement::ActSetPilotPN

#### 8.400.2.2 uint8_t nas_ActPilotPNElement::ActSetPilotPNStrength

## 8.401 nas_AddCDMASysInfo Struct Reference

### Data Fields

- uint16_t geoSysIdx
- uint16_t regPrd

### 8.401.1 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. |

### 8.401.2 Field Documentation

#### 8.401.2.1 uint16_t nas_AddCDMASysInfo::geoSysIdx

#### 8.401.2.2 uint16_t nas_AddCDMASysInfo::regPrd

## 8.402 nas_AddSysInfo Struct Reference

### Data Fields

- uint16_t geoSysIdx
- uint32_t cellBroadcastCap

### 8.402.1 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 |

### 8.402.2 Field Documentation

#### 8.402.2.1 uint32_t nas_AddSysInfo::cellBroadcastCap

#### 8.402.2.2 uint16_t nas_AddSysInfo::geoSysIdx

## 8.403 nas_CallBarringSysInfo Struct Reference

### Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

### 8.403.1 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 |

### 8.403.2 Field Documentation

#### 8.403.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

#### 8.403.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

## 8.404 nas_callBarStatus Struct Reference

### Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

### 8.404.1 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 |

### 8.404.2 Field Documentation

#### 8.404.2.1 uint32_t nas_callBarStatus::csBarStatus

#### 8.404.2.2 uint32_t nas_callBarStatus::psBarStatus

## 8.405 nas_CDMAChannel Struct Reference

### Data Fields

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

### 8.405.1 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 |

### 8.405.2 Field Documentation

#### 8.405.2.1 uint16_t nas_CDMAChannel::priChA

#### 8.405.2.2 uint16_t nas_CDMAChannel::priChB

#### 8.405.2.3 uint16_t nas_CDMAChannel::secChA

#### 8.405.2.4 uint16_t nas_CDMAChannel::secChB

## 8.406 nas_CDMAECIOThresh Struct Reference

### Data Fields

- uint8_t CDMAECIOThreshListLen
- int16_t * pCDMAECIOThreshList

### 8.406.1 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) |

### 8.406.2 Field Documentation

#### 8.406.2.1 uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen

#### 8.406.2.2 int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList

## 8.407 nas_CDMAInfo Struct Reference

### Data Fields

- uint16_t sid

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

### 8.407.1 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 |

### 8.407.2 Field Documentation

#### 8.407.2.1 uint16_t nas_CDMAInfo::baseId

#### 8.407.2.2 uint32_t nas_CDMAInfo::baseLat

#### 8.407.2.3 uint32_t nas_CDMAInfo::baseLong

#### 8.407.2.4 uint16_t nas_CDMAInfo::nid

#### 8.407.2.5 uint16_t nas_CDMAInfo::refpn

#### 8.407.2.6 uint16_t nas_CDMAInfo::sid

## 8.408 nas_CDMARSSIThresh Struct Reference

### Data Fields

- uint8_t CDMARSSIThreshListLen
- int16_t * pCDMARSSIThreshList

### 8.408.1 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). |

### 8.408.2 Field Documentation

#### 8.408.2.1 uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen

#### 8.408.2.2 int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList

## 8.409 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]

### 8.409.1 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}. |

### 8.409.2 Field Documentation

#### 8.409.2.1 uint16_t nas_CDMASysInfo::baseId

#### 8.409.2.2 uint32_t nas_CDMASysInfo::baseLat

#### 8.409.2.3 uint32_t nas_CDMASysInfo::baseLong

#### 8.409.2.4 uint8_t nas_CDMASysInfo::bsInfoValid

#### 8.409.2.5 uint8_t nas_CDMASysInfo::bsPRev

#### 8.409.2.6 uint8_t nas_CDMASysInfo::bsPRevValid

#### 8.409.2.7 uint8_t nas_CDMASysInfo::ccsSupported

#### 8.409.2.8 uint8_t nas_CDMASysInfo::ccsSupportedValid

#### 8.409.2.9 uint8_t nas_CDMASysInfo::cdmaSysIdValid

#### 8.409.2.10 uint8_t nas_CDMASysInfo::isSysPrlMatch

#### 8.409.2.11 uint8_t nas_CDMASysInfo::isSysPrlMatchValid

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

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

#### 8.409.2.14 uint16_t nas_CDMASysInfo::networkID

#### 8.409.2.15 uint8_t nas_CDMASysInfo::networkIdValid

#### 8.409.2.16 uint16_t nas_CDMASysInfo::packetZone

#### 8.409.2.17 uint8_t nas_CDMASysInfo::packetZoneValid

#### 8.409.2.18 uint8_t nas_CDMASysInfo::pRevInUse

#### 8.409.2.19 uint8_t nas_CDMASysInfo::pRevInUseValid

#### 8.409.2.20 nas_sysInfoCommon nas_CDMASysInfo::sysInfoCDMA

#### 8.409.2.21 uint16_t nas_CDMASysInfo::systemID

## 8.410 nas_CDMASysInfoExt Struct Reference

### Data Fields

- uint16_t MCC
- uint8_t imsi_11_12

### 8.410.1 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 |

### 8.410.2 Field Documentation

#### 8.410.2.1 uint8_t nas_CDMASysInfoExt::imsi_11_12

#### 8.410.2.2 uint16_t nas_CDMASysInfoExt::MCC

## 8.411 nas_cellParams Struct Reference

### Data Fields

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

### 8.411.1 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. |

### 8.411.2 Field Documentation

#### 8.411.2.1 uint16_t nas_cellParams::pci

#### 8.411.2.2 int16_t nas_cellParams::rsrp

#### 8.411.2.3 int16_t nas_cellParams::rsrq

#### 8.411.2.4 int16_t nas_cellParams::rssi

#### 8.411.2.5 int16_t nas_cellParams::srxlev

## 8.412 nas_ciotAcqOrderPref Struct Reference

### Data Fields

- uint8_t ciotAcqOrderLen
- uint32_t * pCiotAcqOrder

### 8.412.1 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 |

### 8.412.2 Field Documentation

#### 8.412.2.1 uint8_t nas_ciotAcqOrderPref::ciotAcqOrderLen

#### 8.412.2.2 uint32_t* nas_ciotAcqOrderPref::pCiotAcqOrder

## 8.413 nas_CommInfo Struct Reference

### Data Fields

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

### 8.413.1 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 |

### 8.413.2 Field Documentation

#### 8.413.2.1 uint8_t nas_CommInfo::imsRegState

#### 8.413.2.2 uint8_t nas_CommInfo::modemMode

#### 8.413.2.3 uint8_t nas_CommInfo::psState

#### 8.413.2.4 uint8_t nas_CommInfo::systemMode

#### 8.413.2.5 int8_t nas_CommInfo::temperature

## 8.414 nas_CSGID Struct Reference

### Data Fields

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

### 8.414.1 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 |

### 8.414.2 Field Documentation

#### 8.414.2.1 uint32_t nas_CSGID::id

#### 8.414.2.2 uint16_t nas_CSGID::mcc

#### 8.414.2.3 uint16_t nas_CSGID::mnc

#### 8.414.2.4 uint8_t nas_CSGID::mncPcsDigits

#### 8.414.2.5 uint8_t nas_CSGID::rat

## 8.415 nas_currentPLMN Struct Reference

### Data Fields

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

### 8.415.1 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 |

### 8.415.2 Field Documentation

#### 8.415.2.1 uint16_t nas_currentPLMN::MCC

#### 8.415.2.2 uint16_t nas_currentPLMN::MNC

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

#### 8.415.2.4 uint8_t nas_currentPLMN::netDescrLength

## 8.416 nas_dataSrvCapabilities Struct Reference

### Data Fields

- uint8_t dataCapabilitiesLen
- uint8_t dataCapabilities [32]

### 8.416.1 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+ |

### 8.416.2 Field Documentation

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

#### 8.416.2.2 uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen

## 8.417 nas_DataStatusDetail Struct Reference

## Data Fields

- uint32_t IPAddress
- uint8_t LastErrCode

### 8.417.1 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 |

### 8.417.2 Field Documentation

#### 8.417.2.1 uint32_t nas_DataStatusDetail::IPAddress

#### 8.417.2.2 uint8_t nas_DataStatusDetail::LastErrCode

## 8.418 nas_detailSvcInfo Struct Reference

### Data Fields

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

### 8.418.1 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 |

### 8.418.2 Field Documentation

#### 8.418.2.1 uint8_t nas_detailSvcInfo::hdrHybrid

#### 8.418.2.2 uint8_t nas_detailSvcInfo::hdrSrvStatus

#### 8.418.2.3 uint8_t nas_detailSvcInfo::isSysForbidden

#### 8.418.2.4 uint8_t nas_detailSvcInfo::srvCapability

#### 8.418.2.5 uint8_t nas_detailSvcInfo::srvStatus

## 8.419 nas_DeviceConfigDetail Struct Reference

### Data Fields

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

### 8.419.1 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 |

### 8.419.2 Field Documentation

#### 8.419.2.1 uint8_t nas_DeviceConfigDetail::Chipset

#### 8.419.2.2 uint8_t nas_DeviceConfigDetail::HWVersion

#### 8.419.2.3 uint8_t nas_DeviceConfigDetail::QLIC

#### 8.419.2.4 uint8_t nas_DeviceConfigDetail::Technology

## 8.420 nas_dirNum Struct Reference

### Data Fields

- uint8_t dirNumLen
- uint8_t dirNum [255]

### 8.420.1 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. |

### 8.420.2 Field Documentation

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

#### 8.420.2.2 uint8_t nas_dirNum::dirNumLen

## 8.421 nas_DRCParams Struct Reference

### Data Fields

- uint8_t DRCValue
- uint8_t DRCCover

#### 8.421.1 Detailed Description

This structure contains DRC information.

**Parameters**

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

#### 8.421.2 Field Documentation

##### 8.421.2.1 uint8_t nas_DRCParams::DRCCover

##### 8.421.2.2 uint8_t nas_DRCParams::DRCValue

## 8.422 nas_ecioListElement Struct Reference

### Data Fields

- int16_t ecio
- uint8_t radioIf

#### 8.422.1 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 |

#### 8.422.2 Field Documentation

#### 8.422.2.1 int16_t nas_ecioListElement::ecio

#### 8.422.2.2 uint8_t nas_ecioListElement::radioIf

## 8.423 nas_ECIOThresh Struct Reference

### Data Fields

- uint8_t ECIOThresListLen
- int16_t * pECIOThresList

### 8.423.1 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 |

### 8.423.2 Field Documentation

#### 8.423.2.1 uint8_t nas_ECIOThresh::ECIOThresListLen

#### 8.423.2.2 int16_t* nas_ECIOThresh::pECIOThresList

## 8.424 nas_errorRateListElement Struct Reference

### Data Fields

- uint16_t errorRate
- uint8_t radioIf

### 8.424.1 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 |

### 8.424.2 Field Documentation

#### 8.424.2.1 uint16_t nas_errorRateListElement::errorRate

#### 8.424.2.2 uint8_t nas_errorRateListElement::radioIf

## 8.425 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]

### 8.425.1 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. |

### 8.425.2 Field Documentation

#### 8.425.2.1 uint16_t nas_GERANInfo::arfcn

#### 8.425.2.2 uint8_t nas_GERANInfo::bsic

#### 8.425.2.3 uint32_t nas_GERANInfo::cellID

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

#### 8.425.2.5 uint16_t nas_GERANInfo::lac

#### 8.425.2.6 uint8_t nas_GERANInfo::nmrInst

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

#### 8.425.2.8 uint16_t nas_GERANInfo::rxLev

#### 8.425.2.9 uint32_t nas_GERANInfo::timingAdvance

## 8.426 nas_geranInstInfo Struct Reference

### Data Fields

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

### 8.426.1 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. |

### 8.426.2 Field Documentation

#### 8.426.2.1 uint16_t nas_geranInstInfo::geranArfcn

#### 8.426.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

#### 8.426.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

#### 8.426.2.4 int16_t nas_geranInstInfo::geranRssi

## 8.427 nas_gsmCellInfo Struct Reference

### Data Fields

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

### 8.427.1 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. |

### 8.427.2 Field Documentation

#### 8.427.2.1 uint16_t nas_gsmCellInfo::arfcn

#### 8.427.2.2 uint8_t nas_gsmCellInfo::band1900

#### 8.427.2.3 uint8_t nas_gsmCellInfo::bsicId

#### 8.427.2.4 uint8_t nas_gsmCellInfo::cellIdValid

#### 8.427.2.5 int16_t nas_gsmCellInfo::rssi

#### 8.427.2.6 int16_t nas_gsmCellInfo::srxlev

## 8.428 nas_GSMRSSIThresh Struct Reference

### Data Fields

- uint8_t GSMRSSIThreshListLen
- int16_t * pGSMRSSIThreshList

### 8.428.1 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) |

### 8.428.2 Field Documentation

#### 8.428.2.1 uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen

#### 8.428.2.2 int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList

## 8.429 nas_GSMSrvStatusInfo Struct Reference

### Data Fields

- uint8_t srvStatus
- uint8_t trueSrvStatus
- uint8_t isPrefDataPath

### 8.429.1 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 |

### 8.429.2 Field Documentation

#### 8.429.2.1 uint8_t nas_GSMSrvStatusInfo::isPrefDataPath

#### 8.429.2.2 uint8_t nas_GSMSrvStatusInfo::srvStatus

#### 8.429.2.3 uint8_t nas_GSMSrvStatusInfo::trueSrvStatus

## 8.430 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

### 8.430.1 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_PLMN_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. |
| *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 |

## 8.430.2 Field Documentation

### 8.430.2.1 uint32_t nas_GSMSysInfo::cellId

### 8.430.2.2 uint8_t nas_GSMSysInfo::cellIdValid

### 8.430.2.3 uint8_t nas_GSMSysInfo::dtmSupp

#### 8.430.2.4 uint8_t nas_GSMSysInfo::dtmSuppValid

#### 8.430.2.5 uint8_t nas_GSMSysInfo::egprsSupp

#### 8.430.2.6 uint8_t nas_GSMSysInfo::egprsSuppValid

#### 8.430.2.7 uint16_t nas_GSMSysInfo::lac

#### 8.430.2.8 uint8_t nas_GSMSysInfo::lacValid

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

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

#### 8.430.2.11 uint8_t nas_GSMSysInfo::networkIdValid

#### 8.430.2.12 uint8_t nas_GSMSysInfo::regRejectInfoValid

#### 8.430.2.13 uint8_t nas_GSMSysInfo::rejCause

#### 8.430.2.14 uint8_t nas_GSMSysInfo::rejectSrvDomain

#### 8.430.2.15 nas_sysInfoCommon nas_GSMSysInfo::sysInfoGSM

## 8.431 nas_HDRECIOThresh Struct Reference

### Data Fields

- uint8_t HDRECIOThreshListLen
- int16_t * pHDRECIOThreshList

### 8.431.1 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). |

### 8.431.2 Field Documentation

#### 8.431.2.1 uint8_t nas_HDRECIOThresh::HDRECIOThreshListLen

#### 8.431.2.2 int16_t* nas_HDRECIOThresh::pHDRECIOThreshList

## 8.432 nas_HDRIOThresh Struct Reference

### Data Fields

- uint8_t HDRIOThreshListLen
- int16_t * pHDRIOThreshList

### 8.432.1 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). |

### 8.432.2 Field Documentation

#### 8.432.2.1 uint8_t nas_HDRIOThresh::HDRIOThreshListLen

#### 8.432.2.2 int16_t* nas_HDRIOThresh::pHDRIOThreshList

## 8.433 nas_HDRPersonality_Ind_Data Struct Reference

### Data Fields

- uint16_t * pCurrentPersonality
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtocolSubtypeElement

### 8.433.1 Detailed Description

Elements for HDR Personality indication.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality* | • Current active personality index. |
| *pPersonalityList-Length* | • Number of Personality Protocol Subtype contains in this response. |
| *pProtocol-SubtypeElement* | • See nas_protocolSubtypeElement for more information. |

### 8.433.2 Field Documentation

#### 8.433.2.1 uint16_t* nas_HDRPersonality_Ind_Data::pCurrentPersonality

#### 8.433.2.2 uint8_t* nas_HDRPersonality_Ind_Data::pPersonalityListLength

#### 8.433.2.3 nas_protocolSubtypeElement* nas_HDRPersonality_Ind_Data::pProtocolSubtypeElement

## 8.434 nas_HDRRSSIThresh Struct Reference

### Data Fields

- uint8_t HDRRSSIThreshListLen
- int16_t * pHDRRSSIThreshList

### 8.434.1 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). |

### 8.434.2 Field Documentation

#### 8.434.2.1 uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen

#### 8.434.2.2 int16_t* nas_HDRRSSIThresh::pHDRRSSIThreshList

## 8.435 nas_HDRSINRThresh Struct Reference

### Data Fields

- uint8_t HDRSINRThresListLen
- uint8_t * pHDRSINRThresList

### 8.435.1 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 |

### 8.435.2 Field Documentation

#### 8.435.2.1 uint8_t nas_HDRSINRThresh::HDRSINRThresListLen

#### 8.435.2.2 uint8_t* nas_HDRSINRThresh::pHDRSINRThresList

## 8.436 nas_HDRSINRThreshold Struct Reference

### Data Fields

- uint8_t HDRSINRThreshListLen
- uint16_t * pHDRSINRThreshList

### 8.436.1 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 |

### 8.436.2 Field Documentation

#### 8.436.2.1 uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen

#### 8.436.2.2 uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList

## 8.437 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]

### 8.437.1 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. |

### 8.437.2 Field Documentation

#### 8.437.2.1 uint8_t nas_HDRSysInfo::hdrActiveProt

#### 8.437.2.2 uint8_t nas_HDRSysInfo::hdrActiveProtValid

#### 8.437.2.3 uint8_t nas_HDRSysInfo::hdrPersonality

#### 8.437.2.4 uint8_t nas_HDRSysInfo::hdrPersonalityValid

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

#### 8.437.2.6 uint8_t nas_HDRSysInfo::is856SysIdValid

#### 8.437.2.7 uint8_t nas_HDRSysInfo::isSysPrlMatch

#### 8.437.2.8 uint8_t nas_HDRSysInfo::isSysPrlMatchValid

#### 8.437.2.9 nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR

## 8.438 nas_homeSIDNID Struct Reference

### Data Fields

- uint8_t numInstances
- nas_sidNid SidNid [255]

### 8.438.1 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 |

### 8.438.2 Field Documentation

#### 8.438.2.1 uint8_t nas_homeSIDNID::numInstances

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

## 8.439 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]

### 8.439.1 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. |

### 8.439.2 Field Documentation

#### 8.439.2.1 uint8_t nas_infoInterFreq::cell_resel_priority

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

#### 8.439.2.3 uint8_t nas_infoInterFreq::cells_len

#### 8.439.2.4 uint16_t nas_infoInterFreq::earfcn

#### 8.439.2.5 uint8_t nas_infoInterFreq::threshXHigh

#### 8.439.2.6 uint8_t nas_infoInterFreq::threshXLow

## 8.440 nas_IOThresh Struct Reference

### Data Fields

- uint8_t IOThresListLen
- int32_t * pIOThresList

### 8.440.1 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 |

### 8.440.2 Field Documentation

#### 8.440.2.1 uint8_t nas_IOThresh::IOThresListLen

#### 8.440.2.2 int32_t* nas_IOThresh::pIOThresList

## 8.441 nas_lteEARFCN Struct Reference

### Data Fields

- uint8_t status
- uint32_t earfcn0
- uint32_t earfcn1

### 8.441.1 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. |

### 8.441.2 Field Documentation

#### 8.441.2.1 uint32_t nas_lteEARFCN::earfcn0

#### 8.441.2.2 uint32_t nas_lteEARFCN::earfcn1

#### 8.441.2.3 uint8_t nas_lteEARFCN::status

## 8.442 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]

### 8.442.1 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. |

### 8.442.2 Field Documentation

#### 8.442.2.1 uint8_t nas_lteGsmCellInfo::cellReselPriority

#### 8.442.2.2 uint8_t nas_lteGsmCellInfo::cells_len

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

#### 8.442.2.4 uint8_t nas_lteGsmCellInfo::nccPermitted

#### 8.442.2.5 uint8_t nas_lteGsmCellInfo::threshGsmHigh

#### 8.442.2.6 uint8_t nas_lteGsmCellInfo::threshGsmLow

## 8.443 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

### 8.443.1 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 |

## 8.443.2 Field Documentation

### 8.443.2.1 uint8_t nas_LTEInfo::band

#### 8.443.2.2 uint8_t nas_LTEInfo::bandwidth

#### 8.443.2.3 uint8_t nas_LTEInfo::emmConnState

#### 8.443.2.4 uint8_t nas_LTEInfo::emmState

#### 8.443.2.5 uint8_t nas_LTEInfo::emmSubState

#### 8.443.2.6 uint16_t nas_LTEInfo::RXChan

#### 8.443.2.7 uint16_t nas_LTEInfo::TXChan

## 8.444 nas_LTEInfoInterfreq Struct Reference

### Data Fields

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

### 8.444.1 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. |

### 8.444.2 Field Documentation

#### 8.444.2.1 uint8_t nas_LTEInfoInterfreq::freqsLen

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

#### 8.444.2.3 uint8_t nas_LTEInfoInterfreq::ueInIdle

## 8.445 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]

### 8.445.1 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 |

| | |
|---|---|
| *threshServing-Low* | • 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. |

### 8.445.2 Field Documentation

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

#### 8.445.2.2 uint8_t nas_LTEInfoIntrafreq::cellReselPriority

#### 8.445.2.3 uint8_t nas_LTEInfoIntrafreq::cellsLen

#### 8.445.2.4 uint16_t nas_LTEInfoIntrafreq::earfcn

#### 8.445.2.5 uint32_t nas_LTEInfoIntrafreq::globalCellId

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

#### 8.445.2.7 uint16_t nas_LTEInfoIntrafreq::servingCellId

#### 8.445.2.8 uint8_t nas_LTEInfoIntrafreq::sIntraSearch

#### 8.445.2.9 uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch

#### 8.445.2.10 uint16_t nas_LTEInfoIntrafreq::tac

#### 8.445.2.11 uint8_t nas_LTEInfoIntrafreq::threshServingLow

#### 8.445.2.12 uint8_t nas_LTEInfoIntrafreq::ueInIdle

## 8.446 nas_LTEInfoNeighboringGSM Struct Reference

### Data Fields

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

### 8.446.1 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. |

### 8.446.2 Field Documentation

#### 8.446.2.1 uint8_t nas_LTEInfoNeighboringGSM::freqsLen

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

#### 8.446.2.3 uint8_t nas_LTEInfoNeighboringGSM::ueInIdle

## 8.447 nas_LTEInfoNeighboringWCDMA Struct Reference

### Data Fields

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

### 8.447.1 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. |
| *LTEWCDMA-CellInfo[NAS_M-AX_DESCRIPTI-ON_LENGTH]* | • See nas_lteWcdmaCellInfo for more information. |

### 8.447.2 Field Documentation

#### 8.447.2.1 uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen

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

#### 8.447.2.3 uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle

## 8.448 nas_LTEOperationMode Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t * pLTEOperationMode

### 8.448.1 Detailed Description

This structure contains the nas_LTEOperationMode response parameters.

**Parameters**

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

### 8.448.2 Field Documentation

#### 8.448.2.1 uint32_t* nas_LTEOperationMode::pLTEOperationMode

#### 8.448.2.2 uint8_t nas_LTEOperationMode::TlvPresent

## 8.449 nas_ltePCI Struct Reference

### Data Fields

- uint8_t status
- uint32_t earfcn
- uint32_t pci

### 8.449.1 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 |

### 8.449.2 Field Documentation

#### 8.449.2.1 uint32_t nas_ltePCI::earfcn

#### 8.449.2.2 uint32_t nas_ltePCI::pci

#### 8.449.2.3 uint8_t nas_ltePCI::status

## 8.450 nas_lteRsrpinformation Struct Reference

### Data Fields

- int16_t rsrplevel

### 8.450.1 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). |

### 8.450.2 Field Documentation

#### 8.450.2.1 int16_t nas_lteRsrpinformation::rsrplevel

## 8.451 nas_LTERSRPThresh Struct Reference

### Data Fields

- uint8_t LTERSRPThreshListLen
- int16_t * pLTERSRPThreshList

### 8.451.1 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). |

### 8.451.2 Field Documentation

#### 8.451.2.1 uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen

#### 8.451.2.2 int16_t* nas_LTERSRPThresh::pLTERSRPThreshList

## 8.452 nas_LTERSRQThresh Struct Reference

### Data Fields

- uint8_t LTERSRQThreshListLen
- int16_t * pLTERSRQThreshList

### 8.452.1 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) |

### 8.452.2 Field Documentation

#### 8.452.2.1 uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen

#### 8.452.2.2 int16_t* nas_LTERSRQThresh::pLTERSRQThreshList

## 8.453 nas_LTERSSIThresh Struct Reference

### Data Fields

- uint8_t LTERSSIThreshListLen
- int16_t * pLTERSSIThreshList

### 8.453.1 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) |

### 8.453.2 Field Documentation

#### 8.453.2.1 uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen

#### 8.453.2.2 int16_t* nas_LTERSSIThresh::pLTERSSIThreshList

## 8.454 nas_LTESigRptCfg Struct Reference

### Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

### 8.454.1 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 |

### 8.454.2 Field Documentation

#### 8.454.2.1 uint8_t nas_LTESigRptCfg::avgPeriod

#### 8.454.2.2 uint8_t nas_LTESigRptCfg::rptRate

## 8.455 nas_LTESigRptConfig Struct Reference

### Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

### 8.455.1 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 |

### 8.455.2 Field Documentation

#### 8.455.2.1 uint8_t nas_LTESigRptConfig::avgPeriod

#### 8.455.2.2 uint8_t nas_LTESigRptConfig::rptRate

## 8.456 nas_lteSnrinformation Struct Reference

### Data Fields

- int16_t snrlevel

### 8.456.1 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. |

### 8.456.2 Field Documentation

#### 8.456.2.1 int16_t nas_lteSnrinformation::snrlevel

## 8.457 nas_LTESNRThresh Struct Reference

### Data Fields

- uint8_t LTESNRThresListLen
- int16_t * pLTESNRThresList

### 8.457.1 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 |

### 8.457.2 Field Documentation

#### 8.457.2.1 uint8_t nas_LTESNRThresh::LTESNRThresListLen

#### 8.457.2.2 int16_t* nas_LTESNRThresh::pLTESNRThresList

## 8.458 nas_LTESNRThreshold Struct Reference

### Data Fields

- uint8_t LTESNRThreshListLen
- int16_t * pLTESNRThreshList

### 8.458.1 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). |

### 8.458.2 Field Documentation

#### 8.458.2.1 uint8_t nas_LTESNRThreshold::LTESNRThreshListLen

#### 8.458.2.2 int16_t* nas_LTESNRThreshold::pLTESNRThreshList

## 8.459 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

### 8.459.1 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_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. |
| *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 |

### 8.459.2 Field Documentation

#### 8.459.2.1 uint32_t nas_LTESysInfo::cellId

#### 8.459.2.2 uint8_t nas_LTESysInfo::cellIdValid

#### 8.459.2.3 uint16_t nas_LTESysInfo::lac

#### 8.459.2.4 uint8_t nas_LTESysInfo::lacValid

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

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

#### 8.459.2.7 uint8_t nas_LTESysInfo::networkIdValid

#### 8.459.2.8 uint8_t nas_LTESysInfo::regRejectInfoValid

#### 8.459.2.9 uint8_t nas_LTESysInfo::rejCause

#### 8.459.2.10 uint8_t nas_LTESysInfo::rejectSrvDomain

#### 8.459.2.11 nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE

#### 8.459.2.12 uint16_t nas_LTESysInfo::tac

#### 8.459.2.13 uint8_t nas_LTESysInfo::tacValid

## 8.460 nas_lteWcdmaCellInfo Struct Reference

### Data Fields

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

### 8.460.1 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. |
| *WCDMACell-Info[NAS_MAX_-DESCRIPTION-_LENGTH]* | • See nas_wcdmaCellInfo for more information. |

### 8.460.2 Field Documentation

#### 8.460.2.1 uint8_t nas_lteWcdmaCellInfo::cellReselPriority

#### 8.460.2.2 uint8_t nas_lteWcdmaCellInfo::cellsLen

#### 8.460.2.3 uint16_t nas_lteWcdmaCellInfo::threshXhigh

#### 8.460.2.4 uint16_t nas_lteWcdmaCellInfo::threshXlow

#### 8.460.2.5 uint16_t nas_lteWcdmaCellInfo::uarfcn

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

## 8.461 nas_minBasedIMSI Struct Reference

### Data Fields

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

### 8.461.1 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 |

### 8.461.2 Field Documentation

#### 8.461.2.1 uint16_t nas_minBasedIMSI::imsiM1112

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

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

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

## 8.462 nas_MNRInfo Struct Reference

### Data Fields

- uint16_t mcc
- uint16_t mnc
- uint32_t rat

### 8.462.1 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 |

### 8.462.2 Field Documentation

#### 8.462.2.1 uint16_t nas_MNRInfo::mcc

#### 8.462.2.2 uint16_t nas_MNRInfo::mnc

#### 8.462.2.3 uint32_t nas_MNRInfo::rat

## 8.463 nas_namName Struct Reference

### Data Fields

- uint8_t namNameLen
- uint8_t namName [12]

### 8.463.1 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. |

### 8.463.2 Field Documentation

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

#### 8.463.2.2 uint8_t nas_namName::namNameLen

## 8.464 nas_netSelectionPref Struct Reference

### Data Fields

- uint8_t netReg
- uint16_t mcc
- uint16_t mnc

### 8.464.1 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 |

### 8.464.2 Field Documentation

#### 8.464.2.1 uint16_t nas_netSelectionPref::mcc

#### 8.464.2.2 uint16_t nas_netSelectionPref::mnc

#### 8.464.2.3 uint8_t nas_netSelectionPref::netReg

## 8.465 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

### 8.465.1 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 |

### 8.465.2 Field Documentation

#### 8.465.2.1 uint8_t nas_NetworkStat1x::ActSetCnt

#### 8.465.2.2 uint8_t nas_NetworkStat1x::NeighborSetCnt

#### 8.465.2.3 nas_ActPilotPNElement* nas_NetworkStat1x::pActPilotPNElements

#### 8.465.2.4 uint16_t* nas_NetworkStat1x::pNeighborSetPilotPN

#### 8.465.2.5 uint16_t nas_NetworkStat1x::RX_EC_IO

#### 8.465.2.6 uint32_t nas_NetworkStat1x::RX_PWR

#### 8.465.2.7 uint16_t nas_NetworkStat1x::SO

#### 8.465.2.8 uint8_t nas_NetworkStat1x::State

#### 8.465.2.9 uint32_t nas_NetworkStat1x::TX_PWR

## 8.466 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

### 8.466.1 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) |

### 8.466.2 Field Documentation

#### 8.466.2.1 uint8_t nas_NetworkStatEVDO::MACIndex

#### 8.466.2.2 uint16_t nas_NetworkStatEVDO::PER

#### 8.466.2.3 uint16_t nas_NetworkStatEVDO::PilotEnergy

#### 8.466.2.4 uint16_t* nas_NetworkStatEVDO::pSectorID

#### 8.466.2.5 uint16_t nas_NetworkStatEVDO::RX_PWR

#### 8.466.2.6 uint8_t nas_NetworkStatEVDO::SectorIDLen

#### 8.466.2.7 uint8_t nas_NetworkStatEVDO::SNR

#### 8.466.2.8 uint8_t nas_NetworkStatEVDO::State

## 8.467 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

### 8.467.1 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 |

### 8.467.2 Field Documentation

#### 8.467.2.1 uint16_t nas_nmrCellInfo::nmrArfcn

#### 8.467.2.2 uint8_t nas_nmrCellInfo::nmrBsic

#### 8.467.2.3 uint32_t nas_nmrCellInfo::nmrCellID

#### 8.467.2.4 uint16_t nas_nmrCellInfo::nmrLac

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

#### 8.467.2.6 uint16_t nas_nmrCellInfo::nmrRxLev

## 8.468 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

### 8.468.1 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. |

### 8.468.2 Field Documentation

#### 8.468.2.1 uint64_t nas_nr5gBandPref::bits_129_192

#### 8.468.2.2 uint64_t nas_nr5gBandPref::bits_193_256

#### 8.468.2.3 uint64_t nas_nr5gBandPref::bits_1_64

#### 8.468.2.4 uint64_t nas_nr5gBandPref::bits_65_128

## 8.469 nas_operatorNameString Struct Reference

### Data Fields

- uint8_t PLMNName [255]

### 8.469.1 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. |

### 8.469.2 Field Documentation

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

## 8.470 nas_OperatorPLMNData Struct Reference

### Data Fields

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

### 8.470.1 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. |

### 8.470.2 Field Documentation

#### 8.470.2.1 uint16_t nas_OperatorPLMNData::lac1

#### 8.470.2.2 uint16_t nas_OperatorPLMNData::lac2

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

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

#### 8.470.2.5 uint8_t nas_OperatorPLMNData::PLMNRecID

## 8.471 nas_operatorPLMNList Struct Reference

### Data Fields

- uint16_t numInstance
- nas_OperatorPLMNData PLMNData [255]

### 8.471.1 Detailed Description

This structure contains Operator PLMN List.

**Parameters**

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

### 8.471.2 Field Documentation

#### 8.471.2.1 uint16_t nas_operatorPLMNList::numInstance

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

## 8.472 nas_PhyCaAggPcellInfo Struct Reference

### Data Fields

- uint16_t pci
- uint16_t freq
- NAS_LTE_CPHY_CA_BW_NRB_LITE dl_bw_value
- uint16_t iLTEbandValue
- uint8_t TlvPresent

### 8.472.1 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. |

### 8.472.2 Field Documentation

#### 8.472.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggPcellInfo::dl_bw_value

#### 8.472.2.2 uint16_t nas_PhyCaAggPcellInfo::freq

#### 8.472.2.3 uint16_t nas_PhyCaAggPcellInfo::iLTEbandValue

#### 8.472.2.4 uint16_t nas_PhyCaAggPcellInfo::pci

#### 8.472.2.5 uint8_t nas_PhyCaAggPcellInfo::TlvPresent

## 8.473 nas_PhyCaAggScellDlBw Struct Reference

### Data Fields

- NAS_LTE_CPHY_CA_BW_NRB_LITE dl_bw_value
- uint8_t TlvPresent

### 8.473.1 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. |

### 8.473.2 Field Documentation

#### 8.473.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellDlBw::dl_bw_value

#### 8.473.2.2 uint8_t nas_PhyCaAggScellDlBw::TlvPresent

## 8.474 nas_PhyCaAggScellIndex Struct Reference

### Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

### 8.474.1 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. |

### 8.474.2 Field Documentation

#### 8.474.2.1 uint8_t nas_PhyCaAggScellIndex::scell_idx

#### 8.474.2.2 uint8_t nas_PhyCaAggScellIndex::TlvPresent

## 8.475 nas_PhyCaAggScellIndType Struct Reference

### Data Fields

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

### 8.475.1 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. |

### 8.475.2 Field Documentation

#### 8.475.2.1 uint16_t nas_PhyCaAggScellIndType::freq

#### 8.475.2.2 uint16_t nas_PhyCaAggScellIndType::pci

#### 8.475.2.3 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellIndType::scell_state

#### 8.475.2.4 uint8_t nas_PhyCaAggScellIndType::TlvPresent

## 8.476 nas_PhyCaAggScellInfo Struct Reference

### Data Fields

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

### 8.476.1 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. |

### 8.476.2 Field Documentation

#### 8.476.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellInfo::dl_bw_value

#### 8.476.2.2 uint16_t nas_PhyCaAggScellInfo::freq

#### 8.476.2.3 uint16_t nas_PhyCaAggScellInfo::iLTEbandValue

#### 8.476.2.4 uint16_t nas_PhyCaAggScellInfo::pci

#### 8.476.2.5 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellInfo::scell_state

#### 8.476.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

## 8.477 nas_PilotSetData Struct Reference

### Data Fields

- uint8_t NumPilots
- nas_PilotSetParams * pPilotSetInfo

### 8.477.1 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.

### 8.477.2 Field Documentation

#### 8.477.2.1 uint8_t nas_PilotSetData::NumPilots

#### 8.477.2.2 nas_PilotSetParams* nas_PilotSetData::pPilotSetInfo

## 8.478 nas_PilotSetParams Struct Reference

### Data Fields

- uint32_t PilotType
- uint16_t PilotPN
- uint16_t PilotStrength

### 8.478.1 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) |

### 8.478.2 Field Documentation

#### 8.478.2.1 uint16_t nas_PilotSetParams::PilotPN

#### 8.478.2.2 uint16_t nas_PilotSetParams::PilotStrength

#### 8.478.2.3 uint32_t nas_PilotSetParams::PilotType

## 8.479 nas_PLMNNetworkName Struct Reference

### Data Fields

- uint8_t numInstance
- nas_PLMNNetworkNameData PLMNNetName [255]

### 8.479.1 Detailed Description

This structure contains NITZ Information list.

**Parameters**

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

### 8.479.2 Field Documentation

#### 8.479.2.1 uint8_t nas_PLMNNetworkName::numInstance

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

## 8.480 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]

### 8.480.1 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. |

### 8.480.2 Field Documentation

#### 8.480.2.1 uint8_t nas_PLMNNetworkNameData::codingScheme

#### 8.480.2.2 uint8_t nas_PLMNNetworkNameData::countryInitials

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

#### 8.480.2.4 uint8_t nas_PLMNNetworkNameData::longNameLen

#### 8.480.2.5 uint8_t nas_PLMNNetworkNameData::longNameSpareBits

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

#### 8.480.2.7 uint8_t nas_PLMNNetworkNameData::shortNameLen

#### 8.480.2.8 uint8_t nas_PLMNNetworkNameData::shortNameSpareBits

## 8.481 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

### 8.481.1 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 |

### 8.481.2 Field Documentation

#### 8.481.2.1 uint16_t nas_protocolSubtypeElement::AccessMac

#### 8.481.2.2 uint16_t nas_protocolSubtypeElement::AuthProt

#### 8.481.2.3 uint16_t nas_protocolSubtypeElement::ControlMac

#### 8.481.2.4 uint16_t nas_protocolSubtypeElement::EncryptProt

#### 8.481.2.5 uint16_t nas_protocolSubtypeElement::ForwardMac

#### 8.481.2.6 uint16_t nas_protocolSubtypeElement::IdleState

#### 8.481.2.7 uint16_t nas_protocolSubtypeElement::KeyExchange

#### 8.481.2.8 uint16_t nas_protocolSubtypeElement::MultDisc

#### 8.481.2.9 uint16_t nas_protocolSubtypeElement::PhysicalLayer

#### 8.481.2.10 uint16_t nas_protocolSubtypeElement::ReverseMac

#### 8.481.2.11 uint16_t nas_protocolSubtypeElement::SecProt

#### 8.481.2.12 uint16_t nas_protocolSubtypeElement::VirtStream

## 8.482 nas_qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- uint8_t leapSeconds
- uint8_t localTimeOffset
- uint8_t daylightSavings

### 8.482.1 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) |

### 8.482.2 Field Documentation

#### 8.482.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

#### 8.482.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

#### 8.482.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

## 8.483 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]

### 8.483.1 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. |

### 8.483.2 Field Documentation

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

#### 8.483.2.2 uint32_t nas_QmiNas3GppNetworkInfo::Forbidden

#### 8.483.2.3 uint32_t nas_QmiNas3GppNetworkInfo::InUse

#### 8.483.2.4 uint16_t nas_QmiNas3GppNetworkInfo::MCC

#### 8.483.2.5 uint16_t nas_QmiNas3GppNetworkInfo::MNC

#### 8.483.2.6 uint32_t nas_QmiNas3GppNetworkInfo::Preferred

#### 8.483.2.7 uint32_t nas_QmiNas3GppNetworkInfo::Roaming

## 8.484 nas_QmiNas3GppNetworkRAT Struct Reference

### Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t RAT

### 8.484.1 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 |

### 8.484.2 Field Documentation

#### 8.484.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

#### 8.484.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

#### 8.484.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

## 8.485 nas_QmisNasPcsDigit Struct Reference

### Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t includes_pcs_digit

### 8.485.1 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 |

### 8.485.2 Field Documentation

#### 8.485.2.1 uint8_t nas_QmisNasPcsDigit::includes_pcs_digit

#### 8.485.2.2 uint16_t nas_QmisNasPcsDigit::MCC

#### 8.485.2.3 uint16_t nas_QmisNasPcsDigit::MNC

## 8.486 nas_RankIndicatorTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t count0
- uint32_t count1

### 8.486.1 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 |

### 8.486.2 Field Documentation

#### 8.486.2.1 uint32_t nas_RankIndicatorTlv::count0

#### 8.486.2.2 uint32_t nas_RankIndicatorTlv::count1

#### 8.486.2.3 uint8_t nas_RankIndicatorTlv::TlvPresent

## 8.487 nas_RejectReasonTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t serviceDomain
- uint32_t rejectCause

### 8.487.1 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 |

### 8.487.2 Field Documentation

#### 8.487.2.1 uint32_t nas_RejectReasonTlv::rejectCause

#### 8.487.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

#### 8.487.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

## 8.488 nas_RFBandInfoElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint16_t activeBandClass
- uint16_t activeChannel

### 8.488.1 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) |

### 8.488.2 Field Documentation

#### 8.488.2.1 uint16_t nas_RFBandInfoElements::activeBandClass

#### 8.488.2.2 uint16_t nas_RFBandInfoElements::activeChannel

#### 8.488.2.3 uint8_t nas_RFBandInfoElements::radioInterface

## 8.489 nas_RfBandInfoExtFormat Struct Reference

### Data Fields

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

### 8.489.1 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 |

### 8.489.2 Field Documentation

#### 8.489.2.1 uint8_t* nas_RfBandInfoExtFormat::pInstancesSize

#### 8.489.2.2 struct nas_RfBandInfoExtFormatElements* nas_RfBandInfoExtFormat::pRfBandInfoExtFormatParam

#### 8.489.2.3 uint8_t nas_RfBandInfoExtFormat::TlvPresent

## 8.490 nas_RfBandInfoExtFormatElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint16_t activeBand
- uint32_t activeChannel

### 8.490.1 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) |

### 8.490.2 Field Documentation

#### 8.490.2.1 uint16_t nas_RfBandInfoExtFormatElements::activeBand

#### 8.490.2.2 uint32_t nas_RfBandInfoExtFormatElements::activeChannel

#### 8.490.2.3 uint8_t nas_RfBandInfoExtFormatElements::radioInterface

## 8.491 nas_RfBandInfoList Struct Reference

### Data Fields

- uint8_t * pInstanceSize
- struct nas_RFBandInfoElements * pRFBandInfoParam

### 8.491.1 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 |

### 8.491.2 Field Documentation

#### 8.491.2.1 uint8_t* nas_RfBandInfoList::pInstanceSize

#### 8.491.2.2 struct nas_RFBandInfoElements* nas_RfBandInfoList::pRFBandInfoParam

## 8.492 nas_RfBandwidthInfo Struct Reference

### Data Fields

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

### 8.492.1 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. |
| *pRfBandwidthInfoParam* | • See nas_RfBandwidthInfoElements for more information |

### 8.492.2 Field Documentation

#### 8.492.2.1 uint8_t* nas_RfBandwidthInfo::pInstancesSize

#### 8.492.2.2 struct nas_RfBandwidthInfoElements* nas_RfBandwidthInfo::pRfBandwidthInfoParam

#### 8.492.2.3 uint8_t nas_RfBandwidthInfo::TlvPresent

## 8.493 nas_RfBandwidthInfoElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint32_t bandwidth

### 8.493.1 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 |

### 8.493.2 Field Documentation

#### 8.493.2.1 uint32_t nas_RfBandwidthInfoElements::bandwidth

#### 8.493.2.2 uint8_t nas_RfBandwidthInfoElements::radioInterface

## 8.494 nas_RfDedicatedBandInfo Struct Reference

### Data Fields

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

### 8.494.1 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 |

### 8.494.2 Field Documentation

#### 8.494.2.1 uint8_t* nas_RfDedicatedBandInfo::pInstancesSize

#### 8.494.2.2 struct nas_RfDedicatedBandInfoElements* nas_RfDedicatedBandInfo::pRfDedicatedBandInfoParam

#### 8.494.2.3 uint8_t nas_RfDedicatedBandInfo::TlvPresent

## 8.495 nas_RfDedicatedBandInfoElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint16_t dedicatedBand

### 8.495.1 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 |

### 8.495.2 Field Documentation

#### 8.495.2.1 uint16_t nas_RfDedicatedBandInfoElements::dedicatedBand

#### 8.495.2.2 uint8_t nas_RfDedicatedBandInfoElements::radioInterface

## 8.496 nas_RFInfoTlv Struct Reference

### Data Fields

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

## 8.496.1 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. |

## 8.496.2 Field Documentation

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

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

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

### 8.496.2.4 uint8_t nas_RFInfoTlv::radioInterfaceSize

### 8.496.2.5 uint8_t nas_RFInfoTlv::TlvPresent

## 8.497 nas_roamIndList Struct Reference

### Data Fields

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

### 8.497.1 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 |

### 8.497.2 Field Documentation

#### 8.497.2.1 uint8_t nas_roamIndList::numInstances

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

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

## 8.498 nas_RSRPThresh Struct Reference

### Data Fields

- uint8_t RSRPThresListLen
- int16_t * pRSRPThresList

### 8.498.1 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 |

### 8.498.2 Field Documentation

#### 8.498.2.1 int16_t* nas_RSRPThresh::pRSRPThresList

#### 8.498.2.2 uint8_t nas_RSRPThresh::RSRPThresListLen

## 8.499 nas_rsrqInformation Struct Reference

### Data Fields

- int8_t rsrq
- uint8_t radioIf

### 8.499.1 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 |

### 8.499.2 Field Documentation

#### 8.499.2.1 uint8_t nas_rsrqInformation::radioIf

#### 8.499.2.2 int8_t nas_rsrqInformation::rsrq

## 8.500 nas_RSRQThresh Struct Reference

### Data Fields

- uint8_t RSRQThresListLen
- int8_t * pRSRQThresList

### 8.500.1 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 |

### 8.500.2 Field Documentation

#### 8.500.2.1 int8_t* nas_RSRQThresh::pRSRQThresList

#### 8.500.2.2 uint8_t nas_RSRQThresh::RSRQThresListLen

## 8.501 nas_RSSIThresh Struct Reference

### Data Fields

- uint8_t RSSIThresListLen
- int8_t * pRSSIThresList

### 8.501.1 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. |

### 8.501.2 Field Documentation

#### 8.501.2.1 int8_t* nas_RSSIThresh::pRSSIThresList

#### 8.501.2.2 uint8_t nas_RSSIThresh::RSSIThresListLen

## 8.502 nas_rxInfo Struct Reference

### Data Fields

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

### 8.502.1 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. |

### 8.502.2 Field Documentation

#### 8.502.2.1 int32_t nas_rxInfo::ecio

#### 8.502.2.2 uint8_t nas_rxInfo::isRadioTuned

#### 8.502.2.3 uint32_t nas_rxInfo::phase

#### 8.502.2.4 int32_t nas_rxInfo::rscp

#### 8.502.2.5 int32_t nas_rxInfo::rsrp

#### 8.502.2.6 int32_t nas_rxInfo::rxPower

## 8.503 nas_RxSigInfo Struct Reference

### Data Fields

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

### 8.503.1 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 |

### 8.503.2 Field Documentation

#### 8.503.2.1 uint8_t nas_RxSigInfo::isRadioTuned

#### 8.503.2.2 int32_t nas_RxSigInfo::rsrp

#### 8.503.2.3 uint8_t nas_RxSigInfo::rxChainIndex

#### 8.503.2.4 int32_t nas_RxSigInfo::rxPower

## 8.504 nas_rxSignalStrengthListElement Struct Reference

### Data Fields

- int16_t rxSignalStrength
- uint8_t radioIf

### 8.504.1 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.

### 8.504.2 Field Documentation

#### 8.504.2.1 uint8_t nas_rxSignalStrengthListElement::radioIf

#### 8.504.2.2 int16_t nas_rxSignalStrengthListElement::rxSignalStrength

## 8.505 nas_SccRxInfo Struct Reference

### Data Fields

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

### 8.505.1 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. |

### 8.505.2 Field Documentation

#### 8.505.2.1 uint8_t nas_SccRxInfo::numInstances

#### 8.505.2.2 int32_t nas_SccRxInfo::rsrq

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

#### 8.505.2.4 int16_t nas_SccRxInfo::snr

#### 8.505.2.5 uint8_t nas_SccRxInfo::TlvPresent

## 8.506 nas_serviceProviderName Struct Reference

### Data Fields

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

### 8.506.1 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. |

### 8.506.2 Field Documentation

#### 8.506.2.1 uint8_t nas_serviceProviderName::displayCondition

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

#### 8.506.2.3 uint8_t nas_serviceProviderName::spnLength

## 8.507 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]

### 8.507.1 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 |

### 8.507.2 Field Documentation

#### 8.507.2.1 uint8_t nas_servSystem::csAttachState

#### 8.507.2.2 uint8_t nas_servSystem::numRadioInterfaces

#### 8.507.2.3 uint8_t nas_servSystem::psAttachState

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

#### 8.507.2.5 uint8_t nas_servSystem::regState

#### 8.507.2.6 uint8_t nas_servSystem::selNetwork

## 8.508 nas_sidNid Struct Reference

### Data Fields

- uint16_t nid
- uint16_t sid

### 8.508.1 Detailed Description

This structure contains the parameters for SidNid Information

**Parameters**

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

### 8.508.2 Field Documentation

#### 8.508.2.1 uint16_t nas_sidNid::nid

#### 8.508.2.2 uint16_t nas_sidNid::sid

## 8.509 nas_SignalStrengthTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- int8_t signalStrength
- uint32_t radioInterface

### 8.509.1 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 |

### 8.509.2 Field Documentation

#### 8.509.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

#### 8.509.2.2 int8_t nas_SignalStrengthTlv::signalStrength

#### 8.509.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

## 8.510 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

### 8.510.1 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. |

### 8.510.2 Field Documentation

#### 8.510.2.1 uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta

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

#### 8.510.2.3 uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen

#### 8.510.2.4 uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta

#### 8.510.2.5 uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta

#### 8.510.2.6 uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta

#### 8.510.2.7 uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta

#### 8.510.2.8 uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

#### 8.510.2.9 uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta

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

#### 8.510.2.11 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen

## 8.511 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

### 8.511.1 Detailed Description

This structure contains paramaters signal strengths information.

**Parameters**

| | |
|---|---|
| *rxSignal-StrengthInfo* | signal strength info list |
| *ecioInfo* | ecio info list |
| *io* | received IO in dBm; IO is only applicable for 1xEV-DO |
| *sinr* | SINR level |
| *errorRateInfo* | error rate info |
| *rsrqInfo* | rsrq info |
| *lteSnrinfo* | lte Snr information |
| *lteRsrpinfo* | lte rsrp info |

### 8.511.2 Field Documentation

#### 8.511.2.1 nas_ecioListElement nas_SLQSSignalStrengthsInformation::ecioInfo

#### 8.511.2.2 nas_errorRateListElement nas_SLQSSignalStrengthsInformation::errorRateInfo

#### 8.511.2.3 uint32_t nas_SLQSSignalStrengthsInformation::io

#### 8.511.2.4 nas_lteRsrpinformation nas_SLQSSignalStrengthsInformation::lteRsrpinfo

#### 8.511.2.5 nas_lteSnrinformation nas_SLQSSignalStrengthsInformation::lteSnrinfo

#### 8.511.2.6 nas_rsrqInformation nas_SLQSSignalStrengthsInformation::rsrqInfo

#### 8.511.2.7 nas_rxSignalStrengthListElement nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo

#### 8.511.2.8 uint8_t nas_SLQSSignalStrengthsInformation::sinr

## 8.512 nas_SLQSSignalStrengthsTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- nas_SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo

### 8.512.1 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 |

### 8.512.2 Field Documentation

#### 8.512.2.1 nas_SLQSSignalStrengthsInformation nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo

#### 8.512.2.2 uint8_t nas_SLQSSignalStrengthsTlv::TlvPresent

## 8.513 nas_SrvStatusInfo Struct Reference

### Data Fields

- uint8_t srvStatus
- uint8_t isPrefDataPath

### 8.513.1 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 |

### 8.513.2 Field Documentation

#### 8.513.2.1 uint8_t nas_SrvStatusInfo::isPrefDataPath

#### 8.513.2.2 uint8_t nas_SrvStatusInfo::srvStatus

## 8.514 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

### 8.514.1 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 |

### 8.514.2 Field Documentation

#### 8.514.2.1 uint8_t nas_sysInfoCommon::isSysForbidden

#### 8.514.2.2 uint8_t nas_sysInfoCommon::isSysForbiddenValid

#### 8.514.2.3 uint8_t nas_sysInfoCommon::roamStatus

#### 8.514.2.4 uint8_t nas_sysInfoCommon::roamStatusValid

#### 8.514.2.5 uint8_t nas_sysInfoCommon::srvCapability

#### 8.514.2.6 uint8_t nas_sysInfoCommon::srvCapabilityValid

#### 8.514.2.7 uint8_t nas_sysInfoCommon::srvDomain

#### 8.514.2.8 uint8_t nas_sysInfoCommon::srvDomainValid

## 8.515 nas_TDSCDMAECIOThresh Struct Reference

### Data Fields

- uint8_t TDSCDMAECIOThreshListLen
- float * pTDSCDMAECIOThreshList

### 8.515.1 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. |

### 8.515.2 Field Documentation

#### 8.515.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

#### 8.515.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

## 8.516 nas_TDSCDMARSCPThresh Struct Reference

### Data Fields

- uint8_t TDSCDMARSCPThreshListLen
- int16_t * pTDSCDMARSCPThreshList

### 8.516.1 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). |

### 8.516.2 Field Documentation

#### 8.516.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

#### 8.516.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

## 8.517 nas_TDSCDMARSSIThresh Struct Reference

### Data Fields

- uint8_t TDSCDMARSSIThreshListLen
- float * pTDSCDMARSSIThreshList

### 8.517.1 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. |

### 8.517.2 Field Documentation

#### 8.517.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

#### 8.517.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

## 8.518 nas_TDSCDMASINRCONFThresh Struct Reference

### Data Fields

- uint8_t TDSCDMASINRCONFThreshListLen
- float * pTDSCDMASINRCONFThreshList

### 8.518.1 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 |

### 8.518.2 Field Documentation

#### 8.518.2.1 float* nas_TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

#### 8.518.2.2 uint8_t nas_TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

## 8.519 nas_TDSCDMASINRThresh Struct Reference

### Data Fields

- uint8_t TDSCDMASINRThreshListLen
- float * pTDSCDMASINRThreshList

### 8.519.1 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 |

### 8.519.2 Field Documentation

#### 8.519.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

#### 8.519.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.520 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

### 8.520.1 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. |

### 8.520.2 Field Documentation

#### 8.520.2.1 uint8_t nas_timeInfo::day

#### 8.520.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

#### 8.520.2.3 uint8_t nas_timeInfo::dayOfWeek

#### 8.520.2.4 uint8_t nas_timeInfo::hour

#### 8.520.2.5 uint8_t nas_timeInfo::minute

#### 8.520.2.6 uint8_t nas_timeInfo::month

#### 8.520.2.7 uint8_t nas_timeInfo::radioInterface

#### 8.520.2.8 uint8_t nas_timeInfo::second

#### 8.520.2.9 int8_t nas_timeInfo::timeZone

#### 8.520.2.10 uint8_t nas_timeInfo::TlvPresent

#### 8.520.2.11 uint16_t nas_timeInfo::year

## 8.521 nas_trueIMSI Struct Reference

### Data Fields

- uint8_t mccT [3]
- uint16_t imsiT1112
- uint8_t imsiTS1 [7]
- uint8_t imsiTS2 [3]
- uint8_t imsiTaddrNum

### 8.521.1 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 |

### 8.521.2 Field Documentation

#### 8.521.2.1 uint16_t nas_truelMSI::imsiT1112

#### 8.521.2.2 uint8_t nas_truelMSI::imsiTaddrNum

#### 8.521.2.3 uint8_t nas_truelMSI::imsiTS1[7]

#### 8.521.2.4 uint8_t nas_truelMSI::imsiTS2[3]

#### 8.521.2.5 uint8_t nas_truelMSI::mccT[3]

## 8.522 nas_txInfo Struct Reference

### Data Fields

- uint8_t isInTraffic
- int32_t txPower

### 8.522.1 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. |

### 8.522.2 Field Documentation

#### 8.522.2.1 uint8_t nas_txInfo::isInTraffic

#### 8.522.2.2 int32_t nas_txInfo::txPower

## 8.523 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]

### 8.523.1 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. |

### 8.523.2 Field Documentation

#### 8.523.2.1 uint16_t nas_UMTSInfo::cellID

#### 8.523.2.2 int16_t nas_UMTSInfo::ecio

#### 8.523.2.3 uint8_t nas_UMTSInfo::geranInst

#### 8.523.2.4 nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]

#### 8.523.2.5 uint16_t nas_UMTSInfo::lac

#### 8.523.2.6 uint8_t nas_UMTSInfo::plmn[3]

#### 8.523.2.7 uint16_t nas_UMTSInfo::psc

#### 8.523.2.8 int16_t nas_UMTSInfo::rscp

#### 8.523.2.9 uint16_t nas_UMTSInfo::uarfcn

#### 8.523.2.10 uint8_t nas_UMTSInfo::umtsInst

#### 8.523.2.11 nas_UMTSinstInfo nas_UMTSInfo::UMTSInstInfo[255]

## 8.524 nas_UMTSinstInfo Struct Reference

### Data Fields

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

### 8.524.1 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). |

### 8.524.2 Field Documentation

#### 8.524.2.1 int16_t nas_UMTSinstInfo::umtsEcio

#### 8.524.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

#### 8.524.2.3 int16_t nas_UMTSinstInfo::umtsRscp

#### 8.524.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

## 8.525 nas_umtsLTENbrCell Struct Reference

### Data Fields

- uint16_t earfcn
- uint16_t pci
- uint32_t rsrp
- uint32_t rsrq
- int16_t srxlev
- uint8_t cellIsTDD

### 8.525.1 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. |

### 8.525.2 Field Documentation

#### 8.525.2.1 uint8_t nas_umtsLTENbrCell::cellIsTDD

#### 8.525.2.2 uint16_t nas_umtsLTENbrCell::earfcn

#### 8.525.2.3 uint16_t nas_umtsLTENbrCell::pci

#### 8.525.2.4 uint32_t nas_umtsLTENbrCell::rsrp

#### 8.525.2.5 uint32_t nas_umtsLTENbrCell::rsrq

#### 8.525.2.6 int16_t nas_umtsLTENbrCell::srxlev

## 8.526 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

### 8.526.1 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. |

### 8.526.2 Field Documentation

#### 8.526.2.1 uint8_t nas_UniversalTime::day

#### 8.526.2.2 uint8_t nas_UniversalTime::dayOfWeek

#### 8.526.2.3 uint8_t nas_UniversalTime::hour

#### 8.526.2.4 uint8_t nas_UniversalTime::minute

#### 8.526.2.5 uint8_t nas_UniversalTime::month

#### 8.526.2.6 uint8_t nas_UniversalTime::second

#### 8.526.2.7 uint16_t nas_UniversalTime::year

## 8.527 nas_wcdmaCellInfo Struct Reference

### Data Fields

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

### 8.527.1 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. |

### 8.527.2 Field Documentation

#### 8.527.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

#### 8.527.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

#### 8.527.2.3 uint16_t nas_wcdmaCellInfo::psc

#### 8.527.2.4 int16_t nas_wcdmaCellInfo::srxlev

## 8.528 nas_WCDMAECIOThresh Struct Reference

### Data Fields

- uint8_t WCDMAECIOThreshListLen
- int16_t * pWCDMAECIOThreshList

### 8.528.1 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) |

### 8.528.2 Field Documentation

#### 8.528.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

#### 8.528.2.2 uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen

## 8.529 nas_WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

- uint32_t wcdmaRRCState
- uint8_t umtsLTENbrCellLen
- nas_umtsLTENbrCell UMTSLTENbrCell [255]

### 8.529.1 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. |

### 8.529.2 Field Documentation

#### 8.529.2.1 nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

#### 8.529.2.2 uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

#### 8.529.2.3 uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCState

## 8.530 nas_WCDMARSSIThresh Struct Reference

### Data Fields

- uint8_t WCDMARSSIThreshListLen
- int16_t * pWCDMARSSIThreshList

### 8.530.1 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) |

### 8.530.2 Field Documentation

#### 8.530.2.1 int16_t* nas_WCDMARSSIThresh::pWCDMARSSIThreshList

#### 8.530.2.2 uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen

## 8.531 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

### 8.531.1 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 |

### 8.531.2 Field Documentation

#### 8.531.2.1 uint32_t nas_WCDMASysInfo::cellId

#### 8.531.2.2 uint8_t nas_WCDMASysInfo::cellIdValid

#### 8.531.2.3 uint8_t nas_WCDMASysInfo::hsCallStatus

#### 8.531.2.4 uint8_t nas_WCDMASysInfo::hsCallStatusValid

#### 8.531.2.5 uint8_t nas_WCDMASysInfo::hsInd

#### 8.531.2.6 uint8_t nas_WCDMASysInfo::hsIndValid

#### 8.531.2.7 uint16_t nas_WCDMASysInfo::lac

#### 8.531.2.8 uint8_t nas_WCDMASysInfo::lacValid

#### 8.531.2.9 uint8_t nas_WCDMASysInfo::MCC[3]

#### 8.531.2.10 uint8_t nas_WCDMASysInfo::MNC[3]

#### 8.531.2.11 uint8_t nas_WCDMASysInfo::networkIdValid

#### 8.531.2.12 uint16_t nas_WCDMASysInfo::psc

#### 8.531.2.13 uint8_t nas_WCDMASysInfo::pscValid

#### 8.531.2.14 uint8_t nas_WCDMASysInfo::regRejectInfoValid

#### 8.531.2.15 uint8_t nas_WCDMASysInfo::rejCause

#### 8.531.2.16 uint8_t nas_WCDMASysInfo::rejectSrvDomain

#### 8.531.2.17 nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA

## 8.532 nas_wcdmaUARFCN Struct Reference

### Data Fields

- uint8_t status
- uint32_t uarfcn

### 8.532.1 Detailed Description

This structure contains WCDMA UARFCN infromation.

**Parameters**

| | |
|---|---|
| *status* | • 0 - Disable<br>• 1 - Enable |
| *uarfcn* | • UARFCN to which UE is locked |

### 8.532.2 Field Documentation

#### 8.532.2.1 uint8_t nas_wcdmaUARFCN::status

#### 8.532.2.2 uint32_t nas_wcdmaUARFCN::uarfcn

## 8.533 NASBandPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t band_pref

### 8.533.1 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 |

### 8.533.2 Field Documentation

#### 8.533.2.1 uint64_t NASBandPreferenceTlv::band_pref

#### 8.533.2.2 uint8_t NASBandPreferenceTlv::TlvPresent

## 8.534 nasCellLocationInfoResp Struct Reference

### Data Fields

- GERANInfo * pGERANInfo
- UMTSInfo * pUMTSInfo
- CDMAInfo * pCDMAInfo
- LTEInfoIntrafreq * pLTEInfoIntrafreq
- LTEInfoInterfreq * pLTEInfoInterfreq
- LTEInfoNeighboringGSM * pLTEInfoNeighboringGSM
- LTEInfoNeighboringWCDMA * pLTEInfoNeighboringWCDMA
- ULONG * pUMTSCellID
- WCDMAInfoLTENeighborCell * pWCDMAInfoLTENeighborCell

### 8.534.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.

**Parameters**

| | |
|---|---|
| *pGERANInfo* | • See GERANInfo for more information. |
| *pUMTSInfo* | • See UMTSInfo for more information. |

| | |
|---|---|
| *pCDMAInfo* | • See CDMAInfo for more information. |
| *pLTEInfo-Intrafreq* | • See LTEInfoIntrafreq for more information. |
| *pLTEInfo-Interfreq* | • See LTEInfoInterfreq for more information. |
| *pLTEInfo-NeighboringGS-M* | • See LTEInfoNeighboringGSM for more information. |
| *pLTEInfo-NeighboringWC-DMA* | • See LTEInfoNeighboringWCDMA for more information. |
| *pUMTSCellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present. |
| *pWCDMAInfoLT-ENeighborCell* | • See WCDMAInfoLTENeighborCell for more information. |

### 8.534.2 Field Documentation

#### 8.534.2.1 CDMAInfo∗ nasCellLocationInfoResp::pCDMAInfo

#### 8.534.2.2 GERANInfo∗ nasCellLocationInfoResp::pGERANInfo

#### 8.534.2.3 LTEInfoInterfreq∗ nasCellLocationInfoResp::pLTEInfoInterfreq

#### 8.534.2.4 LTEInfoIntrafreq∗ nasCellLocationInfoResp::pLTEInfoIntrafreq

#### 8.534.2.5 LTEInfoNeighboringGSM∗ nasCellLocationInfoResp::pLTEInfoNeighboringGSM

#### 8.534.2.6 LTEInfoNeighboringWCDMA∗ nasCellLocationInfoResp::pLTEInfoNeighboringWCDMA

#### 8.534.2.7 ULONG∗ nasCellLocationInfoResp::pUMTSCellID

#### 8.534.2.8 UMTSInfo∗ nasCellLocationInfoResp::pUMTSInfo

#### 8.534.2.9 WCDMAInfoLTENeighborCell∗ nasCellLocationInfoResp::pWCDMAInfoLTENeighborCell

## 8.535 NASEmergencyModeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t EmerMode

### 8.535.1 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 |

### 8.535.2 Field Documentation

#### 8.535.2.1 uint8_t NASEmergencyModeTlv::EmerMode

#### 8.535.2.2 uint8_t NASEmergencyModeTlv::TlvPresent

## 8.536 nasGet3GPP2SubscriptionInfoReq Struct Reference

### Data Fields

- BYTE namID

### 8.536.1 Detailed Description

This structure contains the Get3GPP2SubscriptionInfo request parameters

**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. |

### 8.536.2 Field Documentation

#### 8.536.2.1 BYTE nasGet3GPP2SubscriptionInfoReq::namID

## 8.537 nasGet3GPP2SubscriptionInfoResp Struct Reference

### Data Fields

- namName * pNAMNameInfo
- dirNum * pDirNum
- homeSIDNID * pHomeSIDNID
- minBasedIMSI * pMinBasedIMSI
- trueIMSI * pTrueIMSI
- CDMAChannel * pCDMAChannel

### 8.537.1 Detailed Description

This structure contains the SLQSNasGet3GPP2Subscription response parameters.

**Parameters**

| | |
|---|---|
| *pNAMNameInfo* | [Optional]<br>• See namName for more information |
| *pDirNum* | [Optional]<br>• See dirNum for more information |
| *pHomeSIDNID* | [Optional]<br>• See homeSIDNID for more information |
| *pMinBasedIMSI* | [Optional]<br>• See minBasedIMSI for more information |
| *pTrueIMSI* | [Optional]<br>• See trueIMSI for more information |
| *pCDMAChannel* | [Optional]<br>• See CDMAChannel for more information |

### 8.537.2 Field Documentation

#### 8.537.2.1 CDMAChannel∗ nasGet3GPP2SubscriptionInfoResp::pCDMAChannel

#### 8.537.2.2 dirNum∗ nasGet3GPP2SubscriptionInfoResp::pDirNum

#### 8.537.2.3 homeSIDNID∗ nasGet3GPP2SubscriptionInfoResp::pHomeSIDNID

#### 8.537.2.4 minBasedIMSI∗ nasGet3GPP2SubscriptionInfoResp::pMinBasedIMSI

#### 8.537.2.5 namName∗ nasGet3GPP2SubscriptionInfoResp::pNAMNameInfo

#### 8.537.2.6 trueIMSI∗ nasGet3GPP2SubscriptionInfoResp::pTrueIMSI

## 8.538 nasGeteDRXParamsResp Struct Reference

### Data Fields

- BYTE ∗ pCycleLen
- BYTE ∗ pPagingTimeWindow
- BYTE ∗ pEdrxEnable

### 8.538.1 Detailed Description

This structure contains the SLQSNASGeteDRXParams request parameters.

**Parameters**

| | |
|---|---|
| *pCycleLen* | [Optional]<br>• eDRX cycle length<br>• Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32 |

| | |
|---|---|
| *pPagingTime-Window* | [Optional]<br>• eDRX paging time window<br>• Value is per 3GPP TS 24.008 Rel-13 Section10.5.5.32 |
| *pEdrxEnable* | [Optional]<br>• eDRX Enable<br>• Values<br>– 0x00 - Disable<br>– 0x01 - Enable |

### 8.538.2 Field Documentation

#### 8.538.2.1 BYTE* nasGeteDRXParamsResp::pCycleLen

#### 8.538.2.2 BYTE* nasGeteDRXParamsResp::pEdrxEnable

#### 8.538.2.3 BYTE* nasGeteDRXParamsResp::pPagingTimeWindow

## 8.539 nasGetHDRColorCodeResp Struct Reference

### Data Fields

- BYTE * pColorCode

### 8.539.1 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 |

### 8.539.2 Field Documentation

#### 8.539.2.1 BYTE* nasGetHDRColorCodeResp::pColorCode

## 8.540 nasGetLTECphyCa Struct Reference

### Data Fields

- PhyCaAggScellIndType sPhyCaAggScellIndType
- PhyCaAggScellDlBw sPhyCaAggScellDlBw
- PhyCaAggScellInfo sPhyCaAggScellInfo
- PhyCaAggPcellInfo sPhyCaAggPcellInfo
- PhyCaAggScellIndex sPhyCaAggScellIndex

### 8.540.1 Field Documentation

#### 8.540.1.1 PhyCaAggPcellInfo nasGetLTECphyCa::sPhyCaAggPcellInfo

#### 8.540.1.2 PhyCaAggScellDlBw nasGetLTECphyCa::sPhyCaAggScellDlBw

#### 8.540.1.3 PhyCaAggScellIndex nasGetLTECphyCa::sPhyCaAggScellIndex

#### 8.540.1.4 PhyCaAggScellIndType nasGetLTECphyCa::sPhyCaAggScellIndType

#### 8.540.1.5 PhyCaAggScellInfo nasGetLTECphyCa::sPhyCaAggScellInfo

## 8.541 NasGetLTECphyCaInfo Struct Reference

### Data Fields

- NASPhyCaAggScellIndType PhyCaAggScellIndType
- NASPhyCaAggScellDlBw PhyCaAggScellDlBw
- NASPhyCaAggScellInfo PhyCaAggScellInfo
- NASPhyCaAggPcellInfo PhyCaAggPcellInfo
- NASPhyCaAggScellIndex PhyCaAggScellIndex

### 8.541.1 Detailed Description

This structure contains the parameters for aggregation event information.

**Parameters**

| | |
|---|---|
| *PhyCaAggScell-IndType* | • see NASPhyCaAggScellIndType |
| *PhyCaAggScell-DlBw* | • see NASPhyCaAggScellDlBw |
| *PhyCaAggScell-Info* | • see NASPhyCaAggScellInfo |
| *PhyCaAggScell-IndType* | • see NASPhyCaAggScellIndType |
| *PhyCaAggPcell-Info* | • see NASPhyCaAggPcellInfo |
| *PhyCaAggScell-Index* | • see NASPhyCaAggScellIndex |

### 8.541.2 Field Documentation

#### 8.541.2.1 NASPhyCaAggPcellInfo NasGetLTECphyCaInfo::PhyCaAggPcellInfo

#### 8.541.2.2 NASPhyCaAggScellDlBw NasGetLTECphyCaInfo::PhyCaAggScellDlBw

#### 8.541.2.3 NASPhyCaAggScellIndex NasGetLTECphyCaInfo::PhyCaAggScellIndex

#### 8.541.2.4 NASPhyCaAggScellIndType NasGetLTECphyCaInfo::PhyCaAggScellIndType

#### 8.541.2.5 NASPhyCaAggScellInfo NasGetLTECphyCaInfo::PhyCaAggScellInfo

## 8.542 nasGetLTECphyCaResp Struct Reference

### Data Fields

- PhyCaAggScellIndType * pPhyCaAggScellIndType
- PhyCaAggScellDlBw * pPhyCaAggScellDlBw
- PhyCaAggScellInfo * pPhyCaAggScellInfo
- PhyCaAggPcellInfo * pPhyCaAggPcellInfo
- PhyCaAggScellIndex * pPhyCaAggScellIndex

### 8.542.1 Field Documentation

#### 8.542.1.1 PhyCaAggPcellInfo* nasGetLTECphyCaResp::pPhyCaAggPcellInfo

#### 8.542.1.2 PhyCaAggScellDlBw* nasGetLTECphyCaResp::pPhyCaAggScellDlBw

#### 8.542.1.3 PhyCaAggScellIndex* nasGetLTECphyCaResp::pPhyCaAggScellIndex

#### 8.542.1.4 PhyCaAggScellIndType* nasGetLTECphyCaResp::pPhyCaAggScellIndType

#### 8.542.1.5 PhyCaAggScellInfo* nasGetLTECphyCaResp::pPhyCaAggScellInfo

## 8.543 NasGetRFInfoResp Struct Reference

### Data Fields

- RfBandInfoList rfbandInfoList
- RfDedicatedBandInfo * pRfDedicatedBandInfo
- RfBandInfoExtFormat * pRfBandInfoExtFormat
- RfBandwidthInfo * pRfBandwidthInfo
- LTEOperationMode * pLTEOperationMode

### 8.543.1 Detailed Description

This structure contains the SLQSNasGetRFInfo response parameters.

**Parameters**

| | |
|---|---|
| *rfbandInfoList* | • See RfBandInfoList for more information |
| *pRfDedicated-BandInfo* | [Optional]<br>• See RfDedicatedBandInfo for more information |
| *pRfBandInfoExt-Format* | [Optional]<br>• See RfBandInfoExtFormat for more information |

| | |
|---|---|
| *pRfBandwidth-Info* | [Optional]<br>• See RfBandwidthInfo for more information |
| *pLTEOperation-Mode* | [Optional]<br>• See LTEOperationMode for more information |

### 8.543.2 Field Documentation

#### 8.543.2.1 LTEOperationMode∗ NasGetRFInfoResp::pLTEOperationMode

#### 8.543.2.2 RfBandInfoExtFormat∗ NasGetRFInfoResp::pRfBandInfoExtFormat

#### 8.543.2.3 RfBandwidthInfo∗ NasGetRFInfoResp::pRfBandwidthInfo

#### 8.543.2.4 RfDedicatedBandInfo∗ NasGetRFInfoResp::pRfDedicatedBandInfo

#### 8.543.2.5 RfBandInfoList NasGetRFInfoResp::rfbandInfoList

## 8.544 nasGetSigInfoResp Struct Reference

### Data Fields

- CDMASSInfo ∗ pCDMASSInfo
- HDRSSInfo ∗ pHDRSSInfo
- INT8 ∗ pGSMSSInfo
- CDMASSInfo ∗ pWCDMASSInfo
- LTESSInfo ∗ pLTESSInfo
- INT8 ∗ pTDSCDMASigInfoRscp
- TDSCDMASigInfoExt ∗ pTDSCDMASigInfoExt

### 8.544.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

**Parameters**

| | |
|---|---|
| *pCDMASSInfo* | [Optional]<br>• See CDMASSInfo for more information |
| *pHDRSSInfo* | [Optional]<br>• See HDRSSInfo for more information |
| *pGSMSSInfo* | [Optional]<br>• GSM signal strength is the RSSI in dBm (signed value).<br>• A value of -125 dBm or lower is used to indicate No Signal. |
| *pWCDMASSInfo* | [Optional]<br>• See CDMASSInfo for more information |
| *pLTESSInfo* | [Optional]<br>• See LTESSInfo for more information |

| | |
|---|---|
| *pTDSCDMASig-InfoRscp* | [Optional]<br>• RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm.<br>• Measurement range: -120 dBm to -25 dBm. |
| *pTDSCDMASig-InfoExt* | [Optional]<br>• See TDSCDMASigInfoExt for more information. |

### 8.544.2 Field Documentation

#### 8.544.2.1 CDMASSInfo* nasGetSigInfoResp::pCDMASSInfo

#### 8.544.2.2 INT8* nasGetSigInfoResp::pGSMSSInfo

#### 8.544.2.3 HDRSSInfo* nasGetSigInfoResp::pHDRSSInfo

#### 8.544.2.4 LTESSInfo* nasGetSigInfoResp::pLTESSInfo

#### 8.544.2.5 TDSCDMASigInfoExt* nasGetSigInfoResp::pTDSCDMASigInfoExt

#### 8.544.2.6 INT8* nasGetSigInfoResp::pTDSCDMASigInfoRscp

#### 8.544.2.7 CDMASSInfo* nasGetSigInfoResp::pWCDMASSInfo

## 8.545 nasGetSysInfoResp Struct Reference

### Data Fields

- SrvStatusInfo * pCDMASrvStatusInfo
- SrvStatusInfo * pHDRSrvStatusInfo
- GSMSrvStatusInfo * pGSMSrvStatusInfo
- GSMSrvStatusInfo * pWCDMASrvStatusInfo
- GSMSrvStatusInfo * pLTESrvStatusInfo
- CDMASysInfo * pCDMASysInfo
- HDRSysInfo * pHDRSysInfo
- GSMSysInfo * pGSMSysInfo
- WCDMASysInfo * pWCDMASysInfo
- LTESysInfo * pLTESysInfo
- AddCDMASysInfo * pAddCDMASysInfo
- WORD * pAddHDRSysInfo
- AddSysInfo * pAddGSMSysInfo
- AddSysInfo * pAddWCDMASysInfo
- WORD * pAddLTESysInfo
- CallBarringSysInfo * pGSMCallBarringSysInfo
- CallBarringSysInfo * pWCDMACallBarringSysInfo
- BYTE * pLTEVoiceSupportSysInfo
- BYTE * pGSMCipherDomainSysInfo
- BYTE * pWCDMACipherDomainSysInfo

### 8.545.1 Detailed Description

Structure for storing the SLQSNasGetSysInfo response parameters.

**Parameters**

| | |
|---|---|
| *pCDMASrvStatusInfo* | • See SrvStatusInfo for more information. |
| *pHDRSrvStatusInfo* | • See SrvStatusInfo for more information. |
| *pGSMSrvStatusInfo* | • See GSMSrvStatusInfo for more information. |
| *pWCDMASrvStatusInfo* | • See GSMSrvStatusInfo for more information. |
| *pLTESrvStatusInfo* | • See GSMSrvStatusInfo for more information. |
| *pCDMASysInfo* | • See CDMASysInfo for more information. |
| *pHDRSysInfo* | • See HDRSysInfo for more information. |
| *pGSMSysInfo* | • See GSMSysInfo for more information. |
| *pWCDMASysInfo* | • See WCDMASysInfo for more information. |
| *pLTESysInfo* | • See LTESysInfo for more information. |
| *pAddCDMASysInfo* | • See AddCDMASysInfo for more information. |
| *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. |
| *pAddGSMSysInfo* | • See AddSysInfo for more information. |
| *pAddWCDMASysInfo* | • See AddSysInfo for more information. |
| *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. |

| | |
|---|---|
| *pGSMCall-BarringSysInfo* | • See CallBarringSysInfo for more information. |
| *pWCDMACall-BarringSysInfo* | • See CallBarringSysInfo for more information. |
| *pLTEVoice-SupportSysInfo* | • Indicates voice support status on LTE.<br>– 0x00 - Voice is not supported<br>– 0x01 - Voice is supported |
| *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 |
| *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 |

## 8.545.2 Field Documentation

### 8.545.2.1 AddCDMASysInfo∗ nasGetSysInfoResp::pAddCDMASysInfo

### 8.545.2.2 AddSysInfo∗ nasGetSysInfoResp::pAddGSMSysInfo

### 8.545.2.3 WORD∗ nasGetSysInfoResp::pAddHDRSysInfo

### 8.545.2.4 WORD∗ nasGetSysInfoResp::pAddLTESysInfo

### 8.545.2.5 AddSysInfo∗ nasGetSysInfoResp::pAddWCDMASysInfo

### 8.545.2.6 SrvStatusInfo∗ nasGetSysInfoResp::pCDMASrvStatusInfo

### 8.545.2.7 CDMASysInfo∗ nasGetSysInfoResp::pCDMASysInfo

### 8.545.2.8 CallBarringSysInfo∗ nasGetSysInfoResp::pGSMCallBarringSysInfo

### 8.545.2.9 BYTE∗ nasGetSysInfoResp::pGSMCipherDomainSysInfo

### 8.545.2.10 GSMSrvStatusInfo∗ nasGetSysInfoResp::pGSMSrvStatusInfo

### 8.545.2.11 GSMSysInfo∗ nasGetSysInfoResp::pGSMSysInfo

### 8.545.2.12 SrvStatusInfo∗ nasGetSysInfoResp::pHDRSrvStatusInfo

#### 8.545.2.13 HDRSysInfo∗ nasGetSysInfoResp::pHDRSysInfo

#### 8.545.2.14 GSMSrvStatusInfo∗ nasGetSysInfoResp::pLTESrvStatusInfo

#### 8.545.2.15 LTESysInfo∗ nasGetSysInfoResp::pLTESysInfo

#### 8.545.2.16 BYTE∗ nasGetSysInfoResp::pLTEVoiceSupportSysInfo

#### 8.545.2.17 CallBarringSysInfo∗ nasGetSysInfoResp::pWCDMACallBarringSysInfo

#### 8.545.2.18 BYTE∗ nasGetSysInfoResp::pWCDMACipherDomainSysInfo

#### 8.545.2.19 GSMSrvStatusInfo∗ nasGetSysInfoResp::pWCDMASrvStatusInfo

#### 8.545.2.20 WCDMASysInfo∗ nasGetSysInfoResp::pWCDMASysInfo

## 8.546 nasGetTxRxInfoReq Struct Reference

### Data Fields

- BYTE radio_if

### 8.546.1 Detailed Description

This structure contains the GetTxRxInfoReq request parameters

**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 |

### 8.546.2 Field Documentation

#### 8.546.2.1 BYTE nasGetTxRxInfoReq::radio_if

## 8.547 nasGetTxRxInfoResp Struct Reference

### Data Fields

- rxInfo ∗ pRXChain0Info
- rxInfo ∗ pRXChain1Info
- txInfo ∗ pTXInfo

### 8.547.1 Detailed Description

This structure contains the GetTxRxInfoResp response parameters.

**Parameters**

| | |
|---|---|
| *pRXChain0Info* | [Optional]<br>• See rxInfo for more information. |
| *pRXChain1Info* | [Optional]<br>• See rxInfo for more information. |
| *pTXInfo* | [Optional]<br>• See txInfo for more information. |

### 8.547.2 Field Documentation

#### 8.547.2.1 rxInfo∗ nasGetTxRxInfoResp::pRXChain0Info

#### 8.547.2.2 rxInfo∗ nasGetTxRxInfoResp::pRXChain1Info

#### 8.547.2.3 txInfo∗ nasGetTxRxInfoResp::pTXInfo

## 8.548 NASGWAcqOrderPrefTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t GWAcqOrderPref

### 8.548.1 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 |

### 8.548.2 Field Documentation

#### 8.548.2.1 uint32_t NASGWAcqOrderPrefTlv::GWAcqOrderPref

#### 8.548.2.2 uint8_t NASGWAcqOrderPrefTlv::TlvPresent

## 8.549 nasIndicationRegisterReq Struct Reference

### Data Fields

- BYTE ∗ pSystemSelectionInd
- BYTE ∗ pDDTMInd

- BYTE * pServingSystemInd
- BYTE * pDualStandByPrefInd
- BYTE * pSubscriptionInfoInd
- BYTE * pNetworkTimeInd
- BYTE * pSysInfoInd
- BYTE * pSignalStrengthInd
- BYTE * pErrorRateInd
- BYTE * pHDRNewUATIAssInd
- BYTE * pHDRSessionCloseInd
- BYTE * pManagedRoamingInd
- BYTE * pNetworkRejectInd
- BYTE * pSuppressSysInfoInd
- BYTE * pLTECphyCa

### 8.549.1 Detailed Description

This structure contains the SLQSNasIndicationRegisterExt request parameters.

**Parameters**

| | |
|---|---|
| *pSystemSelectionInd* | [Optional]<br>• System Selection Preference indication registration. The following callbacks would not be invoked if the indication is disabled.<br>tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pDDTMInd* | [Optional]<br>• DDTM (Data Dedicated Transmission Mode) indication registration. The following callbacks would not be invoked if the indication is disabled.<br>tFNDDTM<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pServingSystemInd* | [Optional]<br>• Serving System indication registration. The following callbacks would not be invoked if the indication is disabled.<br>tFNBandPreference<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pDualStandByPrefInd* | [Optional]<br>• Dual Standby Preference indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNDualStandByPref<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSubscriptionInfoInd* | [Optional]<br>• Subscription Information indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNSubscriptionInfo<br>– 0x00 - Disable<br>– 0x01 - Enable |

| | |
|---|---|
| *pNetworkTimeInd* | [Optional]<br>• Network Time indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNNetworkTime<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSysInfoInd* | [Optional]<br>• System Information indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNSysInfo<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSignalStrengthInd* | [Optional]<br>• Signal Strength indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNSigInfo<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pErrorRateInd* | [Optional]<br>• Error Rate indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNErrRate<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pHDRNewUATIAssInd* | [Optional]<br>• HDR New UATI Assigned indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNHDRUATIUpdate<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pHDRSessionCloseInd* | [Optional]<br>• HDR Session Closed indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNHDRSessionClose<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pManagedRoamingInd* | [Optional]<br>• Managed Roaming indication registration.The following callbacks would not be invoked if the indication is disabled.<br>tFNManagedRoaming<br>– 0x00 - Disable<br>– 0x01 - Enable |

| | |
|---|---|
| *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 |
| *pLTECphyCa* | [Optional]<br>• LTE Physical Carrier Aggregation Information indication registration.The following callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |

**Note**

Atleast one parameter must be provided as request. 'NULL' value confirms that the indication value is not sent.

### 8.549.2 Field Documentation

#### 8.549.2.1 BYTE* nasIndicationRegisterReq::pDDTMInd

#### 8.549.2.2 BYTE* nasIndicationRegisterReq::pDualStandByPrefInd

#### 8.549.2.3 BYTE* nasIndicationRegisterReq::pErrorRateInd

#### 8.549.2.4 BYTE* nasIndicationRegisterReq::pHDRNewUATIAssInd

#### 8.549.2.5 BYTE* nasIndicationRegisterReq::pHDRSessionCloseInd

#### 8.549.2.6 BYTE* nasIndicationRegisterReq::pLTECphyCa

#### 8.549.2.7 BYTE* nasIndicationRegisterReq::pManagedRoamingInd

#### 8.549.2.8 BYTE* nasIndicationRegisterReq::pNetworkRejectInd

#### 8.549.2.9 BYTE* nasIndicationRegisterReq::pNetworkTimeInd

#### 8.549.2.10 BYTE* nasIndicationRegisterReq::pServingSystemInd

#### 8.549.2.11 BYTE* nasIndicationRegisterReq::pSignalStrengthInd

#### 8.549.2.12 BYTE* nasIndicationRegisterReq::pSubscriptionInfoInd

#### 8.549.2.13 BYTE* nasIndicationRegisterReq::pSuppressSysInfoInd

#### 8.549.2.14 BYTE* nasIndicationRegisterReq::pSysInfoInd

#### 8.549.2.15 BYTE* nasIndicationRegisterReq::pSystemSelectionInd

## 8.550 nasInitNetworkReg Struct Reference

### Data Fields

- ULONG regAction
- MNRInfo * pMNRInfo
- ULONG * pChangeDuration
- BYTE * pMncPcsDigitStatus

### 8.550.1 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 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. |

### 8.550.2 Field Documentation

#### 8.550.2.1 ULONG* nasInitNetworkReg::pChangeDuration

#### 8.550.2.2 BYTE* nasInitNetworkReg::pMncPcsDigitStatus

#### 8.550.2.3 MNRInfo* nasInitNetworkReg::pMNRInfo

#### 8.550.2.4 ULONG nasInitNetworkReg::regAction

## 8.551 NASLTEBandPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t LTEBandPref

### 8.551.1 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 |

### 8.551.2 Field Documentation

#### 8.551.2.1 uint64_t NASLTEBandPreferenceTlv::LTEBandPref

#### 8.551.2.2 uint8_t NASLTEBandPreferenceTlv::TlvPresent

## 8.552 NASLteNasReleaseInfoTlv Struct Reference

### Data Fields

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

### 8.552.1 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 |

### 8.552.2 Field Documentation

#### 8.552.2.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

#### 8.552.2.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

#### 8.552.2.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

#### 8.552.2.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

## 8.553 NASModePreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t ModePref

### 8.553.1 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 |

### 8.553.2 Field Documentation

#### 8.553.2.1 uint16_t NASModePreferenceTlv::ModePref

#### 8.553.2.2 uint8_t NASModePreferenceTlv::TlvPresent

## 8.554 NASNetSelPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t NetSelPref

### 8.554.1 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 |

### 8.554.2 Field Documentation

#### 8.554.2.1 uint8_t NASNetSelPreferenceTlv::NetSelPref

#### 8.554.2.2 uint8_t NASNetSelPreferenceTlv::TlvPresent

## 8.555 nasNetworkReject Struct Reference

## Data Fields

- BYTE radioIf
- TYPE_OF_SERVICE_DOMAIN serviceDomain
- BYTE rejectCause
- PlmnID * pPlmnId
- CsgId * pCsgId

### 8.555.1 Detailed Description

Structure for storing the network reject indication parameters.

**Parameters**

| | |
|---|---|
| *radioIf* | • Radio interface from which to get the information. Values:<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 - TDSCDMA |
| *serviceDomain* | • Type of service domain in which the registration is rejected. Values:<br>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service<br>– 0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only<br>– 0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only<br>– 0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched<br>– 0x04 - SYS_SRV_DOMAIN_CAMPED - Camped |

| | |
|---|---|
| *rejectCause* | • Reject cause values.<br>– 2 - IMSI unknown in HLR<br>– 3 - Illegal MS<br>– 4 - IMSI unknown in VLR<br>– 5 - IMEI not accepted<br>– 6 - Illegal ME<br>– 7 - EPS services not allowed<br>– 8 - EPS services and non-EPS services not allowed<br>– 9 - UE identity cannot be derived by the network<br>– 10 - Implicitly detached<br>– 11 - PLMN not allowed<br>– 12 - Location Area not allowed<br>– 13 - Roaming not allowed in this location area<br>– 14 - EPS services not allowed in this PLMN<br>– 15 - No Suitable Cells In Location Area<br>– 16 - MSC temporarily not reachable<br>– 17 - Network failure<br>– 18 - CS domain not available<br>– 19 - ESM failure<br>– 20 - MAC failure<br>– 21 - Synch failure<br>– 22 - Congestion<br>– 23 - GSM authentication unacceptable<br>– 24 - Security mode rejected, unspecified<br>– 25 - Not authorized for this CSG<br>– 26 - Non-EPS authentication unacceptable<br>– 32 - Service option not supported<br>– 33 - Requested service option not subscribed<br>– 34 - Service option temporarily out of order<br>– 38 - Call cannot be identified<br>– 48 to 63 - retry upon entry into a new cell<br>– 95 - Semantically incorrect message<br>– 96 - Invalid mandatory information<br>– 97 - Message type non-existent or not implemented<br>– 98 - Message type not compatible with the protocol state<br>– 99 - Information element non-existent or not implemented<br>– 100 - Conditional IE error<br>– 101 - Message not compatible with the protocol state<br>– 111 - Protocol error, unspecified<br>– Note - Any other value received by the mobile station shall be treated as 34, 'Service option temporarily out of order'.<br>* Any other value received by the network shall be treated as 111, 'Protocol error, unspecified'.<br>See 3GPP TS 24.008, Section 4.4.4.7 and Section 10.5.3.6 See qaGobiApiTableCallEndReasons.h for Call End reasons |
| *pPlmnId* | (optional)<br>• See PlmnID for more information |

| | |
|---|---|
| *pCsgId* | (optional)<br>• See CsgId for more information. |

### 8.555.2 Field Documentation

#### 8.555.2.1 CsgId* nasNetworkReject::pCsgId

#### 8.555.2.2 PlmnID* nasNetworkReject::pPlmnId

#### 8.555.2.3 BYTE nasNetworkReject::radioIf

#### 8.555.2.4 BYTE nasNetworkReject::rejectCause

#### 8.555.2.5 TYPE_OF_SERVICE_DOMAIN nasNetworkReject::serviceDomain

## 8.556 nasNetworkTime Struct Reference

### Data Fields

- UniversalTime universalTime
- BYTE * pTimeZone
- BYTE * pDayItSavAdj
- BYTE * pRadioInterface

### 8.556.1 Detailed Description

Structure for storing the nasSysInfo indication parameters.

**Parameters**

| | |
|---|---|
| *universalTime* | • See UniversalTime for more information. |
| *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). |
| *pDayItSavAdj* | • Daylight Saving Adjustment.<br>• Daylight saving adjustment in hr.<br>– Possible values: 0, 1, and 2. |

| | |
|---|---|
| *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 |

### 8.556.2 Field Documentation

#### 8.556.2.1 BYTE* nasNetworkTime::pDayltSavAdj

#### 8.556.2.2 BYTE* nasNetworkTime::pRadioInterface

#### 8.556.2.3 BYTE* nasNetworkTime::pTimeZone

#### 8.556.2.4 UniversalTime nasNetworkTime::universalTime

## 8.557 nasOperatorNameResp Struct Reference

### Data Fields

- serviceProviderName * pSrvcProviderName
- operatorPLMNList * pOperatorPLMNList
- PLMNNetworkName * pPLMNNetworkName
- operatorNameString * pOperatorNameString
- PLMNNetworkNameData * pNITZInformation

### 8.557.1 Detailed Description

This structure contains Operator Name Data related from multiple sources.

**Parameters**

| | |
|---|---|
| *pSrvcProvider-Name* | • Refer serviceProviderName for details (Optional).<br>• Can provide NULL if this parameter is not required. |
| *pOperatorPLM-NList* | • Refer operatorPLMNList for details (Optional).<br>• Can provide NULL if this parameter is not required. |
| *pPLMNNetwork-Name* | • Refer PLMNNetworkName for details (Optional).<br>• Can provide NULL if this parameter is not required. |
| *pOperatorName-String* | • Refer operatorNameString for details (Optional).<br>• Can provide NULL if this parameter is not required. |

| | |
|---|---|
| *pNITZ-Information* | • Refer PLMNNetworkNameData for details (Optional).<br>• Can provide NULL if this parameter is not required. |

### 8.557.2 Field Documentation

#### 8.557.2.1 PLMNNetworkNameData∗ nasOperatorNameResp::pNITZInformation

#### 8.557.2.2 operatorNameString∗ nasOperatorNameResp::pOperatorNameString

#### 8.557.2.3 operatorPLMNList∗ nasOperatorNameResp::pOperatorPLMNList

#### 8.557.2.4 PLMNNetworkName∗ nasOperatorNameResp::pPLMNNetworkName

#### 8.557.2.5 serviceProviderName∗ nasOperatorNameResp::pSrvcProviderName

## 8.558 NASOTAMessageTlv Struct Reference

### Data Fields

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

### 8.558.1 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 |

### 8.558.2 Field Documentation

#### 8.558.2.1 uint8_t NASOTAMessageTlv::data_buf[2048]

#### 8.558.2.2 uint16_t NASOTAMessageTlv::data_len

#### 8.558.2.3 uint32_t NASOTAMessageTlv::message_type

#### 8.558.2.4 uint8_t NASOTAMessageTlv::TlvPresent

## 8.559 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

### 8.559.1 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. |

### 8.559.2 Field Documentation

#### 8.559.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value

#### 8.559.2.2 uint32_t NASPhyCaAggPcellInfo::freq

#### 8.559.2.3 uint32_t NASPhyCaAggPcellInfo::iLTEbandValue

#### 8.559.2.4 uint32_t NASPhyCaAggPcellInfo::pci

#### 8.559.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

## 8.560 NASPhyCaAggScellDlBw Struct Reference

### Data Fields

- LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- uint8_t TlvPresent

### 8.560.1 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. |

### 8.560.2 Field Documentation

#### 8.560.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellDlBw::dl_bw_value

#### 8.560.2.2 uint8_t NASPhyCaAggScellDlBw::TlvPresent

## 8.561 NASPhyCaAggScellIndex Struct Reference

### Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

### 8.561.1 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. |

### 8.561.2 Field Documentation

#### 8.561.2.1 uint8_t NASPhyCaAggScellIndex::scell_idx

#### 8.561.2.2 uint8_t NASPhyCaAggScellIndex::TlvPresent

## 8.562 NASPhyCaAggScellIndType Struct Reference

### Data Fields

- uint32_t pci
- uint32_t freq
- LITEQMI_NAS_LTE_CPHY_SCELL_STATE scell_state
- uint8_t TlvPresent

### 8.562.1 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. |

### 8.562.2 Field Documentation

#### 8.562.2.1 uint32_t NASPhyCaAggScellIndType::freq

#### 8.562.2.2 uint32_t NASPhyCaAggScellIndType::pci

#### 8.562.2.3 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

#### 8.562.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

## 8.563 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

### 8.563.1 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. |

### 8.563.2 Field Documentation

#### 8.563.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

#### 8.563.2.2 uint32_t NASPhyCaAggScellInfo::freq

#### 8.563.2.3 uint32_t NASPhyCaAggScellInfo::iLTEbandValue

#### 8.563.2.4 uint32_t NASPhyCaAggScellInfo::pci

#### 8.563.2.5 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

#### 8.563.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

## 8.564 nasPLMNNameReq Struct Reference

### Data Fields

- WORD mcc
- WORD mnc

- BYTE * pMncPcsStatus

### 8.564.1 Detailed Description

Structure for storing the PLMN Name request 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.

### 8.564.2 Field Documentation

#### 8.564.2.1 WORD nasPLMNNameReq::mcc

#### 8.564.2.2 WORD nasPLMNNameReq::mnc

#### 8.564.2.3 BYTE* nasPLMNNameReq::pMncPcsStatus

## 8.565 nasPLMNNameResp Struct Reference

### Data Fields

- BYTE spnEncoding
- BYTE spnLength
- BYTE spn [255]
- BYTE shortNameEn
- BYTE shortNameCI
- BYTE shortNameSB
- BYTE shortNameLen
- BYTE shortName [255]
- BYTE longNameEn
- BYTE longNameCI
- BYTE longNameSB
- BYTE longNameLen
- BYTE longName [255]

### 8.565.1 Detailed Description

Structure for storing the PLMN Name response 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 |
| *spnLength* | • Length of SPN which follows |
| *spn* | • Service Provider name string |
| *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 |
| *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 |
| *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 |
| *shortNameLen* | • Length of shortName which follows |
| *shortName* | • PLMN short name |
| *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 |

| | |
|---|---|
| *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 |
| *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 |
| *longNameLen* | • Length of longName which follows |
| *longName* | • PLMN long name |

**Note**

None

### 8.565.2 Field Documentation

#### 8.565.2.1 BYTE nasPLMNNameResp::longName[255]

#### 8.565.2.2 BYTE nasPLMNNameResp::longNameCI

#### 8.565.2.3 BYTE nasPLMNNameResp::longNameEn

#### 8.565.2.4 BYTE nasPLMNNameResp::longNameLen

#### 8.565.2.5 BYTE nasPLMNNameResp::longNameSB

#### 8.565.2.6 BYTE nasPLMNNameResp::shortName[255]

#### 8.565.2.7 BYTE nasPLMNNameResp::shortNameCI

#### 8.565.2.8 BYTE nasPLMNNameResp::shortNameEn

#### 8.565.2.9 BYTE nasPLMNNameResp::shortNameLen

#### 8.565.2.10 BYTE nasPLMNNameResp::shortNameSB

#### 8.565.2.11 BYTE nasPLMNNameResp::spn[255]

#### 8.565.2.12 BYTE nasPLMNNameResp::spnEncoding

#### 8.565.2.13 BYTE nasPLMNNameResp::spnLength

## 8.566 NASPRLPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t PRLPref

### 8.566.1 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 |

### 8.566.2 Field Documentation

#### 8.566.2.1 uint16_t NASPRLPreferenceTlv::PRLPref

#### 8.566.2.2 uint8_t NASPRLPreferenceTlv::TlvPresent

## 8.567 NASQmiCbkNasSwiOTAMessageInd Struct Reference

### Data Fields

- NASOTAMessageTlv otaMsgTlv
- NASLteNasReleaseInfoTlv nasRelInfoTlv
- NASTimeInfoTlv timeTlv

### 8.567.1 Detailed Description

This structure contains OTA message

**Parameters**

| | |
|---|---|
| *otaMsgTlv* | • See NASOTAMessageTlv for more details |
| *nasRelInfoTlv* | • See NASLteNasReleaseInfoTlv for more details |
| *timeTlv* | • See NASTimeInfoTlv for more details |

### 8.567.2 Field Documentation

#### 8.567.2.1 NASLteNasReleaseInfoTlv NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv

#### 8.567.2.2 NASOTAMessageTlv NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv

#### 8.567.2.3 NASTimeInfoTlv NASQmiCbkNasSwiOTAMessageInd::timeTlv

## 8.568 NASQmiCbkNasSystemSelPrefInd Struct Reference

### Data Fields

- NASEmergencyModeTlv EMTlv
- NASModePreferenceTlv MPTlv
- NASBandPreferenceTlv BPTlv
- NASPRLPreferenceTlv PRLPTlv
- NASRoamPreferenceTlv RPTlv
- NASLTEBandPreferenceTlv LBPTlv
- NASNetSelPreferenceTlv NSPTlv
- NASServDomainPrefTlv SDPTlv
- NASGWAcqOrderPrefTlv GWAOPTlv

### 8.568.1 Detailed Description

Structure used to store all QMI Notification parameters.

**Parameters**

| | |
|---|---|
| *EMTlv* | - Emergency mode Tlv<br>• See NASEmergencyModeTlv for more details |
| *MPTlv* | - Mode preference Tlv<br>• See NASModePreferenceTlv for more details |
| *BPTlv* | - Band preference Tlv<br>• See NASBandPreferenceTlv for more details |
| *PRLPTlv* | - PRL preference Tlv<br>• See NASPRLPreferenceTlv for more details |
| *RPTlv* | - Roaming preference Tlv<br>• See NASRoamPreferenceTlv for more details |
| *LBPTlv* | - LTE band preference Tlv<br>• See NASLTEBandPreferenceTlv for more details |
| *NSPTlv* | - Network selection preference Tlv<br>• See NASNetSelPreferenceTlv for more details |
| *SDPTlv* | - Service domain preference Tlv<br>• See NASServDomainPrefTlv for more details |
| *GWAOPTlv* | - GSM WCDMA Acquisition Order Preference Tlv<br>• See NASGWAcqOrderPrefTlv for more details |

Note: None

### 8.568.2 Field Documentation

#### 8.568.2.1 NASBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::BPTlv

#### 8.568.2.2 NASEmergencyModeTlv NASQmiCbkNasSystemSelPrefInd::EMTlv

#### 8.568.2.3 NASGWAcqOrderPrefTlv NASQmiCbkNasSystemSelPrefInd::GWAOPTlv

#### 8.568.2.4 NASLTEBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::LBPTlv

#### 8.568.2.5 NASModePreferenceTlv NASQmiCbkNasSystemSelPrefInd::MPTlv

#### 8.568.2.6 NASNetSelPreferenceTlv NASQmiCbkNasSystemSelPrefInd::NSPTlv

#### 8.568.2.7 NASPRLPreferenceTlv NASQmiCbkNasSystemSelPrefInd::PRLPTlv

#### 8.568.2.8 NASRoamPreferenceTlv NASQmiCbkNasSystemSelPrefInd::RPTlv

#### 8.568.2.9 NASServDomainPrefTlv NASQmiCbkNasSystemSelPrefInd::SDPTlv

## 8.569 NASRoamPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t RoamPref

### 8.569.1 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 |

### 8.569.2 Field Documentation

#### 8.569.2.1 uint16_t NASRoamPreferenceTlv::RoamPref

#### 8.569.2.2 uint8_t NASRoamPreferenceTlv::TlvPresent

## 8.570 NASServDomainPrefTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t SrvDomainPref

### 8.570.1 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 |

### 8.570.2 Field Documentation

#### 8.570.2.1 uint32_t NASServDomainPrefTlv::SrvDomainPref

#### 8.570.2.2 uint8_t NASServDomainPrefTlv::TlvPresent

## 8.571 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

### 8.571.1 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

### 8.571.2 Field Documentation

#### 8.571.2.1 uint8_t NASServingSystemInfo::csAttachState

#### 8.571.2.2 uint8_t NASServingSystemInfo::hdrPersonality

#### 8.571.2.3 uint8_t NASServingSystemInfo::psAttachState

#### 8.571.2.4 uint8_t NASServingSystemInfo::radioInterfaceList[255]

#### 8.571.2.5 uint8_t NASServingSystemInfo::radioInterfaceNo

#### 8.571.2.6 uint8_t NASServingSystemInfo::registrationState

#### 8.571.2.7 uint8_t NASServingSystemInfo::selectedNetwork

## 8.572 nasSeteDRXParamsReq Struct Reference

### Data Fields

- BYTE * pEdrxEnable
- BYTE * pCycleLen
- BYTE * pPagingTimeWindow

### 8.572.1 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 |

### 8.572.2 Field Documentation

#### 8.572.2.1 BYTE* nasSeteDRXParamsReq::pCycleLen

#### 8.572.2.2 BYTE* nasSeteDRXParamsReq::pEdrxEnable

#### 8.572.2.3 BYTE* nasSeteDRXParamsReq::pPagingTimeWindow

## 8.573 nasSigInfo Struct Reference

### Data Fields

- CDMASSInfo * pCDMASigInfo
- HDRSSInfo * pHDRSigInfo
- INT8 * pGSMSigInfo
- CDMASSInfo * pWCDMASigInfo
- LTESSInfo * pLTESigInfo
- INT8 * pRscp
- TDSCDMASigInfoExt * pTDSCDMASigInfoExt

### 8.573.1 Detailed Description

Structure for storing the nasSigInfo indication parameters.

**Parameters**

| | |
|---|---|
| *pCDMASigInfo* | • See CDMASSInfo for more information. |
| *pHDRSigInfo* | • See HDRSSInfo for more information. |
| *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 |

| | |
|---|---|
| *pWCDMASigInfo* | • See CDMASSInfo for more information. |
| *pLTESigInfo* | • See LTESSInfo for more information. |
| *pRscp* | • RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. Measurement range: -120 dBm to -25 dBm. |
| *pTDSCDMASig-InfoExt* | • See TDSCDMASigInfoExt for more information. |

### 8.573.2 Field Documentation

#### 8.573.2.1 CDMASSInfo* nasSigInfo::pCDMASigInfo

#### 8.573.2.2 INT8* nasSigInfo::pGSMSigInfo

#### 8.573.2.3 HDRSSInfo* nasSigInfo::pHDRSigInfo

#### 8.573.2.4 LTESSInfo* nasSigInfo::pLTESigInfo

#### 8.573.2.5 INT8* nasSigInfo::pRscp

#### 8.573.2.6 TDSCDMASigInfoExt* nasSigInfo::pTDSCDMASigInfoExt

#### 8.573.2.7 CDMASSInfo* nasSigInfo::pWCDMASigInfo

## 8.574 nasSwiGetChannelLockResp Struct Reference

### Data Fields

- wcdmaUARFCN * pWcdmaUARFCN
- lteEARFCN * pLteEARFCN
- ltePCI * pLtePCI

### 8.574.1 Detailed Description

This structure contains the SLQSNASSwiGetChannelLock response parameters.

**Parameters**

| | |
|---|---|
| *pWcdmaUARF-CN* | [Optional]<br>• See wcdmaUARFCN for more information |
| *pLteEARFCN* | [Optional]<br>• See lteEARFCN for more information |
| *pLtePCI* | [Optional]<br>• See ltePCI for more information |

### 8.574.2 Field Documentation

#### 8.574.2.1 lteEARFCN∗ nasSwiGetChannelLockResp::pLteEARFCN

#### 8.574.2.2 ltePCI∗ nasSwiGetChannelLockResp::pLtePCI

#### 8.574.2.3 wcdmaUARFCN∗ nasSwiGetChannelLockResp::pWcdmaUARFCN

## 8.575 NasSwiIndReg Struct Reference

### Data Fields

- BYTE lteEsmUl
- BYTE lteEsmDl
- BYTE lteEmmUl
- BYTE lteEmmDl
- BYTE gsmUmtsUl
- BYTE gsmUmtsDl
- BYTE ∗ pRankIndicatorInd
- BYTE ∗ pTimer

### 8.575.1 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 |

### 8.575.2 Field Documentation

#### 8.575.2.1 BYTE NasSwiIndReg::gsmUmtsDl

#### 8.575.2.2 BYTE NasSwiIndReg::gsmUmtsUl

#### 8.575.2.3 BYTE NasSwiIndReg::lteEmmDl

#### 8.575.2.4 BYTE NasSwiIndReg::lteEmmUl

#### 8.575.2.5 BYTE NasSwiIndReg::lteEsmDl

#### 8.575.2.6 BYTE NasSwiIndReg::lteEsmUl

#### 8.575.2.7 BYTE* NasSwiIndReg::pRankIndicatorInd

#### 8.575.2.8 BYTE* NasSwiIndReg::pTimer

## 8.576 nasSwiSetChannelLockReq Struct Reference

### Data Fields

- wcdmaUARFCN * pWcdmaUARFCN
- lteEARFCN * pLteEARFCN
- ltePCI * pLtePCI

### 8.576.1 Detailed Description

This structure contains the SLQSNASSwiSetChannelLock response parameters.

**Parameters**

| | |
|---|---|
| *pWcdmaUARFCN* | [Optional]<br>• See wcdmaUARCFN for more information |
| *pLteEARFCN* | [Optional]<br>• See lteEARCFN for more information |
| *pLtePCI* | [Optional]<br>• See ltePCI for more information |

### 8.576.2 Field Documentation

#### 8.576.2.1 lteEARFCN* nasSwiSetChannelLockReq::pLteEARFCN

#### 8.576.2.2 ltePCI∗ nasSwiSetChannelLockReq::pLtePCI

#### 8.576.2.3 wcdmaUARFCN∗ nasSwiSetChannelLockReq::pWcdmaUARFCN

## 8.577 nasSysInfo Struct Reference

### Data Fields

- SrvStatusInfo ∗ pCDMASrvStatusInfo
- SrvStatusInfo ∗ pHDRSrvStatusInfo
- GSMSrvStatusInfo ∗ pGSMSrvStatusInfo
- GSMSrvStatusInfo ∗ pWCDMASrvStatusInfo
- GSMSrvStatusInfo ∗ pLTESrvStatusInfo
- CDMASysInfo ∗ pCDMASysInfo
- HDRSysInfo ∗ pHDRSysInfo
- GSMSysInfo ∗ pGSMSysInfo
- WCDMASysInfo ∗ pWCDMASysInfo
- LTESysInfo ∗ pLTESysInfo
- AddCDMASysInfo ∗ pAddCDMASysInfo
- WORD ∗ pAddHDRSysInfo
- AddSysInfo ∗ pAddGSMSysInfo
- AddSysInfo ∗ pAddWCDMASysInfo
- WORD ∗ pAddLTESysInfo
- CallBarringSysInfo ∗ pGSMCallBarringSysInfo
- CallBarringSysInfo ∗ pWCDMACallBarringSysInfo
- BYTE ∗ pLTEVoiceSupportSysInfo
- BYTE ∗ pGSMCipherDomainSysInfo
- BYTE ∗ pWCDMACipherDomainSysInfo
- BYTE ∗ pSysInfoNoChange

### 8.577.1 Detailed Description

Structure for storing the nasSysInfo indication parameters.

**Parameters**

| | |
|---|---|
| *pCDMASrvStatusInfo* | • See SrvStatusInfo for more information. |
| *pHDRSrvStatusInfo* | • See SrvStatusInfo for more information. |
| *pGSMSrvStatusInfo* | • See GSMSrvStatusInfo for more information. |
| *pWCDMASrvStatusInfo* | • See GSMSrvStatusInfo for more information. |
| *pLTESrvStatusInfo* | • See GSMSrvStatusInfo for more information. |
| *pCDMASysInfo* | • See CDMASysInfo for more information. |

| | |
|---|---|
| *pHDRSysInfo* | • See HDRSysInfo for more information. |
| *pGSMSysInfo* | • See GSMSysInfo for more information. |
| *pWCDMASysInfo* | • See WCDMASysInfo for more information. |
| *pLTESysInfo* | • See LTESysInfo for more information. |
| *pAddCDMASysInfo* | • See AddCDMASysInfo for more information. |
| *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. |
| *pAddGSMSysInfo* | • See AddSysInfo for more information. |
| *pAddWCDMASysInfo* | • See AddSysInfo for more information. |
| *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. |
| *pGSMCallBarringSysInfo* | • See CallBarringSysInfo for more information. |
| *pWCDMACallBarringSysInfo* | • See CallBarringSysInfo for more information. |
| *pLTEVoiceSupportSysInfo* | • Indicates voice support status on LTE.<br>– 0x00 - Voice is not supported<br>– 0x01 - Voice is supported |
| *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 |

| | |
|---|---|
| *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 |
| *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 |

### 8.577.2 Field Documentation

#### 8.577.2.1 AddCDMASysInfo* nasSysInfo::pAddCDMASysInfo

#### 8.577.2.2 AddSysInfo* nasSysInfo::pAddGSMSysInfo

#### 8.577.2.3 WORD* nasSysInfo::pAddHDRSysInfo

#### 8.577.2.4 WORD* nasSysInfo::pAddLTESysInfo

#### 8.577.2.5 AddSysInfo* nasSysInfo::pAddWCDMASysInfo

#### 8.577.2.6 SrvStatusInfo* nasSysInfo::pCDMASrvStatusInfo

#### 8.577.2.7 CDMASysInfo* nasSysInfo::pCDMASysInfo

#### 8.577.2.8 CallBarringSysInfo* nasSysInfo::pGSMCallBarringSysInfo

#### 8.577.2.9 BYTE* nasSysInfo::pGSMCipherDomainSysInfo

#### 8.577.2.10 GSMSrvStatusInfo* nasSysInfo::pGSMSrvStatusInfo

#### 8.577.2.11 GSMSysInfo* nasSysInfo::pGSMSysInfo

#### 8.577.2.12 SrvStatusInfo* nasSysInfo::pHDRSrvStatusInfo

#### 8.577.2.13 HDRSysInfo* nasSysInfo::pHDRSysInfo

#### 8.577.2.14 GSMSrvStatusInfo* nasSysInfo::pLTESrvStatusInfo

#### 8.577.2.15 LTESysInfo* nasSysInfo::pLTESysInfo

#### 8.577.2.16 BYTE* nasSysInfo::pLTEVoiceSupportSysInfo

#### 8.577.2.17 BYTE* nasSysInfo::pSysInfoNoChange

#### 8.577.2.18 CallBarringSysInfo* nasSysInfo::pWCDMACallBarringSysInfo

#### 8.577.2.19 BYTE* nasSysInfo::pWCDMACipherDomainSysInfo

#### 8.577.2.20 GSMSrvStatusInfo* nasSysInfo::pWCDMASrvStatusInfo

#### 8.577.2.21 WCDMASysInfo* nasSysInfo::pWCDMASysInfo

## 8.578 NASTimeInfoTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t time

### 8.578.1 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 |

### 8.578.2 Field Documentation

#### 8.578.2.1 uint64_t NASTimeInfoTlv::time

#### 8.578.2.2 uint8_t NASTimeInfoTlv::TlvPresent

## 8.579 nasTimers Struct Reference

### Data Fields

- char t3396_apn [101]
- BYTE t3396_plmn_id [3]
- ULONG t3396_val

### 8.579.1 Detailed Description

This structure contains Network Timer

**Parameters**

| | |
|---|---|
| *t3396_apn* | • APN name |
| *t3396_plmn_id* | • PLMN ID |
| *t3396_val* | • T3396 timer value |

### 8.579.2 Field Documentation

#### 8.579.2.1 char nasTimers::t3396_apn[101]

#### 8.579.2.2 BYTE nasTimers::t3396_plmn_id[3]

#### 8.579.2.3 ULONG nasTimers::t3396_val

## 8.580 netSelectionPref Struct Reference

### Data Fields

- BYTE netReg
- WORD mcc
- WORD mnc

### 8.580.1 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 |

### 8.580.2 Field Documentation

#### 8.580.2.1 WORD netSelectionPref::mcc

#### 8.580.2.2 WORD netSelectionPref::mnc

#### 8.580.2.3 BYTE netSelectionPref::netReg

## 8.581 NetStats Struct Reference

### Data Fields

- ULONG rx_packets
- ULONG tx_packets
- ULONGLONG rx_bytes
- ULONGLONG tx_bytes
- ULONG rx_errors

- ULONG tx_errors
- ULONG rx_overflows
- ULONG tx_overflows

### 8.581.1 Detailed Description

This structure contains the SLQSGetNetStatistic Information

**Parameters**

| | |
|---|---|
| *rx_packets* | • Number of received Packets without error |
| *tx_packets* | • Number of transmitted Packets without error |
| *rx_bytes* | • Number of bytes recieved without error |
| *tx_bytes* | • NNumbero of bytes transmitted without error |
| *rx_error* | • Number of incoming packets with framing errors |
| *tx_error* | • Number of outgoing packets with framing errors |
| *rx_overflows* | • Number of packets dropped because Rx buffer overflowed |
| *tx_overflows* | • Number of packets dropped because Tx buffer overflowed |

### 8.581.2 Field Documentation

#### 8.581.2.1 ULONGLONG NetStats::rx_bytes

#### 8.581.2.2 ULONG NetStats::rx_errors

#### 8.581.2.3 ULONG NetStats::rx_overflows

#### 8.581.2.4 ULONG NetStats::rx_packets

#### 8.581.2.5 ULONGLONG NetStats::tx_bytes

#### 8.581.2.6 ULONG NetStats::tx_errors

#### 8.581.2.7 ULONG NetStats::tx_overflows

#### 8.581.2.8 ULONG NetStats::tx_packets

## 8.582 NetworkDebugResp Struct Reference

### Data Fields

- BYTE * pObjectVer
- NetworkStat1x * pNetworkStat1x
- NetworkStatEVDO * pNetworkStatEVDO
- DeviceConfigDetail * pDeviceConfigDetail
- DataStatusDetail * pDataStatusDetail

### 8.582.1 Detailed Description

This structure contains information about the SLQSSwiNetworkDebug response parameters.

**Parameters**

| | |
|---|---|
| *pObjectVer* | • Object's version number for the host to handle<br>– 0xFF - NA<br>– Others - shows in decimal |
| *pNetworkStat1x* | • See NetworkStat1x for more information |
| *pNetworkStatEVDO* | • See NetworkStatEVDO for more information. |
| *pDeviceConfigDetail* | • See DeviceConfigDetail for more information. |
| *pDataStatusDetail* | • See DataStatusDetail for more information. |

### 8.582.2 Field Documentation

#### 8.582.2.1 DataStatusDetail* NetworkDebugResp::pDataStatusDetail

#### 8.582.2.2 DeviceConfigDetail* NetworkDebugResp::pDeviceConfigDetail

#### 8.582.2.3 NetworkStat1x* NetworkDebugResp::pNetworkStat1x

#### 8.582.2.4 NetworkStatEVDO* NetworkDebugResp::pNetworkStatEVDO

#### 8.582.2.5 BYTE* NetworkDebugResp::pObjectVer

## 8.583 NetworkStat1x Struct Reference

### Data Fields

- BYTE State
- WORD SO
- ULONG RX_PWR

- WORD RX_EC_IO
- ULONG TX_PWR
- BYTE ActSetCnt
- ActPilotPNElement * pActPilotPNElements
- BYTE NeighborSetCnt
- WORD * pNeighborSetPilotPN

### 8.583.1 Detailed Description

This structure contains 1x network status details

**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(* | IN/OUT )<br>• 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 ActPilotPNElement for more information |
| *NeighborSetCnt(* | IN/OUT )<br>• 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 |

## 8.583.2 Field Documentation

### 8.583.2.1 BYTE NetworkStat1x::ActSetCnt

### 8.583.2.2 BYTE NetworkStat1x::NeighborSetCnt

### 8.583.2.3 ActPilotPNElement* NetworkStat1x::pActPilotPNElements

### 8.583.2.4 WORD* NetworkStat1x::pNeighborSetPilotPN

### 8.583.2.5 WORD NetworkStat1x::RX_EC_IO

### 8.583.2.6 ULONG NetworkStat1x::RX_PWR

### 8.583.2.7 WORD NetworkStat1x::SO

### 8.583.2.8 BYTE NetworkStat1x::State

#### 8.583.2.9 ULONG NetworkStat1x::TX_PWR

## 8.584 NetworkStatEVDO Struct Reference

### Data Fields

- BYTE State
- BYTE MACIndex
- BYTE SectorIDLen
- WORD * pSectorID
- WORD RX_PWR
- WORD PER
- WORD PilotEnergy
- BYTE SNR

### 8.584.1 Detailed Description

This structure contains EVDO network status details

**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) |

### 8.584.2 Field Documentation

#### 8.584.2.1 BYTE NetworkStatEVDO::MACIndex

#### 8.584.2.2 WORD NetworkStatEVDO::PER

#### 8.584.2.3 WORD NetworkStatEVDO::PilotEnergy

#### 8.584.2.4 WORD* NetworkStatEVDO::pSectorID

#### 8.584.2.5 WORD NetworkStatEVDO::RX_PWR

#### 8.584.2.6 BYTE NetworkStatEVDO::SectorIDLen

#### 8.584.2.7 BYTE NetworkStatEVDO::SNR

#### 8.584.2.8 BYTE NetworkStatEVDO::State

## 8.585 newMTMessageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSMTMessageInfo MTMessageInfo

### 8.585.1 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 |

### 8.585.2 Field Documentation

#### 8.585.2.1 sMSMTMessageInfo newMTMessageTlv::MTMessageInfo

#### 8.585.2.2 uint8_t newMTMessageTlv::TlvPresent

## 8.586 newPwdData Struct Reference

### Data Fields

- BYTE newPwd [4]
- BYTE newPwdAgain [4]

### 8.586.1 Detailed Description

This structure contains New Password Data.

**Parameters**

| | |
|---|---|
| *newPwd[PASSWORD_LENGTH]* | • New password.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |
| *newPwdAgain[PASSWORD_LENGTH]* | • New password again.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |

### 8.586.2 Field Documentation

#### 8.586.2.1 BYTE newPwdData::newPwd[4]

#### 8.586.2.2 BYTE newPwdData::newPwdAgain[4]

## 8.587 nmrCellInfo Struct Reference

### Data Fields

- ULONG nmrCellID
- BYTE nmrPlmn [3]
- WORD nmrLac
- WORD nmrArfcn
- BYTE nmrBsic
- WORD nmrRxLev

### 8.587.1 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[PLMN-_LENGTH]* | • 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 |

### 8.587.2 Field Documentation

#### 8.587.2.1 WORD nmrCellInfo::nmrArfcn

#### 8.587.2.2 BYTE nmrCellInfo::nmrBsic

#### 8.587.2.3 ULONG nmrCellInfo::nmrCellID

#### 8.587.2.4 WORD nmrCellInfo::nmrLac

#### 8.587.2.5 BYTE nmrCellInfo::nmrPlmn[3]

#### 8.587.2.6 WORD nmrCellInfo::nmrRxLev

## 8.588 nr5gBandPref Struct Reference

### Data Fields

- ULONGLONG bits_1_64
- ULONGLONG bits_65_128
- ULONGLONG bits_129_192
- ULONGLONG bits_193_256

### 8.588.1 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. |

### 8.588.2 Field Documentation

#### 8.588.2.1 ULONGLONG nr5gBandPref::bits_129_192

#### 8.588.2.2 ULONGLONG nr5gBandPref::bits_193_256

#### 8.588.2.3 ULONGLONG nr5gBandPref::bits_1_64

#### 8.588.2.4 ULONGLONG nr5gBandPref::bits_65_128

## 8.589 NSSAudioCtrl Struct Reference

### Data Fields

- BYTE upLink
- BYTE downLink

### 8.589.1 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]. |

### 8.589.2 Field Documentation

#### 8.589.2.1 BYTE NSSAudioCtrl::downLink

#### 8.589.2.2 BYTE NSSAudioCtrl::upLink

## 8.590 NWProfile Struct Reference

### Data Fields

- WORD tech
- BYTE * pProfSz
- WORD * pProfValues

### 8.590.1 Detailed Description

This structure contains Network supported QoS profile

**Parameters**

| | |
|---|---|
| *tech* | Technology on which the network supported QoS profiles are being returned:<br>• CDMA – 0x8001 |
| *exponent* | |

### 8.590.2 Field Documentation

#### 8.590.2.1 BYTE* NWProfile::pProfSz

#### 8.590.2.2 WORD* NWProfile::pProfValues

#### 8.590.2.3 WORD NWProfile::tech

## 8.591 omaDmConfigTlv Struct Reference

### Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- USHORT alertmsglength
- BYTE alertmsg [256]

### 8.591.1 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) |

### 8.591.2 Field Documentation

#### 8.591.2.1 BYTE omaDmConfigTlv::alertmsg[256]

#### 8.591.2.2 USHORT omaDmConfigTlv::alertmsglength

#### 8.591.2.3 BYTE omaDmConfigTlv::state

#### 8.591.2.4 BYTE omaDmConfigTlv::userInputReq

#### 8.591.2.5 USHORT omaDmConfigTlv::userInputTimeout

## 8.592 omaDmConfigTlvExt Struct Reference

### Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- USHORT alertmsglength
- BYTE alertmsg [256]

### 8.592.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

**Parameters**

| | |
|---|---|
| *state* | • 1 - reserved<br>• 2 - reserved<br>• 3 - reserved<br>• 4 - CI DC Success<br>• 5 - CI DC Failure<br>• 6 - User/device initiated PRL update success.<br>• 7 - User/device initiated PRL update failure.<br>• 8 - HFA DC session start<br>• 9 - HFA DC sucess.<br>• 10 - HFA is cancelled.<br>• 11 - HFA retry. UI Screen 13[1] with 0 percent progress bar should be shown.<br>• 12 - HFA fail after 5 retries. UI Screen 2[1] should be displayed.<br>• 13 - HFA retry down counter. Used to update the process bar of UI Screen 13[1].<br>• 14 - HFA PRL session start, UI screen 4[1] should be displayed.<br>• 15 - HFA PRL update success.<br>• 16 - Device is launching a NI session. UI Screen 1[1] should be displayed.<br>• 17 - An empty session. UI Screen 2[1] should be displayed.<br>• 18 - No network coverage.<br>• 19 - HFA is not enabled.<br>• 20 - CI DC Start, UI Screen 1[1] should be displayed.<br>• 21 - CI PRL start, UI screen 4[1] should be displayed.<br>• 22 - HFA PRL updates fail.<br>• 23 - Device reboot.<br>• 24 - CI DC is cancelled.<br>• 25 - User/device initiated PRL update is cancelled.<br>• 26 - NI session is cancelled.<br>• 27 - Current NI session is not enabled.<br>• 28 - NI DC success.<br>• 29 - NI DC Fail.<br>• 30 - NI PRL success<br>• 31 - NI PRL fail.<br>• 32 - Reserved<br>• 33 - NI fumo fail<br>• 34 - NI session fail, unable to point out the session type. |
| *user_input_req* | - OMA task stop to wait user's input if this field is valid. until user input selection or after "UI Timer out (next field). In the case of timeout, a default selection of "YES/OK" is accepted. Note that this option is valid when DM state is 4/6/12/28/30. 0 - user/host doesn't need to input anything, and OMA task doesn't blocked by UI. 1 - user/host must input "YES/OK/CANCEL". 2 - User/host must input "NO/CANCEL". 3 - user/host must input "YES/OK/NO/CANCEL. |
| *user_input_timeout* | • Timeout for user input in seconds. This indicates how many seconds OMA task stop to wait for host/user's response. |
| *alertmsglength* | • Length of Alert message string in word(16-bit) |

| | |
|---|---|
| *alertmsg* | • Alert message in UCS2 (Max 256 characters)<br>• This string is printed by host |

### 8.592.2 Field Documentation

#### 8.592.2.1 BYTE omaDmConfigTlvExt::alertmsg[256]

#### 8.592.2.2 USHORT omaDmConfigTlvExt::alertmsglength

#### 8.592.2.3 BYTE omaDmConfigTlvExt::state

#### 8.592.2.4 BYTE omaDmConfigTlvExt::userInputReq

#### 8.592.2.5 USHORT omaDmConfigTlvExt::userInputTimeout

## 8.593 omaDmFotaTlv Struct Reference

### Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- ULONG fwdloadsize
- ULONG fwloadComplete
- USHORT updateCompleteStatus
- BYTE severity
- USHORT versionlength
- BYTE version [256]
- USHORT namelength
- BYTE package_name [256]
- USHORT descriptionlength
- BYTE description [256]
- BYTE sessionType

### 8.593.1 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 |

### 8.593.2 Field Documentation

#### 8.593.2.1 BYTE omaDmFotaTlv::description[256]

#### 8.593.2.2 USHORT omaDmFotaTlv::descriptionlength

#### 8.593.2.3 ULONG omaDmFotaTlv::fwdloadsize

#### 8.593.2.4 ULONG omaDmFotaTlv::fwloadComplete

#### 8.593.2.5 USHORT omaDmFotaTlv::namelength

#### 8.593.2.6 BYTE omaDmFotaTlv::package_name[256]

#### 8.593.2.7 BYTE omaDmFotaTlv::sessionType

#### 8.593.2.8 BYTE omaDmFotaTlv::severity

#### 8.593.2.9 BYTE omaDmFotaTlv::state

#### 8.593.2.10 USHORT omaDmFotaTlv::updateCompleteStatus

#### 8.593.2.11 BYTE omaDmFotaTlv::userInputReq

#### 8.593.2.12 USHORT omaDmFotaTlv::userInputTimeout

#### 8.593.2.13 BYTE omaDmFotaTlv::version[256]

#### 8.593.2.14 USHORT omaDmFotaTlv::versionlength

## 8.594 omaDmFotaTlvExt Struct Reference

### Data Fields

- BYTE state
- BYTE reserved
- USHORT userInputTimeout
- ULONG packageSize
- ULONG receivedBytes
- USHORT fumoResultCode
- USHORT versionlength
- BYTE version [256]
- USHORT namelength
- BYTE package_name [256]
- USHORT descriptionlength
- BYTE description [256]

### 8.594.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

**Parameters**

| | |
|---|---|
| *state* | • 0x01 - No Firmware available<br>• 0x02 - reserved<br>• 0x03 - Update progress bar, UI screen 7[1] should be displayed<br>• 0x04 - reserved<br>• 0x05 - FUMO image download success, UI screen 8[1] should be displayed<br>• 0x06 - reserved<br>• 0x07 - FUMO image installation result, used to return error code.<br>• 0x08 - FUMO session start<br>• 0x09 - UI Screen 7[1] with 0 percent bar should be displayed<br>• 0x0A - FUMO image installation is cancelled by user.<br>• 0x0B - FUMO session fail<br>• 0x0C - Device is sending a report to OMA Server.<br>• 0x0D - Report to server success.<br>• 0x0E - Report to server fails.<br>• 0x0F - reserved<br>• 0x10 - FUMO session is cancelled before image download success.<br>• 0x11 - UI Screen 16[1] should be displayed, FUMO delay option, OMA task is blocked until a valid response is recevied. |
| *reserved* | - For sprint reserved |
| *user_input_timeout* | • How many seconds OMA task stop to wait for user/host response. |
| *packageSize* | • The size (in bytes) of the firmware update package (only valid for states 3/5/7). |
| *receivedBytes* | • The number of bytes downloaded. Useful for FUMO state 3. |
| *fumoResultCode* | • Used when fumo state is 7/11. REsult code of FUMO image installation<br>– 200 image install success<br>• Others: image install fail |
| *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) |

### 8.594.2 Field Documentation

#### 8.594.2.1 BYTE omaDmFotaTlvExt::description[256]

#### 8.594.2.2 USHORT omaDmFotaTlvExt::descriptionlength

#### 8.594.2.3 USHORT omaDmFotaTlvExt::fumoResultCode

#### 8.594.2.4 USHORT omaDmFotaTlvExt::namelength

#### 8.594.2.5 BYTE omaDmFotaTlvExt::package_name[256]

#### 8.594.2.6 ULONG omaDmFotaTlvExt::packageSize

#### 8.594.2.7 ULONG omaDmFotaTlvExt::receivedBytes

#### 8.594.2.8 BYTE omaDmFotaTlvExt::reserved

#### 8.594.2.9 BYTE omaDmFotaTlvExt::state

#### 8.594.2.10 USHORT omaDmFotaTlvExt::userInputTimeout

#### 8.594.2.11 BYTE omaDmFotaTlvExt::version[256]

#### 8.594.2.12 USHORT omaDmFotaTlvExt::versionlength

## 8.595 omaDmNotificationsTlv Struct Reference

### Data Fields

- BYTE notification
- USHORT sessionStatus

### 8.595.1 Detailed Description

This structure will hold the SwiOmaDmConfig session 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. |

### 8.595.2 Field Documentation

#### 8.595.2.1 BYTE omaDmNotificationsTlv::notification

#### 8.595.2.2 USHORT omaDmNotificationsTlv::sessionStatus

## 8.596 operatorNameString Struct Reference

### Data Fields

- BYTE PLMNName [255]

### 8.596.1 Detailed Description

This structure contains Operator Name String as defined in CPHS4_2.WW6(Feb 27, 1997) (Section B.4.1.2)from multiple sources.

**Parameters**

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

### 8.596.2 Field Documentation

#### 8.596.2.1 BYTE operatorNameString::PLMNName[255]

## 8.597 OperatorPLMNData Struct Reference

### Data Fields

- BYTE mcc [3]
- BYTE mnc [3]
- WORD lac1
- WORD lac2
- BYTE PLMNRecID

### 8.597.1 Detailed Description

This structure contains Operator PLMN Data from multiple sources.

**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. |

### 8.597.2 Field Documentation

#### 8.597.2.1 WORD OperatorPLMNData::lac1

#### 8.597.2.2 WORD OperatorPLMNData::lac2

#### 8.597.2.3 BYTE OperatorPLMNData::mcc[3]

#### 8.597.2.4 BYTE OperatorPLMNData::mnc[3]

#### 8.597.2.5 BYTE OperatorPLMNData::PLMNRecID

## 8.598 operatorPLMNList Struct Reference

### Data Fields

- WORD numInstance
- OperatorPLMNData PLMNData [255]

### 8.598.1 Detailed Description

This structure contains Operator PLMN List as defined in 3GPP TS 31.102 (Section 4.2.59) from multiple sources.

**Parameters**

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

### 8.598.2 Field Documentation

#### 8.598.2.1 WORD operatorPLMNList::numInstance

#### 8.598.2.2 OperatorPLMNData operatorPLMNList::PLMNData[255]

## 8.599 pack_dms_ActivateAutomatic_t Struct Reference

### Data Fields

- uint8_t actCode [81]

### 8.599.1 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" |

### 8.599.2 Field Documentation

#### 8.599.2.1 uint8_t pack_dms_ActivateAutomatic_t::actCode[81]

## 8.600 pack_dms_GetCustFeaturesV2_t Struct Reference

### Data Fields

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

### 8.600.1 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 |

### 8.600.2 Field Documentation

#### 8.600.2.1 uint8_t pack_dms_GetCustFeaturesV2_t::cust_id[64+1]

#### 8.600.2.2 uint8_t pack_dms_GetCustFeaturesV2_t::list_type

#### 8.600.2.3 uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult

## 8.601 pack_dms_ResetToFactoryDefaults_t Struct Reference

### Data Fields

- uint8_t spc [6]

### 8.601.1 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) |

### 8.601.2 Field Documentation

#### 8.601.2.1 uint8_t pack_dms_ResetToFactoryDefaults_t::spc[6]

## 8.602 pack_dms_SetActivationStatusCallback_t Struct Reference

### Data Fields

- uint8_t activationState

### 8.602.1 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 |

### 8.602.2 Field Documentation

#### 8.602.2.1 uint8_t pack_dms_SetActivationStatusCallback_t::activationState

## 8.603 pack_dms_SetCrashAction_t Struct Reference

### Data Fields

- uint8_t crashAction

### 8.603.1 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 |

### 8.603.2 Field Documentation

#### 8.603.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

## 8.604 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

### 8.604.1 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 |

### 8.604.2 Field Documentation

#### 8.604.2.1 uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled

#### 8.604.2.2 uint8_t pack_dms_SetCustFeature_t::DisableIMSI

#### 8.604.2.3 uint32_t pack_dms_SetCustFeature_t::GpsEnable

#### 8.604.2.4 uint8_t pack_dms_SetCustFeature_t::GPSLPM

#### 8.604.2.5 uint8_t pack_dms_SetCustFeature_t::GPSSel

#### 8.604.2.6 uint16_t pack_dms_SetCustFeature_t::IPFamSupport

#### 8.604.2.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

#### 8.604.2.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

#### 8.604.2.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

## 8.605 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

### 8.605.1 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 |

### 8.605.2 Field Documentation

#### 8.605.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

#### 8.605.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

#### 8.605.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

#### 8.605.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

## 8.606 pack_dms_SetEventReport_t Struct Reference

### Data Fields

- uint8_t mode

### 8.606.1 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 |

### 8.606.2 Field Documentation

#### 8.606.2.1 uint8_t pack_dms_SetEventReport_t::mode

## 8.607 pack_dms_SetIndicationRegister_t Struct Reference

### Data Fields

- uint8_t PSMStatus
- uint8_t PSMCfgChangeInfo
- uint8_t RptIMSCapability

### 8.607.1 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 |

### 8.607.2 Field Documentation

#### 8.607.2.1 uint8_t pack_dms_SetIndicationRegister_t::PSMCfgChangeInfo

#### 8.607.2.2 uint8_t pack_dms_SetIndicationRegister_t::PSMStatus

#### 8.607.2.3 uint8_t pack_dms_SetIndicationRegister_t::RptIMSCapability

## 8.608 pack_dms_SetPower_t Struct Reference

### Data Fields

- uint32_t mode
- uint16_t Tlvresult

### 8.608.1 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. |

### 8.608.2 Field Documentation

#### 8.608.2.1 uint32_t pack_dms_SetPower_t::mode

#### 8.608.2.2 uint16_t pack_dms_SetPower_t::Tlvresult

## 8.609 pack_dms_SetUSBComp_t Struct Reference

### Data Fields

- uint8_t USBComp
- uint16_t Tlvresult

### 8.609.1 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. |

### 8.609.2 Field Documentation

#### 8.609.2.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

#### 8.609.2.2 uint8_t pack_dms_SetUSBComp_t::USBComp

## 8.610 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

### Data Fields

- uint8_t resetInfoInd

### 8.610.1 Detailed Description

This structure contains set registration state for different indication pack

**Parameters**

| | |
|---|---|
| *resetInfoInd[IN]* | • Values<br>– 0 - Disable<br>– 1 - Enable |

### 8.610.2 Field Documentation

#### 8.610.2.1 uint8_t pack_dms_SLQSDmsSwiIndicationRegister_t::resetInfoInd

## 8.611 pack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference

### Data Fields

- uint8_t * pPsmEnableState
- uint32_t * pActiveTimer
- uint32_t * pPeriodicUpdateTimer

### 8.611.1 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. |

### 8.611.2 Field Documentation

#### 8.611.2.1 uint32_t* pack_dms_SLQSSetPowerSaveModeConfig_t::pActiveTimer

#### 8.611.2.2 uint32_t* pack_dms_SLQSSetPowerSaveModeConfig_t::pPeriodicUpdateTimer

#### 8.611.2.3 uint8_t* pack_dms_SLQSSetPowerSaveModeConfig_t::pPsmEnableState

## 8.612 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference

### Data Fields

- uint8_t clear

### 8.612.1 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 |

### 8.612.2 Field Documentation

#### 8.612.2.1 uint8_t pack_dms_SLQSSwiGetCrashInfo_t::clear

## 8.613 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

### Data Fields

- uint8_t * pDestSMSNum
- uint8_t * pDestSMSContent

### 8.613.1 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. |
| *pDestSMS-Content[IN]* | • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars.<br>• Optional parameter. |

### 8.613.2 Field Documentation

#### 8.613.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

#### 8.613.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

## 8.614 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

### Data Fields

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

### 8.614.1 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 |

### 8.614.2 Field Documentation

#### 8.614.2.1 char pack_dms_SLQSSwiSetHostDevInfo_t::manString[255]

#### 8.614.2.2 char pack_dms_SLQSSwiSetHostDevInfo_t::modelString[255]

#### 8.614.2.3 char pack_dms_SLQSSwiSetHostDevInfo_t::plasmaIDString[255]

#### 8.614.2.4 char pack_dms_SLQSSwiSetHostDevInfo_t::swVerString[255]

## 8.615 pack_dms_SLQSSwiSetOSInfo_t Struct Reference

### Data Fields

- char nameString [255]
- char versionString [255]

### 8.615.1 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. |

### 8.615.2 Field Documentation

#### 8.615.2.1 char pack_dms_SLQSSwiSetOSInfo_t::nameString[255]

#### 8.615.2.2 char pack_dms_SLQSSwiSetOSInfo_t::versionString[255]

## 8.616 pack_dms_SwiSetEventReport_t Struct Reference

### Data Fields

- uint8_t * pTempReport
- uint8_t * pVoltReport

### 8.616.1 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 |

### 8.616.2 Field Documentation

#### 8.616.2.1 uint8_t* pack_dms_SwiSetEventReport_t::pTempReport

#### 8.616.2.2 uint8_t* pack_dms_SwiSetEventReport_t::pVoltReport

## 8.617 pack_dms_UIMChangePIN_t Struct Reference

### Data Fields

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

### 8.617.1 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 |

### 8.617.2 Field Documentation

#### 8.617.2.1 uint8_t pack_dms_UIMChangePIN_t::id

#### 8.617.2.2 uint8_t pack_dms_UIMChangePIN_t::newValue[255]

#### 8.617.2.3 uint8_t pack_dms_UIMChangePIN_t::oldValue[255]

## 8.618 pack_dms_UIMGetControlKeyStatus_t Struct Reference

### Data Fields

- uint8_t facility

### 8.618.1 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) |

### 8.618.2 Field Documentation

#### 8.618.2.1 uint8_t pack_dms_UIMGetControlKeyStatus_t::facility

## 8.619 pack_dms_UIMGetICCID_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.619.1 Detailed Description

This structure contains UIM Get ICCID pack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Pack result. |

### 8.619.2 Field Documentation

#### 8.619.2.1 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

## 8.620 pack_dms_UIMSetControlKeyProtection_t Struct Reference

### Data Fields

- uint8_t facility
- uint8_t facilityState
- uint8_t facilityCk [8]

### 8.620.1 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) |

### 8.620.2 Field Documentation

#### 8.620.2.1 uint8_t pack_dms_UIMSetControlKeyProtection_t::facility

#### 8.620.2.2 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityCk[8]

#### 8.620.2.3 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityState

## 8.621 pack_dms_UIMSetPINProtection_t Struct Reference

### Data Fields

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

### 8.621.1 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 |

### 8.621.2 Field Documentation

#### 8.621.2.1 uint8_t pack_dms_UIMSetPINProtection_t::bEnable

#### 8.621.2.2 uint8_t pack_dms_UIMSetPINProtection_t::id

#### 8.621.2.3 uint8_t pack_dms_UIMSetPINProtection_t::value[255]

## 8.622 pack_dms_UIMUnblockControlKey_t Struct Reference

### Data Fields

- uint8_t facility
- uint8_t facilityCk [8]

### 8.622.1 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) |

### 8.622.2 Field Documentation

#### 8.622.2.1 uint8_t pack_dms_UIMUnblockControlKey_t::facility

#### 8.622.2.2 uint8_t pack_dms_UIMUnblockControlKey_t::facilityCk[8]

## 8.623 pack_dms_UIMUnblockPIN_t Struct Reference

### Data Fields

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

### 8.623.1 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 |

### 8.623.2 Field Documentation

#### 8.623.2.1 uint8_t pack_dms_UIMUnblockPIN_t::id

#### 8.623.2.2 uint8_t pack_dms_UIMUnblockPIN_t::newPin[255]

#### 8.623.2.3 uint8_t pack_dms_UIMUnblockPIN_t::pukValue[255]

## 8.624 pack_dms_UIMVerifyPIN_t Struct Reference

### Data Fields

- uint8_t id
- uint8_t value [255]

### 8.624.1 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 |

### 8.624.2 Field Documentation

#### 8.624.2.1 uint8_t pack_dms_UIMVerifyPIN_t::id

#### 8.624.2.2 uint8_t pack_dms_UIMVerifyPIN_t::value[255]

## 8.625 pack_fms_GetImagesPreference_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.625.1 Detailed Description

This structure contains the Get Image Preference information pack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Pack result |

### 8.625.2 Field Documentation

#### 8.625.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

## 8.626 pack_fms_GetStoredImages_t Struct Reference

### Data Fields

- uint16_t TlvResult

### 8.626.1 Detailed Description

This structure contains the Get Stored Images pack

**Parameters**

| | |
|---|---|
| *TlvResult* | • Pack result |

### 8.626.2 Field Documentation

#### 8.626.2.1 uint16_t pack_fms_GetStoredImages_t::TlvResult

## 8.627 pack_fms_SetImagesPreference_t Struct Reference

### Data Fields

- uint32_t imageListSize
- FMSPrefImageList * pImageList
- uint32_t bForceDownload
- uint8_t modemindex
- uint16_t TlvResult

### 8.627.1 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 |

### 8.627.2 Field Documentation

#### 8.627.2.1 uint32_t pack_fms_SetImagesPreference_t::bForceDownload

#### 8.627.2.2 uint32_t pack_fms_SetImagesPreference_t::imageListSize

#### 8.627.2.3 uint8_t pack_fms_SetImagesPreference_t::modemindex

#### 8.627.2.4 FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList

#### 8.627.2.5 uint16_t pack_fms_SetImagesPreference_t::Tlvresult

## 8.628 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

### 8.628.1 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. |

### 8.628.2 Field Documentation

#### 8.628.2.1 loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo

#### 8.628.2.2 loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb

#### 8.628.2.3 loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo

#### 8.628.2.4 loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData

#### 8.628.2.5 loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo

#### 8.628.2.6 uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult

## 8.629 pack_loc_EventRegister_t Struct Reference

### Data Fields

- uint64_t eventRegister
- uint16_t Tlvresult

### 8.629.1 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. |

### 8.629.2 Field Documentation

#### 8.629.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

#### 8.629.2.2 uint16_t pack_loc_EventRegister_t::TlvResult

## 8.630 pack_loc_SetExtPowerState_t Struct Reference

### Data Fields

- uint32_t extPowerState
- uint16_t TlvResult

### 8.630.1 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. |

### 8.630.2 Field Documentation

#### 8.630.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

#### 8.630.2.2 uint16_t pack_loc_SetExtPowerState_t::TlvResult

## 8.631 pack_loc_SetOperationMode_t Struct Reference

### Data Fields

- uint32_t mode
- uint16_t TlvResult

### 8.631.1 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. |

### 8.631.2 Field Documentation

#### 8.631.2.1 uint32_t pack_loc_SetOperationMode_t::mode

#### 8.631.2.2 uint16_t pack_loc_SetOperationMode_t::Tlvresult

## 8.632 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

### Data Fields

- uint32_t xid
- uint16_t Tlvresult

### 8.632.1 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. |

### 8.632.2 Field Documentation

#### 8.632.2.1 uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

#### 8.632.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

## 8.633 pack_loc_SLQSLOCGetServer_t Struct Reference

### Data Fields

- uint32_t serverType
- int has_serverAddrTypeMask
- uint8_t serverAddrTypeMask

### 8.633.1 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 |

### 8.633.2 Field Documentation

#### 8.633.2.1 int pack_loc_SLQSLOCGetServer_t::has_serverAddrTypeMask

#### 8.633.2.2 uint8_t pack_loc_SLQSLOCGetServer_t::serverAddrTypeMask

#### 8.633.2.3 uint32_t pack_loc_SLQSLOCGetServer_t::serverType

## 8.634 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

### 8.634.1 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. |

### 8.634.2 Field Documentation

#### 8.634.2.1 altSrcInfo_t pack_loc_SLQSLOCInjectPosition_t::altitudeSrcInfo

#### 8.634.2.2 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtEllipsoid

#### 8.634.2.3 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtMeanSeaLevel

#### 8.634.2.4 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeSrcInfo

#### 8.634.2.5 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtEllipsoid

#### 8.634.2.6 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtMeanSeaLevel

#### 8.634.2.7 int pack_loc_SLQSLOCInjectPosition_t::has_horConfidence

#### 8.634.2.8 int pack_loc_SLQSLOCInjectPosition_t::has_horReliability

#### 8.634.2.9 int pack_loc_SLQSLOCInjectPosition_t::has_horUncCircular

#### 8.634.2.10 int pack_loc_SLQSLOCInjectPosition_t::has_latitude

#### 8.634.2.11 int pack_loc_SLQSLOCInjectPosition_t::has_longitude

#### 8.634.2.12 int pack_loc_SLQSLOCInjectPosition_t::has_positionSrc

#### 8.634.2.13 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorConfidence

#### 8.634.2.14 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorUncCircular

#### 8.634.2.15 int pack_loc_SLQSLOCInjectPosition_t::has_timestampAge

#### 8.634.2.16 int pack_loc_SLQSLOCInjectPosition_t::has_timestampUtc

#### 8.634.2.17 int pack_loc_SLQSLOCInjectPosition_t::has_vertConfidence

#### 8.634.2.18 int pack_loc_SLQSLOCInjectPosition_t::has_vertRelicability

#### 8.634.2.19 int pack_loc_SLQSLOCInjectPosition_t::has_vertUnc

#### 8.634.2.20 uint8_t pack_loc_SLQSLOCInjectPosition_t::horConfidence

#### 8.634.2.21 uint32_t pack_loc_SLQSLOCInjectPosition_t::horReliability

#### 8.634.2.22 float pack_loc_SLQSLOCInjectPosition_t::horUncCircular

#### 8.634.2.23 double pack_loc_SLQSLOCInjectPosition_t::latitude

#### 8.634.2.24 double pack_loc_SLQSLOCInjectPosition_t::longitude

#### 8.634.2.25 uint32_t pack_loc_SLQSLOCInjectPosition_t::positionSrc

#### 8.634.2.26 uint8_t pack_loc_SLQSLOCInjectPosition_t::rawHorConfidence

#### 8.634.2.27 float pack_loc_SLQSLOCInjectPosition_t::rawHorUncCircular

#### 8.634.2.28 uint32_t pack_loc_SLQSLOCInjectPosition_t::timestampAge

#### 8.634.2.29 uint64_t pack_loc_SLQSLOCInjectPosition_t::timestampUtc

#### 8.634.2.30 uint8_t pack_loc_SLQSLOCInjectPosition_t::vertConfidence

#### 8.634.2.31 uint32_t pack_loc_SLQSLOCInjectPosition_t::vertReliability

#### 8.634.2.32 float pack_loc_SLQSLOCInjectPosition_t::vertUnc

## 8.635 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

### 8.635.1 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 |

## 8.635.2 Field Documentation

### 8.635.2.1 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData

### 8.635.2.2 uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc

### 8.635.2.3 tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp

### 8.635.2.4 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::gyroData

#### 8.635.2.5 tempData_t pack_loc_SLQSLOCInjectSensorData_t::gyroTemp

#### 8.635.2.6 uint32_t pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc

#### 8.635.2.7 int pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc

#### 8.635.2.8 int pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp

#### 8.635.2.9 int pack_loc_SLQSLOCInjectSensorData_t::has_accleroData

#### 8.635.2.10 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroData

#### 8.635.2.11 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp

#### 8.635.2.12 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc

#### 8.635.2.13 int pack_loc_SLQSLOCInjectSensorData_t::has_opaqueId

#### 8.635.2.14 uint32_t pack_loc_SLQSLOCInjectSensorData_t::opaqueId

## 8.636 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference

### Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec

### 8.636.1 Detailed Description

This structure contains inject UTC time parameter.

**Parameters**

| | |
|---|---|
| *timeMsec* | • The UTC time since Jan. 1, 1970 |
| *timeUncMsec* | • The time Uncertainty |

### 8.636.2 Field Documentation

#### 8.636.2.1 uint64_t pack_loc_SLQSLOCInjectUTCTime_t::timeMsec

#### 8.636.2.2 uint32_t pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec

## 8.637 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference

### Data Fields

- uint32_t state
- int has_confidence
- uint8_t confidence

### 8.637.1 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 |

### 8.637.2 Field Documentation

#### 8.637.2.1 uint8_t pack_loc_SLQSLOCSetCradleMountConfig_t::confidence

#### 8.637.2.2 int pack_loc_SLQSLOCSetCradleMountConfig_t::has_confidence

#### 8.637.2.3 uint32_t pack_loc_SLQSLOCSetCradleMountConfig_t::state

## 8.638 pack_loc_SLQSLOCSetServer_t Struct Reference

### Data Fields

- uint32_t serverType
- loc_IPv4Config * pIPv4Config
- loc_IPv6Config * pIPv6Config
- loc_URLAddrInfo * pURLAddr

### 8.638.1 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 |

### 8.638.2 Field Documentation

#### 8.638.2.1 loc_IPv4Config* pack_loc_SLQSLOCSetServer_t::pIPv4Config

#### 8.638.2.2 loc_IPv6Config* pack_loc_SLQSLOCSetServer_t::pIPv6Config

#### 8.638.2.3 loc_URLAddrInfo* pack_loc_SLQSLOCSetServer_t::pURLAddr

#### 8.638.2.4 uint32_t pack_loc_SLQSLOCSetServer_t::serverType

## 8.639 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

### 8.639.1 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. |

### 8.639.2 Field Documentation

#### 8.639.2.1 loc_LocApplicationInfo* pack_loc_Start_t::pApplicationInfo

#### 8.639.2.2 uint32_t* pack_loc_Start_t::pConfigAltitudeAssumed

#### 8.639.2.3 uint32_t* pack_loc_Start_t::pHorizontalAccuracyLvl

#### 8.639.2.4 uint32_t* pack_loc_Start_t::pIntermediateReportState

#### 8.639.2.5 uint32_t* pack_loc_Start_t::pMinIntervalTime

#### 8.639.2.6 uint32_t* pack_loc_Start_t::pRecurrenceType

#### 8.639.2.7 uint8_t pack_loc_Start_t::SessionId

#### 8.639.2.8 uint16_t pack_loc_Start_t::Tlvresult

## 8.640 pack_loc_Stop_t Struct Reference

### Data Fields

- uint8_t SessionId
- uint16_t Tlvresult

### 8.640.1 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. |

### 8.640.2 Field Documentation

#### 8.640.2.1 uint8_t pack_loc_Stop_t::SessionId

#### 8.640.2.2 uint16_t pack_loc_Stop_t::Tlvresult

## 8.641 pack_nas_InitiateDomainAttach_t Struct Reference

### Data Fields

- uint32_t action

### 8.641.1 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 |

### 8.641.2 Field Documentation

#### 8.641.2.1 uint32_t pack_nas_InitiateDomainAttach_t::action

## 8.642 pack_nas_SetACCOLC_t Struct Reference

### Data Fields

- int8_t spc [6]
- uint8_t accolc

### 8.642.1 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 |

### 8.642.2 Field Documentation

#### 8.642.2.1 uint8_t pack_nas_SetACCOLC_t::accolc

#### 8.642.2.2 int8_t pack_nas_SetACCOLC_t::spc[6]

## 8.643 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

### 8.643.1 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 |

## 8.643.2 Field Documentation

### 8.643.2.1 uint32_t* pack_nas_SetCDMANetworkParameters_t::pApplication

### 8.643.2.2 uint32_t* pack_nas_SetCDMANetworkParameters_t::pBroadcast

### 8.643.2.3 uint8_t* pack_nas_SetCDMANetworkParameters_t::pCustomSCP

### 8.643.2.4 uint8_t* pack_nas_SetCDMANetworkParameters_t::pForceRev0

### 8.643.2.5 uint32_t* pack_nas_SetCDMANetworkParameters_t::pProtocol

### 8.643.2.6 uint32_t* pack_nas_SetCDMANetworkParameters_t::pRoaming

#### 8.643.2.7 char* pack_nas_SetCDMANetworkParameters_t::pSPC

## 8.644 pack_nas_SetNetworkPreference_t Struct Reference

### Data Fields

- uint32_t TechnologyPref
- uint32_t Duration
- uint16_t Tlvresult

### 8.644.1 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 |

### 8.644.2 Field Documentation

#### 8.644.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

#### 8.644.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

#### 8.644.2.3 uint16_t pack_nas_SetNetworkPreference_t::Tlvresult

## 8.645 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

## 8.645.1 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 |

### 8.645.2 Field Documentation

#### 8.645.2.1 nas_ECIOThresh∗ pack_nas_SLQSConfigSigInfo_t::pECIOThresh

#### 8.645.2.2 nas_HDRSINRThresh∗ pack_nas_SLQSConfigSigInfo_t::pHDRSINRThresh

#### 8.645.2.3 nas_IOThresh∗ pack_nas_SLQSConfigSigInfo_t::pIOThresh

#### 8.645.2.4 nas_LTESigRptCfg∗ pack_nas_SLQSConfigSigInfo_t::pLTESigRptCfg

#### 8.645.2.5 nas_LTESNRThresh∗ pack_nas_SLQSConfigSigInfo_t::pLTESNRThresh

#### 8.645.2.6 nas_RSRPThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRPThresh

#### 8.645.2.7 nas_RSRQThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRQThresh

#### 8.645.2.8 nas_RSSIThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSSIThresh

#### 8.645.2.9 nas_TDSCDMASINRCONFThresh∗ pack_nas_SLQSConfigSigInfo_t::pTDSCDMASINRCONFThresh

## 8.646 pack_nas_SLQSGetPLMNName_t Struct Reference

### Data Fields

- uint16_t mcc
- uint16_t mnc
- uint8_t ∗ pMncPcsStatus

### 8.646.1 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.

### 8.646.2 Field Documentation

#### 8.646.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

#### 8.646.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

#### 8.646.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

## 8.647 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference

### Data Fields

- uint32_t regAction
- nas_MNRInfo * pMNRInfo
- uint32_t * pChangeDuration
- uint8_t * pMncPcsDigitStatus

### 8.647.1 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. |

### 8.647.2 Field Documentation

#### 8.647.2.1 uint32_t* pack_nas_SLQSInitiateNetworkRegistration_t::pChangeDuration

#### 8.647.2.2 uint8_t* pack_nas_SLQSInitiateNetworkRegistration_t::pMncPcsDigitStatus

#### 8.647.2.3 nas_MNRInfo* pack_nas_SLQSInitiateNetworkRegistration_t::pMNRInfo

#### 8.647.2.4 uint32_t pack_nas_SLQSInitiateNetworkRegistration_t::regAction

## 8.648 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

### 8.648.1 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.

### 8.648.2 Field Documentation

#### 8.648.2.1 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta

#### 8.648.2.2 nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh

#### 8.648.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta

#### 8.648.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh

#### 8.648.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta

#### 8.648.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh

#### 8.648.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta

#### 8.648.2.8 nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh

#### 8.648.2.9 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta

#### 8.648.2.10 nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh

#### 8.648.2.11 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIDelta

#### 8.648.2.12 nas_HDRRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIThresh

#### 8.648.2.13 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRDelta

#### 8.648.2.14 nas_HDRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRThresh

8.648.2.15 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta

8.648.2.16 nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh

8.648.2.17 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta

8.648.2.18 nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh

8.648.2.19 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta

8.648.2.20 nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh

8.648.2.21 nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig

8.648.2.22 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta

8.648.2.23 nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh

8.648.2.24 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta

8.648.2.25 nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh

8.648.2.26 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta

8.648.2.27 nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh

8.648.2.28 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta

8.648.2.29 nas_TDSCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh

8.648.2.30 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta

8.648.2.31 nas_TDSCDMASINRThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh

8.648.2.32 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIODelta

8.648.2.33 nas_WCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh

8.648.2.34 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta

8.648.2.35 nas_WCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh

## 8.649 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference

### Data Fields

- uint8_t namID

### 8.649.1 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. |

### 8.649.2 Field Documentation

#### 8.649.2.1 uint8_t pack_nas_SLQSNasGet3GPP2Subscription_t::namID

## 8.650 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference

### Data Fields

- uint8_t radio_if

### 8.650.1 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 |

### 8.650.2 Field Documentation

#### 8.650.2.1 uint8_t pack_nas_SLQSNasGetTxRxInfo_t::radio_if

## 8.651 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

### 8.651.1 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 |

## 8.651.2 Field Documentation

### 8.651.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.651.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.651.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.651.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

8.651.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

8.651.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

8.651.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

8.651.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkRejectInd

8.651.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

8.651.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

8.651.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

8.651.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

8.651.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSuppressSysInfoInd

8.651.2.14 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

8.651.2.15 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

## 8.652 pack_nas_SLQSNASSeteDRXParams_t Struct Reference

### Data Fields

- uint8_t * pEdrxEnable
- uint8_t * pCycleLen
- uint8_t * pPagingTimeWindow

### 8.652.1 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 |

### 8.652.2 Field Documentation

#### 8.652.2.1 uint8_t* pack_nas_SLQSNASSeteDRXParams_t::pCycleLen

#### 8.652.2.2 uint8_t* pack_nas_SLQSNASSeteDRXParams_t::pEdrxEnable

#### 8.652.2.3 uint8_t* pack_nas_SLQSNASSeteDRXParams_t::pPagingTimeWindow

## 8.653 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

### 8.653.1 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 |

### 8.653.2 Field Documentation

#### 8.653.2.1 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsDl

#### 8.653.2.2 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsUl

#### 8.653.2.3 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmDl

#### 8.653.2.4 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmUl

#### 8.653.2.5 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmDl

#### 8.653.2.6 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmUl

#### 8.653.2.7 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pRankIndicatorInd

#### 8.653.2.8 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pTimer

## 8.654 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference

### Data Fields

- nas_wcdmaUARFCN * pWcdmaUARFCN
- nas_lteEARFCN * pLteEARFCN
- nas_ltePCI * pLtePCI

### 8.654.1 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 |

### 8.654.2 Field Documentation

#### 8.654.2.1 nas_lteEARFCN∗ pack_nas_SLQSNASSwiSetChannelLock_t::pLteEARFCN

#### 8.654.2.2 nas_ltePCI∗ pack_nas_SLQSNASSwiSetChannelLock_t::pLtePCI

#### 8.654.2.3 nas_wcdmaUARFCN∗ pack_nas_SLQSNASSwiSetChannelLock_t::pWcdmaUARFCN

## 8.655 pack_nas_SLQSSetBandPreference_t Struct Reference

### Data Fields

- uint64_t bandPref

### 8.655.1 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 |

### 8.655.2 Field Documentation

#### 8.655.2.1 uint64_t pack_nas_SLQSSetBandPreference_t::bandPref

## 8.656 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

### Data Fields

- uint8_t bEnable
- nas_SLQSSignalStrengthsIndReq * pSigIndReq

### 8.656.1 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 |

### 8.656.2 Field Documentation

#### 8.656.2.1 uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable

#### 8.656.2.2 nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq

## 8.657 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

### 8.657.1 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 |

### 8.657.2 Field Documentation

#### 8.657.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

#### 8.657.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

#### 8.657.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

#### 8.657.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

#### 8.657.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

#### 8.657.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

#### 8.657.2.7 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref

#### 8.657.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat

#### 8.657.2.9 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref

#### 8.657.2.10 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref

#### 8.657.2.11 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref

#### 8.657.2.12 unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT

#### 8.657.2.13 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref

#### 8.657.2.14 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref

#### 8.657.2.15 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction

#### 8.657.2.16 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdscdmaBandPref

## 8.658 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
- uint64_t * pCiotLteOpModePref
- uint64_t * pLteM1BandPref
- uint64_t * pLteNb1BandPref
- uint32_t * pCiotLteOpMode
- struct nas_ciotAcqOrderPref * pCiotAcqOrderPref
- struct nas_nr5gBandPref * pNr5gBandPref

### 8.658.1 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 |
| *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 |

## 8.658.2 Field Documentation

### 8.658.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pAcqOrderPref

### 8.658.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pBandPref

### 8.658.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pChgDuration

### 8.658.2.4 struct nas_ciotAcqOrderPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotAcqOrderPref

### 8.658.2.5 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotLteOpMode

### 8.658.2.6 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotLteOpModePref

### 8.658.2.7 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPrefExt_t::pCSGID

### 8.658.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pEmerMode

### 8.658.2.9 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pGWAcqOrderPref

### 8.658.2.10 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLTEBandPref

### 8.658.2.11 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLteM1BandPref

### 8.658.2.12 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLteNb1BandPref

### 8.658.2.13 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pMNCIncPCSDigStat

### 8.658.2.14 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pModePref

### 8.658.2.15 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pNetSelPref

### 8.658.2.16 struct nas_nr5gBandPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pNr5gBandPref

### 8.658.2.17 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pPRLPref

#### 8.658.2.18 unsigned char* pack_nas_SLQSSetSysSelectionPrefExt_t::pRAT

#### 8.658.2.19 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pRoamPref

#### 8.658.2.20 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pSrvDomainPref

#### 8.658.2.21 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pSrvRegRestriction

#### 8.658.2.22 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pTdscdmaBandPref

## 8.659 pack_nas_SLQSSwiPSDetach_t Struct Reference

### Data Fields

- uint8_t * pDetachAction

### 8.659.1 Detailed Description

Structure for PS connection detach pack.

**Parameters**

| | |
|---|---|
| *pDetachAction[IN]* | • Values<br>– 2- Initiates an immediate packet domain detach. |

### 8.659.2 Field Documentation

#### 8.659.2.1 uint8_t* pack_nas_SLQSSwiPSDetach_t::pDetachAction

## 8.660 pack_pds_PDSInjectTimeReference_t Struct Reference

### Data Fields

- uint64_t systemTime
- uint16_t systemDiscontinuities

### 8.660.1 Detailed Description

Structure contain the parameter for PDS Inject time.

**Parameters**

| | |
|---|---|
| *systemTime* | • System time( milliseconds ) |
| *systemDiscontinuities* | • Number of system time discontinuities |

### 8.660.2 Field Documentation

#### 8.660.2.1 uint16_t pack_pds_PDSInjectTimeReference_t::systemDiscontinuities

#### 8.660.2.2 uint64_t pack_pds_PDSInjectTimeReference_t::systemTime

## 8.661 pack_pds_ResetPDSData_t Struct Reference

### Data Fields

- uint32_t * pGPSDataMask
- uint32_t * pCellDataMask

### 8.661.1 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 |

### 8.661.2 Field Documentation

#### 8.661.2.1 uint32_t* pack_pds_ResetPDSData_t::pCellDataMask

#### 8.661.2.2 uint32_t* pack_pds_ResetPDSData_t::pGPSDataMask

## 8.662 pack_pds_SetEventReportCallback_t Struct Reference

### Data Fields

- uint8_t posDataNmea
- uint8_t rptPosData

### 8.662.1 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 |

### 8.662.2 Field Documentation

#### 8.662.2.1 uint8_t pack_pds_SetEventReportCallback_t::posDataNmea

#### 8.662.2.2 uint8_t pack_pds_SetEventReportCallback_t::rptPosData

## 8.663 pack_pds_SetPDSDefaults_t Struct Reference

### Data Fields

- uint32_t operation
- uint8_t timeout
- uint32_t interval
- uint32_t accuracy

### 8.663.1 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) |

### 8.663.2 Field Documentation

#### 8.663.2.1 uint32_t pack_pds_SetPDSDefaults_t::accuracy

#### 8.663.2.2 uint32_t pack_pds_SetPDSDefaults_t::interval

#### 8.663.2.3 uint32_t pack_pds_SetPDSDefaults_t::operation

#### 8.663.2.4 uint8_t pack_pds_SetPDSDefaults_t::timeout

## 8.664 pack_pds_SetPDSState_t Struct Reference

### Data Fields

- uint32_t enable

### 8.664.1 Detailed Description

Structure contain PDS state parameters.

**Parameters**

| | |
|---|---|
| *enable* | • Desired PDS state<br>– Zero - disable<br>– Non-Zero - enable |

### 8.664.2 Field Documentation

#### 8.664.2.1 uint32_t pack_pds_SetPDSState_t::enable

## 8.665 pack_pds_SetPortAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t bAuto

#### 8.665.1 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 |

#### 8.665.2 Field Documentation

##### 8.665.2.1 uint32_t pack_pds_SetPortAutomaticTracking_t::bAuto

## 8.666 pack_pds_SetServiceAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t bAuto

#### 8.666.1 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 |

#### 8.666.2 Field Documentation

##### 8.666.2.1 uint32_t pack_pds_SetServiceAutomaticTracking_t::bAuto

## 8.667 pack_pds_SetXTRAAutomaticDownload_t Struct Reference

### Data Fields

- uint32_t bEnabled
- uint16_t interval

### 8.667.1 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 |

### 8.667.2 Field Documentation

#### 8.667.2.1 uint32_t pack_pds_SetXTRAAutomaticDownload_t::bEnabled

#### 8.667.2.2 uint16_t pack_pds_SetXTRAAutomaticDownload_t::interval

## 8.668 pack_pds_SetXTRANetwork_t Struct Reference

### Data Fields

- uint32_t preference

### 8.668.1 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 |

### 8.668.2 Field Documentation

#### 8.668.2.1 uint32_t pack_pds_SetXTRANetwork_t::preference

## 8.669 pack_pds_SLQSGetAGPSConfig_t Struct Reference

### Data Fields

- uint8_t * pNetworkMode

### 8.669.1 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 |

### 8.669.2 Field Documentation

#### 8.669.2.1 uint8_t* pack_pds_SLQSGetAGPSConfig_t::pNetworkMode

## 8.670 pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference

### Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec
- uint8_t timeBase
- uint8_t forceFlag

### 8.670.1 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 |

### 8.670.2 Field Documentation

#### 8.670.2.1 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::forceFlag

#### 8.670.2.2 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeBase

#### 8.670.2.3 uint64_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeMsec

#### 8.670.2.4 uint32_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeUncMsec

## 8.671 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

### 8.671.1 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 |

### 8.671.2 Field Documentation

#### 8.671.2.1 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtEllipsoid

#### 8.671.2.2 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtSealevel

#### 8.671.2.3 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalConfidence

#### 8.671.2.4 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalUncCircular

#### 8.671.2.5 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLatitude

#### 8.671.2.6 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLongitude

#### 8.671.2.7 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pPositionSource

#### 8.671.2.8 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeStamp

#### 8.671.2.9 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeType

#### 8.671.2.10 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalConfidence

#### 8.671.2.11 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalUnc

## 8.672 pack_pds_SLQSSetAGPSConfig_t Struct Reference

### Data Fields

- uint32_t * pServerAddress
- uint32_t * pServerPort
- uint8_t * pServerURL
- uint8_t * pServerURLLength
- uint8_t * pNetworkMode

### 8.672.1 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 |

### 8.672.2 Field Documentation

#### 8.672.2.1 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pNetworkMode

#### 8.672.2.2 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerAddress

#### 8.672.2.3 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerPort

#### 8.672.2.4 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURL

#### 8.672.2.5 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURLLength

## 8.673 pack_pds_SLQSSetPositionMethodState_t Struct Reference

### Data Fields

- uint8_t * pXtraTimeState

- uint8_t * pXtraDataState
- uint8_t * pWifiState

### 8.673.1 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 |

### 8.673.2 Field Documentation

#### 8.673.2.1 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pWifiState

#### 8.673.2.2 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraDataState

#### 8.673.2.3 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraTimeState

## 8.674 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

### 8.674.1 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) |

### 8.674.2 Field Documentation

#### 8.674.2.1 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixAccuracy

#### 8.674.2.2 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixCount

#### 8.674.2.3 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixInterval

#### 8.674.2.4 uint8_t pack_pds_StartPDSTrackingSessionExt_t::fixTimeout

#### 8.674.2.5 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionControl

#### 8.674.2.6 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionOperation

#### 8.674.2.7 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionServerOption

#### 8.674.2.8 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionType

## 8.675 pack_qmi_t Struct Reference

### Data Fields

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

### 8.675.1 Detailed Description

qmi request context

**Parameters**

| | | |
|---|---|---|
| in | *xid* | transaction id |
| out | *timeout* | timeout recommended in seconds |
| out | *msgid* | message id |
| out | *svc* | qmi service |

### 8.675.2 Field Documentation

#### 8.675.2.1 uint16_t pack_qmi_t::msgid

#### 8.675.2.2 uint8_t pack_qmi_t::svc

#### 8.675.2.3 int pack_qmi_t::timeout

#### 8.675.2.4 uint16_t pack_qmi_t::xid

## 8.676 pack_qos_BindDataPort_t Struct Reference

### Data Fields

- qos_BindDataPortPeripheralEndPointID_t * pPeripheralEndPointID
- qos_BindDataPortMuxID_t * pMuxID
- qos_BindDataPortSIODataPort_t * pSIODataPort

### 8.676.1 Detailed Description

Structure that contains the request Binds a control point to a data port.

**Parameters**

| | |
|---|---|
| *pPeripheralEnd-PointID* | • See qos_BindDataPortPeripheralEndPointID_t for more information. |
| *pMuxID* | • See qos_BindDataPortMuxID_t for more information. |
| *pSIODataPort* | • See qos_BindDataPortSIODataPort_t for more information. |

### 8.676.2 Field Documentation

#### 8.676.2.1 qos_BindDataPortMuxID_t* pack_qos_BindDataPort_t::pMuxID

#### 8.676.2.2 qos_BindDataPortPeripheralEndPointID_t* pack_qos_BindDataPort_t::pPeripheralEndPointID

#### 8.676.2.3 qos_BindDataPortSIODataPort_t* pack_qos_BindDataPort_t::pSIODataPort

## 8.677 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

### Data Fields

- uint32_t apnId

### 8.677.1 Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

**Parameters**

| | |
|---|---|
| *apnId[IN]* | • APN id |

### 8.677.2 Field Documentation

#### 8.677.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

## 8.678 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

### Data Fields

- uint32_t apnId

### 8.678.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

**Parameters**

| | |
|---|---|
| *apnId[IN]* | • APN id |

### 8.678.2 Field Documentation

#### 8.678.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

## 8.679 pack_qos_SLQSSetQosEventCallback_t Struct Reference

### Data Fields

- uint8_t enable

### 8.679.1 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 |

### 8.679.2 Field Documentation

#### 8.679.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

## 8.680 pack_rms_SetSMSWake_t Struct Reference

### Data Fields

- uint32_t enabled
- uint32_t wake_mask

### 8.680.1 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) |

### 8.680.2 Field Documentation

#### 8.680.2.1 uint32_t pack_rms_SetSMSWake_t::enabled

#### 8.680.2.2 uint32_t pack_rms_SetSMSWake_t::wake_mask

## 8.681 pack_sar_SLQSSetRfSarState_t Struct Reference

### Data Fields

- uint32_t RfSarState

### 8.681.1 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 |

### 8.681.2 Field Documentation

#### 8.681.2.1 uint32_t pack_sar_SLQSSetRfSarState_t::RfSarState

## 8.682 pack_sms_SaveSMS_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage

### 8.682.1 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 |

### 8.682.2 Field Documentation

#### 8.682.2.1 uint32_t pack_sms_SaveSMS_t::messageFormat

#### 8.682.2.2 uint32_t pack_sms_SaveSMS_t::messageSize

#### 8.682.2.3 uint8_t* pack_sms_SaveSMS_t::pMessage

#### 8.682.2.4 uint32_t pack_sms_SaveSMS_t::storageType

## 8.683 pack_sms_SendSMS_t Struct Reference

### Data Fields

- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage
- uint8_t * pLinktimer

### 8.683.1 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 |

### 8.683.2 Field Documentation

#### 8.683.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

#### 8.683.2.2 uint32_t pack_sms_SendSMS_t::messageSize

#### 8.683.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

#### 8.683.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

## 8.684 pack_sms_SetNewSMSCallback_t Struct Reference

### Data Fields

- enum eqmiCbkSetStatus status

### 8.684.1 Detailed Description

This strucure contains pack set new SMS callback parameters.

**Parameters**

| | |
|---|---|
| *status* | callback parameter |

### 8.684.2 Field Documentation

#### 8.684.2.1 enum eqmiCbkSetStatus pack_sms_SetNewSMSCallback_t::status

## 8.685 pack_sms_SetSMSCAddress_t Struct Reference

### Data Fields

- uint8_t * pSMSCAddress
- uint8_t * pSMSCType

### 8.685.1 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). |

### 8.685.2 Field Documentation

#### 8.685.2.1 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCAddress

#### 8.685.2.2 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCType

## 8.686 pack_sms_SLQSDeleteSMS_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t * pMessageIndex
- uint32_t * pMessageTag
- uint8_t * pMessageMode

### 8.686.1 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) |

### 8.686.2 Field Documentation

#### 8.686.2.1 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageIndex

#### 8.686.2.2 uint8_t* pack_sms_SLQSDeleteSMS_t::pMessageMode

#### 8.686.2.3 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageTag

#### 8.686.2.4 uint32_t pack_sms_SLQSDeleteSMS_t::storageType

## 8.687 pack_sms_SLQSGetSMS_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint8_t * pMessageMode

### 8.687.1 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) |

### 8.687.2 Field Documentation

#### 8.687.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

#### 8.687.2.2 uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode

#### 8.687.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

## 8.688 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

### Data Fields

- uint8_t mode

### 8.688.1 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) |

### 8.688.2 Field Documentation

#### 8.688.2.1 uint8_t pack_sms_SLQSGetSmsBroadcastConfig_t::mode

## 8.689 pack_sms_SLQSGetSMSList_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t * pRequestedTag
- uint8_t * pMessageMode

### 8.689.1 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) |

### 8.689.2 Field Documentation

#### 8.689.2.1 uint8_t* pack_sms_SLQSGetSMSList_t::pMessageMode

#### 8.689.2.2 uint32_t* pack_sms_SLQSGetSMSList_t::pRequestedTag

#### 8.689.2.3 uint32_t pack_sms_SLQSGetSMSList_t::storageType

## 8.690 pack_sms_SLQSModifySMSStatus_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint32_t messageTag
- uint8_t * pMessageMode

### 8.690.1 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) |

### 8.690.2 Field Documentation

#### 8.690.2.1 uint32_t pack_sms_SLQSModifySMSStatus_t::messageIndex

#### 8.690.2.2 uint32_t pack_sms_SLQSModifySMSStatus_t::messageTag

#### 8.690.2.3 uint8_t* pack_sms_SLQSModifySMSStatus_t::pMessageMode

#### 8.690.2.4 uint32_t pack_sms_SLQSModifySMSStatus_t::storageType

## 8.691 pack_sms_SLQSSendAsyncSMS_t Struct Reference

### Data Fields

- sms_sendAsyncsmsParams * pSendSmsParams

### 8.691.1 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 |

### 8.691.2 Field Documentation

#### 8.691.2.1 sms_sendAsyncsmsParams* pack_sms_SLQSSendAsyncSMS_t::pSendSmsParams

## 8.692 pack_sms_SLQSSetIndicationRegister_t Struct Reference

### Data Fields

- sms_setIndicationReg * pSetIndicationRegReq

### 8.692.1 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 |

### 8.692.2 Field Documentation

#### 8.692.2.1 sms_setIndicationReg* pack_sms_SLQSSetIndicationRegister_t::pSetIndicationRegReq

## 8.693 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference

### Data Fields

- uint8_t mode
- uint8_t broadcastActivate

### 8.693.1 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 |

### 8.693.2 Field Documentation

#### 8.693.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::broadcastActivate

#### 8.693.2.2 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::mode

## 8.694 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference

### Data Fields

- uint8_t mode
- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

### 8.694.1 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). |

### 8.694.2 Field Documentation

#### 8.694.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastConfig_t::mode

#### 8.694.2.2 sms_qaQmi3GPPBroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pBroadcastConfig

#### 8.694.2.3 sms_qaQmi3GPP2BroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pCDMABroadcastConfig

## 8.695 pack_sms_SLQSSetSmsStorage_t Struct Reference

### Data Fields

- uint8_t smsStorage

### 8.695.1 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 |

### 8.695.2 Field Documentation

#### 8.695.2.1 uint8_t pack_sms_SLQSSetSmsStorage_t::smsStorage

## 8.696 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

### Data Fields

- sms_maxStorageSizeReq * pMaxStorageSizeReq

### 8.696.1 Detailed Description

This structure contains get maximum storage size.

**Parameters**

| | |
|---|---|
| *pMaxStorage-SizeReq* | • Request parameters for SmsSLQSGetMaxStorageSize<br>– See sms_maxStorageSizeReq for more information |

### 8.696.2 Field Documentation

#### 8.696.2.1 sms_maxStorageSizeReq* pack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeReq

## 8.697 pack_sms_SLQSSmsSetRoutes_t Struct Reference

### Data Fields

- sms_setRoutesReq * pSetRoutesReq

### 8.697.1 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 |

### 8.697.2 Field Documentation

#### 8.697.2.1 sms_setRoutesReq* pack_sms_SLQSSmsSetRoutes_t::pSetRoutesReq

## 8.698 pack_swidms_SLQSSwiDmsSetMTU_t Struct Reference

### Data Fields

- uint16_t MTUSize

### 8.698.1 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. |

### 8.698.2 Field Documentation

#### 8.698.2.1 uint16_t pack_swidms_SLQSSwiDmsSetMTU_t::MTUSize

## 8.699 pack_swidms_SLQSSwiDmsSetUsbNetNum_t Struct Reference

### Data Fields

- uint8_t nUsbNetNum

### 8.699.1 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 |

### 8.699.2 Field Documentation

#### 8.699.2.1 uint8_t pack_swidms_SLQSSwiDmsSetUsbNetNum_t::nUsbNetNum

## 8.700 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

### 8.700.1 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 |

### 8.700.2 Field Documentation

#### 8.700.2.1 uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate

#### 8.700.2.2 uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type

#### 8.700.2.3 uint8_t pack_swiloc_SwiLocSetAutoStart_t::function

#### 8.700.2.4 uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist

#### 8.700.2.5 uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time

#### 8.700.2.6 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate

#### 8.700.2.7 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type

#### 8.700.2.8 int pack_swiloc_SwiLocSetAutoStart_t::set_function

#### 8.700.2.9 int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist

#### 8.700.2.10 int pack_swiloc_SwiLocSetAutoStart_t::set_max_time

## 8.701 pack_swioma_SLQSOMADMCancelSession_t Struct Reference

### Data Fields

- uint32_t sessionType

### 8.701.1 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 |

### 8.701.2 Field Documentation

#### 8.701.2.1 uint32_t pack_swioma_SLQSOMADMCancelSession_t::sessionType

## 8.702 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference

### Data Fields

- uint32_t SessionType

### 8.702.1 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 |

### 8.702.2 Field Documentation

#### 8.702.2.1 uint32_t pack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

## 8.703 pack_swioma_SLQSOMADMSendSelection_t Struct Reference

### Data Fields

- uint32_t selection
- uint32_t * pDeferTime
- uint32_t * pRejectReason

### 8.703.1 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. |

### 8.703.2 Field Documentation

#### 8.703.2.1 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pDeferTime

#### 8.703.2.2 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pRejectReason

#### 8.703.2.3 uint32_t pack_swioma_SLQSOMADMSendSelection_t::selection

## 8.704 pack_swioma_SLQSOMADMSetSettings_t Struct Reference

### Data Fields

- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t * pAutosdm
- uint8_t * pFwAutoCheck

### 8.704.1 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 |

### 8.704.2 Field Documentation

#### 8.704.2.1 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAdownload

#### 8.704.2.2 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAUpdate

#### 8.704.2.3 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pAutosdm

#### 8.704.2.4 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pFwAutoCheck

## 8.705 pack_swioma_SLQSOMADMStartSession_t Struct Reference

### Data Fields

- uint32_t sessionType

### 8.705.1 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 |

### 8.705.2 Field Documentation

#### 8.705.2.1 uint32_t pack_swioma_SLQSOMADMStartSession_t::sessionType

## 8.706 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

### 8.706.1 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.

### 8.706.2 Field Documentation

#### 8.706.2.1 uim_changeUIMPIN pack_uim_ChangePin_t::changePIN

#### 8.706.2.2 uim_encryptedPIN1 pack_uim_ChangePin_t::EncryptedPIN1

#### 8.706.2.3 uint32_t* pack_uim_ChangePin_t::pIndicationToken

#### 8.706.2.4 uint8_t* pack_uim_ChangePin_t::pKeyReferenceID

#### 8.706.2.5 uim_sessionInformation pack_uim_ChangePin_t::sessionInfo

#### 8.706.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

## 8.707 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

### 8.707.1 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.

### 8.707.2 Field Documentation

#### 8.707.2.1 uim_fileInfo pack_uim_ReadTransparent_t::fileIndex

#### 8.707.2.2 uint8_t* pack_uim_ReadTransparent_t::pEncryptData

#### 8.707.2.3 uint32_t* pack_uim_ReadTransparent_t::pIndicationToken

#### 8.707.2.4 uim_readTransparentInfo pack_uim_ReadTransparent_t::readTransparent

#### 8.707.2.5 uim_sessionInformation pack_uim_ReadTransparent_t::sessionInfo

#### 8.707.2.6 uint16_t pack_uim_ReadTransparent_t::Tlvresult

## 8.708 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

### 8.708.1 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. |
| *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 |

**Note**

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

### 8.708.2 Field Documentation

#### 8.708.2.1 uim_encryptedPIN1 pack_uim_SetPinProtection_t::EncryptedPIN1

#### 8.708.2.2 uint32_t* pack_uim_SetPinProtection_t::pIndicationToken

#### 8.708.2.3 uim_setPINProtection pack_uim_SetPinProtection_t::pinProtection

#### 8.708.2.4 uint8_t* pack_uim_SetPinProtection_t::pKeyReferenceID

#### 8.708.2.5 uim_sessionInformation pack_uim_SetPinProtection_t::sessionInfo

#### 8.708.2.6 uint16_t pack_uim_SetPinProtection_t::Tlvresult

## 8.709 pack_uim_SLQSUIMAuthenticate_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uim_authenticationData authData
- uint32_t * pIndicationToken

### 8.709.1 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.

### 8.709.2 Field Documentation

#### 8.709.2.1 uim_authenticationData pack_uim_SLQSUIMAuthenticate_t::authData

#### 8.709.2.2 uint32_t* pack_uim_SLQSUIMAuthenticate_t::pIndicationToken

#### 8.709.2.3 uim_sessionInformation pack_uim_SLQSUIMAuthenticate_t::sessionInfo

## 8.710 pack_uim_SLQSUIMDepersonalization_t Struct Reference

### Data Fields

- uim_depersonalizationInformation depersonilisationInfo

### 8.710.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *depersonilisation-Info* | • See uim_depersonalizationInformation for more information. |

### 8.710.2 Field Documentation

#### 8.710.2.1 uim_depersonalizationInformation pack_uim_SLQSUIMDepersonalization_t::depersonilisationInfo

## 8.711 pack_uim_SLQSUIMEventRegister_t Struct Reference

### Data Fields

- uint32_t eventMask

### 8.711.1 Detailed Description

This structure contains pack event register parameter.

**Parameters**

| | |
|---|---|
| *eventMask* | - bit 1 - card status<br>• bit 4 - physical slot status |

### 8.711.2 Field Documentation

#### 8.711.2.1 uint32_t pack_uim_SLQSUIMEventRegister_t::eventMask

## 8.712 pack_uim_SLQSUIMGetConfiguration_t Struct Reference

### Data Fields

- uint32_t * pConfigurationMask

### 8.712.1 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.

### 8.712.2 Field Documentation

#### 8.712.2.1 uint32_t* pack_uim_SLQSUIMGetConfiguration_t::pConfigurationMask

## 8.713 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uim_fileInfo fileIndex
- uint32_t * pIndicationToken

### 8.713.1 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.

### 8.713.2 Field Documentation

#### 8.713.2.1 uim_fileInfo pack_uim_SLQSUIMGetFileAttributes_t::fileIndex

#### 8.713.2.2 uint32_t* pack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

#### 8.713.2.3 uim_sessionInformation pack_uim_SLQSUIMGetFileAttributes_t::sessionInfo

## 8.714 pack_uim_SLQSUIMPowerDown_t Struct Reference

### Data Fields

- uint8_t slot

### 8.714.1 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 |

### 8.714.2 Field Documentation

#### 8.714.2.1 uint8_t pack_uim_SLQSUIMPowerDown_t::slot

## 8.715 pack_uim_SLQSUIMPowerUp_t Struct Reference

### Data Fields

- uint8_t slot
- uint8_t * pIgnoreHotSwapSwitch

### 8.715.1 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 |

### 8.715.2 Field Documentation

#### 8.715.2.1 uint8_t* pack_uim_SLQSUIMPowerUp_t::pIgnoreHotSwapSwitch

#### 8.715.2.2 uint8_t pack_uim_SLQSUIMPowerUp_t::slot

## 8.716 pack_uim_SLQSUIMRefreshComplete_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uint8_t refreshComplete

### 8.716.1 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 |

### 8.716.2 Field Documentation

#### 8.716.2.1 uint8_t pack_uim_SLQSUIMRefreshComplete_t::refreshComplete

#### 8.716.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshComplete_t::sessionInfo

## 8.717 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo

### 8.717.1 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. |

### 8.717.2 Field Documentation

#### 8.717.2.1 uim_sessionInformation pack_uim_SLQSUIMRefreshGetLastEvent_t::sessionInfo

## 8.718 pack_uim_SLQSUIMRefreshOK_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uint8_t OKtoRefresh

### 8.718.1 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 |

### 8.718.2 Field Documentation

#### 8.718.2.1 uint8_t pack_uim_SLQSUIMRefreshOK_t::OKtoRefresh

#### 8.718.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshOK_t::sessionInfo

## 8.719 pack_uim_SLQSUIMRefreshRegister_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uim_registerRefresh regRefresh

### 8.719.1 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 |

### 8.719.2 Field Documentation

#### 8.719.2.1 uim_registerRefresh pack_uim_SLQSUIMRefreshRegister_t::regRefresh

#### 8.719.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshRegister_t::sessionInfo

## 8.720 pack_uim_SLQSUIMSwitchSlot_t Struct Reference

### Data Fields

- uint8_t bLogicalSlot
- uint32_t ulPhysicalSlot

### 8.720.1 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 |

### 8.720.2 Field Documentation

#### 8.720.2.1 uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot

#### 8.720.2.2 uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot

## 8.721 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

### 8.721.1 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. |
| *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 |

### 8.721.2 Field Documentation

#### 8.721.2.1 uim_encryptedPIN1 pack_uim_UnblockPin_t::EncryptedPIN1

#### 8.721.2.2 uint32_t* pack_uim_UnblockPin_t::pIndicationToken

#### 8.721.2.3 uim_unblockUIMPIN pack_uim_UnblockPin_t::pinProtection

#### 8.721.2.4 uint8_t* pack_uim_UnblockPin_t::pKeyReferenceID

#### 8.721.2.5 uim_sessionInformation pack_uim_UnblockPin_t::sessionInfo

#### 8.721.2.6 uint16_t pack_uim_UnblockPin_t::TlvresultIt

## 8.722 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

### 8.722.1 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.

### 8.722.2 Field Documentation

#### 8.722.2.1 uim_encryptedPIN1* pack_uim_VerifyPin_t::pEncryptedPIN1

#### 8.722.2.2 uint32_t* pack_uim_VerifyPin_t::pIndicationToken

#### 8.722.2.3 uint8_t* pack_uim_VerifyPin_t::pKeyReferenceID

#### 8.722.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

#### 8.722.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

#### 8.722.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

## 8.723 pack_voice_AnswerUSSD_t Struct Reference

### Data Fields

- uint8_t * pInfo

### 8.723.1 Detailed Description

This structure contains pack anser USSD parameter.

**Parameters**

| | |
|---|---|
| *pInfo* | • USS information<br>• See voice_USSInfo for more details |

### 8.723.2 Field Documentation

#### 8.723.2.1 uint8_t* pack_voice_AnswerUSSD_t::pInfo

## 8.724 pack_voice_OriginateUSSD_t Struct Reference

### Data Fields

- uint8_t * pInfo

### 8.724.1 Detailed Description

This structure contains pack orginate USSD parameter.

**Parameters**

| | |
|---|---|
| *pInfo* | • USS information<br>• See voice_USSInfo for more details |

### 8.724.2 Field Documentation

#### 8.724.2.1 uint8_t* pack_voice_OriginateUSSD_t::pInfo

## 8.725 pack_voice_SLQSOriginateUSSD_t Struct Reference

### Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

### 8.725.1 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 |

### 8.725.2 Field Documentation

#### 8.725.2.1 uint8_t pack_voice_SLQSOriginateUSSD_t::ussData[182]

#### 8.725.2.2 uint8_t pack_voice_SLQSOriginateUSSD_t::ussDCS

#### 8.725.2.3 uint8_t pack_voice_SLQSOriginateUSSD_t::ussLen

## 8.726 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference

### Data Fields

- uint8_t lineValue

### 8.726.1 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 |

### 8.726.2 Field Documentation

#### 8.726.2.1 uint8_t pack_voice_SLQSVoiceALSSelectLine_t::lineValue

## 8.727 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference

### Data Fields

- uint8_t switchOption

### 8.727.1 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 |

### 8.727.2 Field Documentation

#### 8.727.2.1 uint8_t pack_voice_SLQSVoiceALSSetLineSwitching_t::switchOption

## 8.728 pack_voice_SLQSVoiceAnswerCall_t Struct Reference

### Data Fields

- uint8_t * pCallId

### 8.728.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

**Parameters**

| | |
|---|---|
| *pCallId* | • Unique call identifier for the call that must be answered. |

### 8.728.2 Field Documentation

#### 8.728.2.1 uint8_t* pack_voice_SLQSVoiceAnswerCall_t::pCallId

## 8.729 pack_voice_SLQSVoiceBindSubscription_t Struct Reference

### Data Fields

- uint8_t subsType

### 8.729.1 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 |

### 8.729.2 Field Documentation

#### 8.729.2.1 uint8_t pack_voice_SLQSVoiceBindSubscription_t::subsType

## 8.730 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference

### Data Fields

- voice_burstDTMFInfo BurstDTMFInfo
- voice_DTMFLengths * pBurstDTMFLengths

### 8.730.1 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. |

### 8.730.2 Field Documentation

#### 8.730.2.1 voice_burstDTMFInfo pack_voice_SLQSVoiceBurstDTMF_t::BurstDTMFInfo

#### 8.730.2.2 voice_DTMFLengths* pack_voice_SLQSVoiceBurstDTMF_t::pBurstDTMFLengths

## 8.731 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

### 8.731.1 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] |
| *pCallType(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 |
| *pCLIRType(optional)* | • CLIR type values are:<br>– 0x01 - CLIR_SUPPRESSION - Suppression<br>– 0x02 - CLIR_INVOCATION - Invocation |
| *pUUSInFo(optional)* | • Pointer to structure of UUSInfo<br>– See voice_UUSInfo for more information |
| *pCUGInfo(optional)* | • Pointer to structure of CUGInfo<br>– See voice_CUGInfo for more information |
| *pEmergencyCategory(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 |
| *pCallPartySubAdd(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 |

### 8.731.2 Field Documentation

#### 8.731.2.1 uint8_t pack_voice_SLQSVoiceDialCall_t::callNumber[81]

#### 8.731.2.2 voice_calledPartySubAdd* pack_voice_SLQSVoiceDialCall_t::pCallPartySubAdd

#### 8.731.2.3 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCallType

#### 8.731.2.4 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCLIRType

#### 8.731.2.5 voice_CUGInfo* pack_voice_SLQSVoiceDialCall_t::pCUGInfo

#### 8.731.2.6 uint8_t* pack_voice_SLQSVoiceDialCall_t::pEmergencyCategory

#### 8.731.2.7 uint8_t* pack_voice_SLQSVoiceDialCall_t::pSvcType

#### 8.731.2.8 voice_UUSInfo* pack_voice_SLQSVoiceDialCall_t::pUUSInFo

## 8.732 pack_voice_SLQSVoiceEndCall_t Struct Reference

### Data Fields

- uint8_t * pCallId

### 8.732.1 Detailed Description

This structure contain pack voice end call parameter.

**Parameters**

| | |
|---|---|
| *pCallId* | • Unique call identifier for the call that must be ended |

### 8.732.2 Field Documentation

#### 8.732.2.1 uint8_t* pack_voice_SLQSVoiceEndCall_t::pCallId

## 8.733 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference

### Data Fields

- uint8_t reason
- uint8_t * pSvcClass

### 8.733.1 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 |

### 8.733.2 Field Documentation

#### 8.733.2.1 uint8_t* pack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

#### 8.733.2.2 uint8_t pack_voice_SLQSVoiceGetCallBarring_t::reason

## 8.734 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference

### Data Fields

- uint8_t Reason
- uint8_t * pSvcClass

### 8.734.1 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. |

### 8.734.2 Field Documentation

#### 8.734.2.1 uint8_t* pack_voice_SLQSVoiceGetCallForwardingStatus_t::pSvcClass

#### 8.734.2.2 uint8_t pack_voice_SLQSVoiceGetCallForwardingStatus_t::Reason

## 8.735 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference

### Data Fields

- uint8_t callID

### 8.735.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *callID* | • Call identifier for the call queried for information. |

### 8.735.2 Field Documentation

#### 8.735.2.1 uint8_t pack_voice_SLQSVoiceGetCallInfo_t::callID

## 8.736 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

### Data Fields

- uint8_t * pSvcClass

### 8.736.1 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 |

### 8.736.2 Field Documentation

#### 8.736.2.1 uint8_t* pack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

## 8.737 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

### 8.737.1 Detailed Description

This structure contains Voice Get Configuration Request Parameters

**Parameters**

| | |
|---|---|
| *pAuto-Answer(optional)* | • Indicator to retrieve the Auto Answer Information.<br>– 0x01 - Include auto answer information |
| *pAir-Timer(optional)* | • Indicator to retrieve the Air Timer Information.<br>– 0x01 - Include air calls timer count information<br>• Currently Not Supported. |
| *pRoam-Timer(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.

## 8.737.2 Field Documentation

### 8.737.2.1 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAirTimer

### 8.737.2.2 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAMRStatus

### 8.737.2.3 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAutoAnswer

### 8.737.2.4 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pNamID

### 8.737.2.5 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoicePrivacy

### 8.737.2.6 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoiceSO

### 8.737.2.7 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pRoamTimer

### 8.737.2.8 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pTTYMode

### 8.737.2.9 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pVoiceDomainPref

## 8.738 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference

### Data Fields

- uint8_t * pRegDTMFEvents
- uint8_t * pRegVoicePrivacyEvents
- uint8_t * pSuppsNotifEvents

### 8.738.1 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.

### 8.738.2 Field Documentation

#### 8.738.2.1 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegDTMFEvents

#### 8.738.2.2 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegVoicePrivacyEvents

#### 8.738.2.3 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pSuppsNotifEvents

## 8.739 pack_voice_SLQSVoiceManageCalls_t Struct Reference

### Data Fields

- uint8_t SUPSType

- uint8_t * pCallID

### 8.739.1 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. |

### 8.739.2 Field Documentation

#### 8.739.2.1 uint8_t* pack_voice_SLQSVoiceManageCalls_t::pCallID

#### 8.739.2.2 uint8_t pack_voice_SLQSVoiceManageCalls_t::SUPSType

## 8.740 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference

### Data Fields

- struct voice_USSInfo USSInformation

### 8.740.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

**Parameters**

| | |
|---|---|
| *USSInformation* | • See voice_USSInfo for more information. |

### 8.740.2 Field Documentation

#### 8.740.2.1 struct voice_USSInfo pack_voice_SLQSVoiceOrigUSSDNoWait_t::USSInformation

## 8.741 pack_voice_SLQSVoiceSendFlash_t Struct Reference

### Data Fields

- uint8_t * pCallID
- uint8_t * pFlashPayLd
- uint8_t * pFlashType

### 8.741.1 Detailed Description

This structure contains the flash information associated with a call.

**Parameters**

| | |
|---|---|
| *pCallID* | • Unique call identifier associated with the current call. |
| *pFlashPay-Ld(optional)* | • Payload in ASCII to be sent in Flash.<br>• Variable Length, NULL terminated. |
| *pFlash-Type(optional)* | • Flash type.<br>– 0 - Simple Flash (default)<br>– 1 - Activate answer hold<br>– 2 - Deactivate answer hold |

### 8.741.2 Field Documentation

#### 8.741.2.1 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pCallID

#### 8.741.2.2 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashPayLd

#### 8.741.2.3 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashType

## 8.742 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

### Data Fields

- uint8_t Reason
- uint8_t oldPasswd [4]
- uint8_t newPasswd [4]

- uint8_t newPasswdAgain [4]

### 8.742.1 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. |

### 8.742.2 Field Documentation

#### 8.742.2.1 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswd[4]

#### 8.742.2.2 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswdAgain[4]

#### 8.742.2.3 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::oldPasswd[4]

#### 8.742.2.4 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::Reason

## 8.743 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

## 8.743.1 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.

### 8.743.2 Field Documentation

#### 8.743.2.1 voice_airTimer* pack_voice_SLQSVoiceSetConfig_t::pAirTimerConfig

#### 8.743.2.2 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pAutoAnswer

#### 8.743.2.3 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceDomain

#### 8.743.2.4 voice_prefVoiceSO* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSO

#### 8.743.2.5 voice_roamTimer* pack_voice_SLQSVoiceSetConfig_t::pRoamTimerConfig

#### 8.743.2.6 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pTTYMode

## 8.744 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference

### Data Fields

- uint8_t privacyPref

### 8.744.1 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 |

### 8.744.2 Field Documentation

#### 8.744.2.1 uint8_t pack_voice_SLQSVoiceSetPreferredPrivacy_t::privacyPref

## 8.745 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

### 8.745.1 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 |

### 8.745.2 Field Documentation

#### 8.745.2.1 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallBarringPasswd

#### 8.745.2.2 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallForwardingNumber

#### 8.745.2.3 voice_callFwdTypeAndPlan* pack_voice_SLQSVoiceSetSUPSService_t::pCallFwdTypeAndPlan

#### 8.745.2.4 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pServiceClass

#### 8.745.2.5 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pTimerVal

#### 8.745.2.6 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::reason

#### 8.745.2.7 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::voiceSvc

## 8.746 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference

### Data Fields

- uint8_t * pCallID
- uint8_t DTMFdigit

### 8.746.1 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. |

### 8.746.2 Field Documentation

#### 8.746.2.1 uint8_t pack_voice_SLQSVoiceStartContDTMF_t::DTMFdigit

#### 8.746.2.2 uint8_t* pack_voice_SLQSVoiceStartContDTMF_t::pCallID

## 8.747 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference

### Data Fields

- uint8_t callID

### 8.747.1 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 |

### 8.747.2 Field Documentation

#### 8.747.2.1 uint8_t pack_voice_SLQSVoiceStopContDTMF_t::callID

## 8.748 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference

### Data Fields

- uint8_t * pEnableNotification

### 8.748.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

**Parameters**

| | |
|---|---|
| *pEnable-Notification* | • Enable Notification or not |

### 8.748.2 Field Documentation

#### 8.748.2.1 uint8_t* pack_wds_DHCPv4ClientLeaseChange_t::pEnableNotification

## 8.749 pack_wds_GetDefaultProfile_t Struct Reference

### Data Fields

- uint32_t profiletype

### 8.749.1 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 |

### 8.749.2 Field Documentation

#### 8.749.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

## 8.750 pack_wds_GetDefaultProfileNum_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t family

### 8.750.1 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 |

### 8.750.2 Field Documentation

#### 8.750.2.1 uint8_t pack_wds_GetDefaultProfileNum_t::family

#### 8.750.2.2 uint8_t pack_wds_GetDefaultProfileNum_t::type

## 8.751 pack_wds_GetDormancyState_t Struct Reference

### 8.751.1 Detailed Description

This structure contains pack get dormancy state information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.752 pack_wds_GetLastMobileIPError_t Struct Reference

### 8.752.1 Detailed Description

This structure contains pack get last mobile IP error information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.753 pack_wds_GetMobileIP_t Struct Reference

### 8.753.1 Detailed Description

This structure contains pack get mobile IP information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.754 pack_wds_GetMobileIPProfile_t Struct Reference

### Data Fields

- uint8_t index

### 8.754.1 Detailed Description

This structure contains pack get mobile IP profile information.

**Parameters**

| | |
|---|---|
| *index* | • Mobile IP profile ID |

### 8.754.2 Field Documentation

#### 8.754.2.1 uint8_t pack_wds_GetMobileIPProfile_t::index

## 8.755 pack_wds_GetPacketStatistics_t Struct Reference

### Data Fields

- uint32_t * pStatMask

### 8.755.1 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 |

### 8.755.2 Field Documentation

#### 8.755.2.1 uint32_t* pack_wds_GetPacketStatistics_t::pStatMask

## 8.756 pack_wds_GetPacketStatus_t Struct Reference

### Data Fields

- uint32_t statmask

### 8.756.1 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 |

### 8.756.2 Field Documentation

#### 8.756.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

## 8.757 pack_wds_GetSessionDuration_t Struct Reference

### 8.757.1 Detailed Description

This structure contains pack get session duration information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.758 pack_wds_RMSetTransferStatistics_t Struct Reference

### Data Fields

- rmTrasnferStaticsReq RmTrasnferStaticsReq

### 8.758.1 Detailed Description

This structure contains pack fetch current data system transfer statistics information.

**Parameters**

| | |
|---|---|
| *RmTrasnfer-StaticsReq[IN]* | • See rmTrasnferStaticsReq for more information |

### 8.758.2 Field Documentation

#### 8.758.2.1 rmTrasnferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq

## 8.759 pack_wds_SetAutoconnect_t Struct Reference

### Data Fields

- uint8_t acsetting
- uint8_t acroamsetting

### 8.759.1 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 |

### 8.759.2 Field Documentation

#### 8.759.2.1 uint8_t pack_wds_SetAutoconnect_t::acroamsetting

#### 8.759.2.2 uint8_t pack_wds_SetAutoconnect_t::acsetting

## 8.760 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

### 8.760.1 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) |

### 8.760.2 Field Documentation

#### 8.760.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

#### 8.760.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

#### 8.760.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

#### 8.760.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

#### 8.760.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

#### 8.760.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

#### 8.760.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

#### 8.760.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

#### 8.760.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

#### 8.760.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

## 8.761 pack_wds_SetDefaultProfileNum_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t family
- uint8_t index

### 8.761.1 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. |

### 8.761.2 Field Documentation

#### 8.761.2.1 uint8_t pack_wds_SetDefaultProfileNum_t::family

#### 8.761.2.2 uint8_t pack_wds_SetDefaultProfileNum_t::index

#### 8.761.2.3 uint8_t pack_wds_SetDefaultProfileNum_t::type

## 8.762 pack_wds_SetMobileIP_t Struct Reference

### Data Fields

- uint32_t mode

### 8.762.1 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 |

### 8.762.2 Field Documentation

#### 8.762.2.1 uint32_t pack_wds_SetMobileIP_t::mode

## 8.763 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

### 8.763.1 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 |

### 8.763.2 Field Documentation

#### 8.763.2.1 uint8_t* pack_wds_SetMobileIPParameters_t::pHA2002bis

#### 8.763.2.2 uint8_t* pack_wds_SetMobileIPParameters_t::pHAAuthenticator

#### 8.763.2.3 uint32_t* pack_wds_SetMobileIPParameters_t::pMode

#### 8.763.2.4 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegPeriod

#### 8.763.2.5 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegTraffic

#### 8.763.2.6 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryInterval

#### 8.763.2.7 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryLimit

#### 8.763.2.8 char* pack_wds_SetMobileIPParameters_t::pSPC

## 8.764 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

### 8.764.1 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 |

### 8.764.2 Field Documentation

#### 8.764.2.1 uint8_t pack_wds_SetMobileIPProfile_t::index

#### 8.764.2.2 uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI

#### 8.764.2.3 uint32_t* pack_wds_SetMobileIPProfile_t::pAddress

#### 8.764.2.4 uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled

#### 8.764.2.5 uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI

#### 8.764.2.6 int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA

#### 8.764.2.7 int8_t* pack_wds_SetMobileIPProfile_t::pMNHA

#### 8.764.2.8 int8_t* pack_wds_SetMobileIPProfile_t::pNAI

#### 8.764.2.9 uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA

#### 8.764.2.10 uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling

#### 8.764.2.11 uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA

#### 8.764.2.12 int8_t pack_wds_SetMobileIPProfile_t::spc[10]

## 8.765 pack_wds_SLQSCreateProfile_t Struct Reference

### Data Fields

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

### 8.765.1 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

### 8.765.2 Field Documentation

#### 8.765.2.1 wds_profileInfo* pack_wds_SLQSCreateProfile_t::pCurProfile

#### 8.765.2.2 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileId

#### 8.765.2.3 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileType

## 8.766 pack_wds_SLQSDeleteProfile_t Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileIndex

### 8.766.1 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) |

### 8.766.2 Field Documentation

#### 8.766.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

#### 8.766.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

## 8.767 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

### 8.767.1 Detailed Description

This structure contains pack get current data system state information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.768 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

### 8.768.1 Detailed Description

This structure contains pack get data bearer technology information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.769 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

### 8.769.1 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 |
| *pReportConn-Status* | • Connect Status Indicator<br>– 0 - Do not report<br>– 1 - Report connection status and call end reason |
| *pTransferStatInd* | • See wds_transferStatInd for more information |

| | |
|---|---|
| *pReportDormStatus* | • Dormancy Status Indicator<br>– 0 - Do not report<br>– 1 - Report traffic channel state of interface used for data connection |
| *pReportDataBearerTech* | • Current Data Bearer Technology Indicator<br>– 0 - Do not report<br>– 1 - Report radio interface used for data transfer when it changes |
| *pReportChannelRate* | • Channel Rate Indicator<br>– 0 - Do not report<br>– 1 - Report channel rate |

### 8.769.2 Field Documentation

#### 8.769.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

#### 8.769.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

#### 8.769.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

#### 8.769.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

#### 8.769.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

#### 8.769.2.6 wds_transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

## 8.770 pack_wds_SLQSGetProfileSettings_t Struct Reference

### Data Fields

- uint8_t ProfileId
- uint8_t ProfileType

### 8.770.1 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

### 8.770.2 Field Documentation

#### 8.770.2.1 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId

#### 8.770.2.2 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType

## 8.771 pack_wds_SLQSGetRuntimeSettings_t Struct Reference

### Data Fields

- uint32_t * pReqSettings

### 8.771.1 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) |

### 8.771.2 Field Documentation

#### 8.771.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

## 8.772 pack_wds_SLQSModifyProfile_t Struct Reference

### Data Fields

- uint8_t * pProfileId
- uint8_t * pProfileType
- wds_profileInfo curProfile

### 8.772.1 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

### 8.772.2 Field Documentation

#### 8.772.2.1 wds_profileInfo pack_wds_SLQSModifyProfile_t::curProfile

#### 8.772.2.2 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileId

#### 8.772.2.3 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileType

## 8.773 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

### 8.773.1 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 |

### 8.773.2 Field Documentation

#### 8.773.2.1 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

#### 8.773.2.2 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease

#### 8.773.2.3 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled

#### 8.773.2.4 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile

#### 8.773.2.5 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList

#### 8.773.2.6 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pProfileList

## 8.774 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

### Data Fields

- uint8_t IPFamilyPreference

### 8.774.1 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 |

### 8.774.2 Field Documentation

#### 8.774.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

## 8.775 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

### 8.775.1 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 |

### 8.775.2 Field Documentation

#### 8.775.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

#### 8.775.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

#### 8.775.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearerTechExt

#### 8.775.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

#### 8.775.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

#### 8.775.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

#### 8.775.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

#### 8.775.2.8 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

## 8.776 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

### Data Fields

- wdsDhcpv4ProfileId * pProfileId

### 8.776.1 Detailed Description

This structure contain get DHCPv4 client configure.

**Parameters**

| | |
|---|---|
| *pProfileId* | pointer to Profile Id structure<br>• See wdsDhcpv4ProfileId |

### 8.776.2 Field Documentation

#### 8.776.2.1 wdsDhcpv4ProfileId* pack_wds_SLQSSGetDHCPv4ClientConfig_t::pProfileId

## 8.777 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference

### Data Fields

- wds_DHCPv4ProfileId * pProfileId
- wds_DHCPv4HWConfig * pHwConfig
- wds_DHCPv4OptionList * pRequestOptionList

### 8.777.1 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 |

### 8.777.2 Field Documentation

#### 8.777.2.1 wds_DHCPv4HWConfig∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pHwConfig

#### 8.777.2.2 wds_DHCPv4ProfileId∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pProfileId

#### 8.777.2.3 wds_DHCPv4OptionList∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pRequestOptionList

## 8.778 pack_wds_SLQSSSetLoopback_t Struct Reference

### Data Fields

- uint8_t loopbackMode
- uint8_t loopbackMultiplier

### 8.778.1 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. |

### 8.778.2 Field Documentation

#### 8.778.2.1 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMode

#### 8.778.2.2 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMultiplier

## 8.779 pack_wds_SLQSStartDataSession_t Struct Reference

### Data Fields

- uint8_t ∗ pTech

- uint32_t * pprofileid3gpp
- uint32_t * pprofileid3gpp2
- uint32_t * pAuth
- char * pUser
- char * pPass

### 8.779.1 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 |

### 8.779.2 Field Documentation

#### 8.779.2.1 uint32_t* pack_wds_SLQSStartDataSession_t::pAuth

#### 8.779.2.2 char* pack_wds_SLQSStartDataSession_t::pPass

#### 8.779.2.3 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp

#### 8.779.2.4 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp2

#### 8.779.2.5 uint8_t* pack_wds_SLQSStartDataSession_t::pTech

#### 8.779.2.6 char* pack_wds_SLQSStartDataSession_t::pUser

## 8.780 pack_wds_SLQSStopDataSession_t Struct Reference

### Data Fields

- uint32_t * psid

### 8.780.1 Detailed Description

This structure contains pack stop data session information.

**Parameters**

| | |
|---|---|
| *psid* | session id |

### 8.780.2 Field Documentation

#### 8.780.2.1 uint32_t* pack_wds_SLQSStopDataSession_t::psid

## 8.781 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

### 8.781.1 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 |

| | |
|---|---|
| *pDormancyStatusInd* | (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 |
| *pCurrDataBearerTechInd* | (optional)<br>• Current Data Bearer Technology Indicator.<br>– 0 - Do not report<br>– 1 - Report current data bearer technology when it changes |
| *pDataCallStatusChangeInd* | (optional)<br>• Data Call Status Change Indicator.<br>– 0 - Do not report<br>– 1 - Report data call status change when it changes |
| *pCurrPrefDataSysInd* | (optional)<br>• Current Preferred Data System Indicator.<br>– 0 - Do not report<br>– 1 - Report preferred data system when it changes |
| *pEVDOPageMonPerChangeInd* | (optional)<br>• EV-DO Page Monitor Period Change Indicator.<br>– 0 - Do not report<br>– 1 - Report EV-DO page monitor period change event |
| *pDataSystemStatusChangeInd* | (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.

### 8.781.2 Field Documentation

#### 8.781.2.1 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd

#### 8.781.2.2 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd

#### 8.781.2.3 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd

#### 8.781.2.4 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd

#### 8.781.2.5 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd

#### 8.781.2.6 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd

#### 8.781.2.7 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd

#### 8.781.2.8 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd

#### 8.781.2.9 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd

#### 8.781.2.10 wds_TrStatInd* pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd

## 8.782 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

### Data Fields

- uint8_t contextId
- uint8_t contextType

### 8.782.1 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 |

### 8.782.2 Field Documentation

#### 8.782.2.1 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

#### 8.782.2.2 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType

## 8.783 PackCreateProfileOut Struct Reference

### Data Fields

- uint8_t ProfileType
- uint8_t ProfileIndex
- uint16_t ExtErrorCode

### 8.783.1 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 |

### 8.783.2 Field Documentation

#### 8.783.2.1 uint16_t PackCreateProfileOut::ExtErrorCode

#### 8.783.2.2 uint8_t PackCreateProfileOut::ProfileIndex

#### 8.783.2.3 uint8_t PackCreateProfileOut::ProfileType

## 8.784 packgetDyingGaspCfg Struct Reference

### Data Fields

- uint8_t * pDestSMSNum
- uint8_t * pDestSMSContent

### 8.784.1 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. |

### 8.784.2 Field Documentation

#### 8.784.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

#### 8.784.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

## 8.785 packgetDyingGaspStatistics Struct Reference

### Data Fields

- uint32_t * pTimeStamp

- uint8_t * pSMSAttemptedFlag

### 8.785.1 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. |

### 8.785.2 Field Documentation

#### 8.785.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

#### 8.785.2.2 uint32_t* packgetDyingGaspStatistics::pTimeStamp

## 8.786 PCMparams Struct Reference

### Data Fields

- BYTE iFaceTabLen
- BYTE iFaceTab [255]

### 8.786.1 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 |

### 8.786.2 Field Documentation

#### 8.786.2.1 BYTE PCMparams::iFaceTab[255]

#### 8.786.2.2 BYTE PCMparams::iFaceTabLen

## 8.787 PCSCFFQDNAddress Struct Reference

### Data Fields

- WORD fqdnLen
- CHAR fqdnAddr [256]

### 8.787.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

**Parameters**

| | |
|---|---|
| *fqdnLen* | • length of the received FQDN address |
| *fqdnAddr* | • FQDN address(Max 256 characters) |

### 8.787.2 Field Documentation

#### 8.787.2.1 CHAR PCSCFFQDNAddress::fqdnAddr[256]

#### 8.787.2.2 WORD PCSCFFQDNAddress::fqdnLen

## 8.788 PCSCFFQDNAddressList Struct Reference

### Data Fields

- BYTE numInstances
- struct PCSCFFQDNAddress pcsfFQDNAddress [10]

### 8.788.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of FQDN addresses received |
| *pcsfFQDN-Address* | • FQDN address information(Max 10 addresses) |

### 8.788.2 Field Documentation

#### 8.788.2.1 BYTE PCSCFFQDNAddressList::numInstances

#### 8.788.2.2 struct PCSCFFQDNAddress PCSCFFQDNAddressList::pcsfFQDNAddress[10]

## 8.789 PCSCFIPv4ServerAddressList Struct Reference

### Data Fields

- BYTE numInstances
- ULONG pscsfIPv4Addr [64]

### 8.789.1 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) |

### 8.789.2 Field Documentation

#### 8.789.2.1 BYTE PCSCFIPv4ServerAddressList::numInstances

#### 8.789.2.2 ULONG PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

## 8.790 PDSPositionData Struct Reference

### Data Fields

- ULONGLONG * pTimeStamp
- ULONGLONG * pLatitude
- ULONGLONG * pLongitude
- ULONG * pAltitudeWrtEllipsoid
- ULONG * pAltitudeWrtSealevel
- ULONG * pHorizontalUncCircular
- ULONG * pVerticalUnc
- BYTE * pHorizontalConfidence
- BYTE * pVerticalConfidence
- BYTE * pPositionSource
- BYTE * pTimeType

### 8.790.1 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 |

### 8.790.2 Field Documentation

#### 8.790.2.1 ULONG* PDSPositionData::pAltitudeWrtEllipsoid

#### 8.790.2.2 ULONG* PDSPositionData::pAltitudeWrtSealevel

#### 8.790.2.3 BYTE* PDSPositionData::pHorizontalConfidence

#### 8.790.2.4 ULONG* PDSPositionData::pHorizontalUncCircular

#### 8.790.2.5 ULONGLONG* PDSPositionData::pLatitude

#### 8.790.2.6 ULONGLONG* PDSPositionData::pLongitude

#### 8.790.2.7 BYTE* PDSPositionData::pPositionSource

#### 8.790.2.8 ULONGLONG* PDSPositionData::pTimeStamp

#### 8.790.2.9 BYTE* PDSPositionData::pTimeType

#### 8.790.2.10 BYTE* PDSPositionData::pVerticalConfidence

#### 8.790.2.11 ULONG* PDSPositionData::pVerticalUnc

## 8.791 PDSPosMethodStateReq Struct Reference

### Data Fields

- BYTE * pXtraTimeState
- BYTE * pXtraDataState
- BYTE * pWifiState

### 8.791.1 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 |

### 8.791.2 Field Documentation

#### 8.791.2.1 BYTE* PDSPosMethodStateReq::pWifiState

#### 8.791.2.2 BYTE* PDSPosMethodStateReq::pXtraDataState

#### 8.791.2.3 BYTE* PDSPosMethodStateReq::pXtraTimeState

## 8.792 peerNumberInfo Struct Reference

### Data Fields

- BYTE callID
- BYTE numPI
- BYTE numSI
- BYTE numType
- BYTE numPlan
- BYTE numLen
- BYTE number [81]

### 8.792.1 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_CALL_NO_LEN]* | • number of numLen length, NULL terminated. |

### 8.792.2 Field Documentation

#### 8.792.2.1 BYTE peerNumberInfo::callID

#### 8.792.2.2 BYTE peerNumberInfo::number[81]

#### 8.792.2.3 BYTE peerNumberInfo::numLen

#### 8.792.2.4 BYTE peerNumberInfo::numPI

#### 8.792.2.5 BYTE peerNumberInfo::numPlan

#### 8.792.2.6 BYTE peerNumberInfo::numSI

#### 8.792.2.7 BYTE peerNumberInfo::numType

## 8.793 personalizationStatus Struct Reference

### Data Fields

- BYTE numFeatures
- BYTE feature [12]
- BYTE verifyLeft [12]
- BYTE unblockLeft [12]

### 8.793.1 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. |

### 8.793.2 Field Documentation

#### 8.793.2.1 BYTE personalizationStatus::feature[12]

#### 8.793.2.2 BYTE personalizationStatus::numFeatures

#### 8.793.2.3 BYTE personalizationStatus::unblockLeft[12]

#### 8.793.2.4 BYTE personalizationStatus::verifyLeft[12]

## 8.794 PhyCaAggPcellInfo Struct Reference

### Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- int iLTEbandValue
- BYTE TlvPresent

### 8.794.1 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 for more information. |
| *scell_state* | • Scell state Values.<br>• See NAS_LTE_CPHY_SCELL_STATE for more information. |
| *TlvPresent* | • Tlv Present. |

### 8.794.2 Field Documentation

#### 8.794.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggPcellInfo::dl_bw_value

#### 8.794.2.2 int PhyCaAggPcellInfo::freq

#### 8.794.2.3 int PhyCaAggPcellInfo::iLTEbandValue

#### 8.794.2.4 int PhyCaAggPcellInfo::pci

#### 8.794.2.5 BYTE PhyCaAggPcellInfo::TlvPresent

## 8.795 PhyCaAggScellDlBw Struct Reference

### Data Fields

- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value

- BYTE TlvPresent

### 8.795.1 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 for more information. |

### 8.795.2 Field Documentation

#### 8.795.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellDlBw::dl_bw_value

#### 8.795.2.2 BYTE PhyCaAggScellDlBw::TlvPresent

## 8.796 PhyCaAggScellIndex Struct Reference

### Data Fields

- BYTE scell_idx
- BYTE TlvPresent

### 8.796.1 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. |

### 8.796.2 Field Documentation

#### 8.796.2.1 BYTE PhyCaAggScellIndex::scell_idx

#### 8.796.2.2 BYTE PhyCaAggScellIndex::TlvPresent

## 8.797 PhyCaAggScellIndType Struct Reference

### Data Fields

- int pci
- int freq

- NAS_LTE_CPHY_SCELL_STATE scell_state
- BYTE TlvPresent

### 8.797.1 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 NAS_LTE_CPHY_SCELL_STATE for more information. |
| *TlvPresent* | • Tlv Present. |

### 8.797.2 Field Documentation

#### 8.797.2.1 int PhyCaAggScellIndType::freq

#### 8.797.2.2 int PhyCaAggScellIndType::pci

#### 8.797.2.3 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellIndType::scell_state

#### 8.797.2.4 BYTE PhyCaAggScellIndType::TlvPresent

## 8.798 PhyCaAggScellInfo Struct Reference

### Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- int iLTEbandValue
- NAS_LTE_CPHY_SCELL_STATE scell_state
- BYTE TlvPresent

### 8.798.1 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 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 for more information. |

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

### 8.798.2 Field Documentation

#### 8.798.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellInfo::dl_bw_value

#### 8.798.2.2 int PhyCaAggScellInfo::freq

#### 8.798.2.3 int PhyCaAggScellInfo::iLTEbandValue

#### 8.798.2.4 int PhyCaAggScellInfo::pci

#### 8.798.2.5 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellInfo::scell_state

#### 8.798.2.6 BYTE PhyCaAggScellInfo::TlvPresent

## 8.799 PilotSetData Struct Reference

### Data Fields

- BYTE NumPilots
- PilotSetParams * pPilotSetInfo

### 8.799.1 Detailed Description

This structure contains Pilot Set Data

**Parameters**

| | |
|---|---|
| *NumPilots(IN/OUT)* | • 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 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.

### 8.799.2 Field Documentation

#### 8.799.2.1 BYTE PilotSetData::NumPilots

#### 8.799.2.2 PilotSetParams* PilotSetData::pPilotSetInfo

# 8.800 PilotSetParams Struct Reference

## Data Fields

- ULONG PilotType
- WORD PilotPN
- WORD PilotStrength

### 8.800.1 Detailed Description

This structure contains Pilot Set 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) |

### 8.800.2 Field Documentation

#### 8.800.2.1 WORD PilotSetParams::PilotPN

#### 8.800.2.2 WORD PilotSetParams::PilotStrength

#### 8.800.2.3 ULONG PilotSetParams::PilotType

# 8.801 pktErrRate Struct Reference

## Data Fields

- WORD multiplier
- WORD exponent

### 8.801.1 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) |

### 8.801.2 Field Documentation

#### 8.801.2.1 WORD pktErrRate::exponent

#### 8.801.2.2 WORD pktErrRate::multiplier

## 8.802 PlmnID Struct Reference

### Data Fields

- WORD mcc
- WORD mnc
- BOOL pcsDigit
- BYTE TlvPresent

### 8.802.1 Detailed Description

This structure contains PLMN ID information

**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 |
| *pcsDigit* | • This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field. Values:<br>– TRUE - MNC is a three-digit value; for example, a reported value of 90 corresponds to an MNC value of 090<br>– FALSE - MNC is a two-digit value; for example, a reported value of 90 corresponds to an MNC value of 90 |
| *TlvPresent* | • Tlv Present or not |

### 8.802.2 Field Documentation

#### 8.802.2.1 WORD PlmnID::mcc

#### 8.802.2.2 WORD PlmnID::mnc

#### 8.802.2.3 BOOL PlmnID::pcsDigit

#### 8.802.2.4 BYTE PlmnID::TlvPresent

## 8.803 PLMNNetworkName Struct Reference

### Data Fields

- BYTE numInstance
- PLMNNetworkNameData PLMNNetName [255]

### 8.803.1 Detailed Description

This structure contains PLMN Network Name as defined in 3GPP TS 24.008 (Section 10.5.3.5a) from multiple sources.

**Parameters**

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

### 8.803.2 Field Documentation

#### 8.803.2.1 BYTE PLMNNetworkName::numInstance

#### 8.803.2.2 PLMNNetworkNameData PLMNNetworkName::PLMNNetName[255]

## 8.804 PLMNNetworkNameData Struct Reference

### Data Fields

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

### 8.804.1 Detailed Description

This structure contains PLMN Network Name Data from multiple sources.

**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. |

### 8.804.2 Field Documentation

#### 8.804.2.1 BYTE PLMNNetworkNameData::codingScheme

#### 8.804.2.2 BYTE PLMNNetworkNameData::countryInitials

#### 8.804.2.3 BYTE PLMNNetworkNameData::longName[255]

#### 8.804.2.4 BYTE PLMNNetworkNameData::longNameLen

#### 8.804.2.5 BYTE PLMNNetworkNameData::longNameSpareBits

#### 8.804.2.6 BYTE PLMNNetworkNameData::shortName[255]

#### 8.804.2.7 BYTE PLMNNetworkNameData::shortNameLen

#### 8.804.2.8 BYTE PLMNNetworkNameData::shortNameSpareBits

## 8.805 Port Struct Reference

### Data Fields

- WORD port
- WORD range

### 8.805.1 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 |

### 8.805.2 Field Documentation

#### 8.805.2.1 WORD Port::port

#### 8.805.2.2 WORD Port::range

## 8.806 PowerSaveModeConfigReq Struct Reference

### Data Fields

- BYTE * pPsmEnableState
- ULONG * pActiveTimer
- ULONG * pPeriodicUpdateTimer

### 8.806.1 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. |

### 8.806.2 Field Documentation

#### 8.806.2.1 ULONG* PowerSaveModeConfigReq::pActiveTimer

#### 8.806.2.2 ULONG* PowerSaveModeConfigReq::pPeriodicUpdateTimer

#### 8.806.2.3 BYTE* PowerSaveModeConfigReq::pPsmEnableState

## 8.807 PowerSaveModeConfigResp Struct Reference

### Data Fields

- PSMEnableStateTlv * pPsmEnableState
- PSMDurationThresholdTlv * pDurationThreshold
- PSMDurationDueToOOSTlv * pDurationDueToOOS
- PSMRandomizationWindowTlv * pRandomizationWindow
- PSMActiveTimerTlv * pActiveTimer
- PSMPeriodicUpdateTimerTlv * pPeriodicUpdateTimer
- PSMEarlyWakeupTimeTlv * pEarlyWakeupTime

### 8.807.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

**Parameters**

| | |
|---|---|
| *pPsmEnable-State* | [Optional]<br>• See PSMEnableStateTlv for more information |
| *pDuration-Threshold* | [Optional]<br>• See PSMDurationThresholdTlv for more information |
| *pDurationDueTo-OOS* | [Optional]<br>• See PSMDurationDueToOOSTlv for more information |
| *pRandomization-Window* | [Optional]<br>• See PSMRandomizationWindowTlv for more information |
| *pActiveTimer* | [Optional]<br>• See PSMActiveTimerTlv for more information |
| *pPeriodic-UpdateTimer* | [Optional]<br>• See PSMPeriodicUpdateTimerTlv for more information |
| *pEarlyWakeup-Time* | [Optional]<br>• See PSMEarlyWakeupTimeTlv for more information |

### 8.807.2 Field Documentation

#### 8.807.2.1 PSMActiveTimerTlv* PowerSaveModeConfigResp::pActiveTimer

#### 8.807.2.2 PSMDurationDueToOOSTlv* PowerSaveModeConfigResp::pDurationDueToOOS

#### 8.807.2.3 PSMDurationThresholdTlv* PowerSaveModeConfigResp::pDurationThreshold

#### 8.807.2.4 PSMEarlyWakeupTimeTlv* PowerSaveModeConfigResp::pEarlyWakeupTime

#### 8.807.2.5 PSMPeriodicUpdateTimerTlv* PowerSaveModeConfigResp::pPeriodicUpdateTimer

#### 8.807.2.6 PSMEnableStateTlv* PowerSaveModeConfigResp::pPsmEnableState

#### 8.807.2.7 PSMRandomizationWindowTlv* PowerSaveModeConfigResp::pRandomizationWindow

## 8.808 precisionDilution_s Struct Reference

### Data Fields

- ULONG PDOP
- ULONG HDOP
- ULONG VDOP

### 8.808.1 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) |

### 8.808.2 Field Documentation

#### 8.808.2.1 ULONG precisionDilution_s::HDOP

#### 8.808.2.2 ULONG precisionDilution_s::PDOP

#### 8.808.2.3 ULONG precisionDilution_s::VDOP

## 8.809 PrefImageList Struct Reference

### Data Fields

- BYTE listSize
- struct ImageElement listEntries [2]

### 8.809.1 Detailed Description

This structure contains the Preference Image List information

**Parameters**

| | |
|---|---|
| *listSize* | • The number of elements in the image list |
| *listEntries* | • Array of Image entries( Max array size 2 )<br>• See ImageElement |

### 8.809.2 Field Documentation

#### 8.809.2.1 struct ImageElement PrefImageList::listEntries[2]

#### 8.809.2.2 BYTE PrefImageList::listSize

## 8.810 prefVoiceSO Struct Reference

### Data Fields

- BYTE namID
- BYTE evrcCapability
- WORD homePageVoiceSO
- WORD homeOrigVoiceSO
- WORD roamOrigVoiceSO

### 8.810.1 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 |

| | |
|---|---|
| *roamOrigVoice-SO* | • 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 |

### 8.810.2 Field Documentation

#### 8.810.2.1 BYTE prefVoiceSO::evrcCapability

#### 8.810.2.2 WORD prefVoiceSO::homeOrigVoiceSO

#### 8.810.2.3 WORD prefVoiceSO::homePageVoiceSO

#### 8.810.2.4 BYTE prefVoiceSO::namID

#### 8.810.2.5 WORD prefVoiceSO::roamOrigVoiceSO

## 8.811 Profile3GPP Struct Reference

### Data Fields

- CHAR * pProfilename
- WORD * pProfilenameSize
- BYTE * pPDPtype
- BYTE * pPdpHdrCompType
- BYTE * pPdpDataCompType
- CHAR * pAPNName
- WORD * pAPNnameSize
- ULONG * pPriDNSIPv4AddPref
- ULONG * pSecDNSIPv4AddPref
- struct UMTSQoS * pUMTSReqQoS
- struct UMTSQoS * pUMTSMinQoS
- struct GPRSRequestedQoS * pGPRSRequestedQos
- struct GPRSRequestedQoS * pGPRSMinimumQoS
- CHAR * pUsername
- WORD * pUsernameSize
- CHAR * pPassword
- WORD * pPasswordSize
- BYTE * pAuthenticationPref
- ULONG * pIPv4AddrPref

- BYTE * pPcscfAddrUsingPCO
- BYTE * pPdpAccessConFlag
- BYTE * pPcscfAddrUsingDhcp
- BYTE * pImCnFlag
- struct TFTIDParams * pTFTID1Params
- struct TFTIDParams * pTFTID2Params
- BYTE * pPdpContext
- BYTE * pSecondaryFlag
- BYTE * pPrimaryID
- USHORT * pIPv6AddPref
- struct UMTSReqQoSSigInd * pUMTSReqQoSSigInd
- struct UMTSReqQoSSigInd * pUMTSMinQosSigInd
- USHORT * pPriDNSIPv6addpref
- USHORT * pSecDNSIPv6addpref
- BYTE * pAddrAllocPref
- struct QosClassID * pQosClassID
- BYTE * pAPNDisabledFlag
- ULONG * pPDNInactivTimeout
- BYTE * pAPNClass
- BYTE * pSupportEmergencyCalls

### 8.811.1 Detailed Description

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

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

**Parameters**

| | |
|---|---|
| *pProfileName* | • One or more bytes describing the profile |
| *pProfilename-Size;* | • 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 |
| *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 |

| | |
|---|---|
| *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. |
| *pPriDNSIPv4AddPref* | • Primary DNS IPv4 Address Preference |
| *pSecDNSIPv4AddPref* | • Secondary DNS IPv4 Address Preference |
| *pUMTSReqQoS* | • UMTS Requested QoS |
| *pUMTSMinQoS* | • UMTS Minimum QoS |
| *pGPRSRequestedQoS* | • 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 |
| *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 |
| *pSecondaryDN-SIPv6addpref* | • Secondary DNS IPv6 address preference |
| *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 |
| *pQosClassID* | • 3GPP LTE QoS parameters |
| *pAPNDisabled-Flag* | • Optional 1 Byte 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(). |
| *pPDNInactiv-Timeout* | • 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(). |

| | |
|---|---|
| *pAPNClass* | • Optional 1 Byte 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(). |
| *pSupport-EmergencyCalls* | • 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 |

## 8.811.2 Field Documentation

### 8.811.2.1 BYTE* Profile3GPP::pAddrAllocPref

### 8.811.2.2 BYTE* Profile3GPP::pAPNClass

### 8.811.2.3 BYTE* Profile3GPP::pAPNDisabledFlag

### 8.811.2.4 CHAR* Profile3GPP::pAPNName

### 8.811.2.5 WORD* Profile3GPP::pAPNnameSize

### 8.811.2.6 BYTE* Profile3GPP::pAuthenticationPref

### 8.811.2.7 struct GPRSRequestedQoS* Profile3GPP::pGPRSMinimumQoS

### 8.811.2.8 struct GPRSRequestedQoS* Profile3GPP::pGPRSRequestedQos

### 8.811.2.9 BYTE* Profile3GPP::pImCnFlag

### 8.811.2.10 ULONG* Profile3GPP::pIPv4AddrPref

### 8.811.2.11 USHORT* Profile3GPP::pIPv6AddPref

### 8.811.2.12 CHAR* Profile3GPP::pPassword

### 8.811.2.13 WORD* Profile3GPP::pPasswordSize

### 8.811.2.14 BYTE* Profile3GPP::pPcscfAddrUsingDhcp

### 8.811.2.15 BYTE* Profile3GPP::pPcscfAddrUsingPCO

### 8.811.2.16 ULONG* Profile3GPP::pPDNInactivTimeout

### 8.811.2.17 BYTE* Profile3GPP::pPdpAccessConFlag

### 8.811.2.18 BYTE* Profile3GPP::pPdpContext

### 8.811.2.19 BYTE* Profile3GPP::pPdpDataCompType

8.811.2.20 **BYTE*** Profile3GPP::pPdpHdrCompType

8.811.2.21 **BYTE*** Profile3GPP::pPDPtype

8.811.2.22 **ULONG*** Profile3GPP::pPriDNSIPv4AddPref

8.811.2.23 **USHORT*** Profile3GPP::pPriDNSIPv6addpref

8.811.2.24 **BYTE*** Profile3GPP::pPrimaryID

8.811.2.25 **CHAR*** Profile3GPP::pProfilename

8.811.2.26 **WORD*** Profile3GPP::pProfilenameSize

8.811.2.27 struct **QosClassID*** Profile3GPP::pQosClassID

8.811.2.28 **ULONG*** Profile3GPP::pSecDNSIPv4AddPref

8.811.2.29 **USHORT*** Profile3GPP::pSecDNSIPv6addpref

8.811.2.30 **BYTE*** Profile3GPP::pSecondaryFlag

8.811.2.31 **BYTE*** Profile3GPP::pSupportEmergencyCalls

8.811.2.32 struct **TFTIDParams*** Profile3GPP::pTFTID1Params

8.811.2.33 struct **TFTIDParams*** Profile3GPP::pTFTID2Params

8.811.2.34 struct **UMTSQoS*** Profile3GPP::pUMTSMinQoS

8.811.2.35 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSMinQosSigInd

8.811.2.36 struct **UMTSQoS*** Profile3GPP::pUMTSReqQoS

8.811.2.37 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSReqQoSSigInd

8.811.2.38 **CHAR*** Profile3GPP::pUsername

8.811.2.39 **WORD*** Profile3GPP::pUsernameSize

## 8.812 Profile3GPP2 Struct Reference

### Data Fields

- BYTE * pNegoDnsSrvrPref
- ULONG * pPppSessCloseTimerDO
- ULONG * pPppSessCloseTimer1x
- BYTE * pAllowLinger
- USHORT * pLcpAckTimeout
- USHORT * pIpcpAckTimeout
- USHORT * pAuthTimeout
- BYTE * pLcpCreqRetryCount
- BYTE * pIpcpCreqRetryCount
- BYTE * pAuthRetryCount
- BYTE * pAuthProtocol

- CHAR * pUserId
- WORD * pUserIdSize
- CHAR * pAuthPassword
- WORD * pAuthPasswordSize
- BYTE * pDataRate
- ULONG * pAppType
- BYTE * pDataMode
- BYTE * pAppPriority
- CHAR * pApnString
- WORD * pApnStringSize
- BYTE * pPdnType
- BYTE * pIsPcscfAddressNedded
- ULONG * pPrimaryV4DnsAddress
- ULONG * pSecondaryV4DnsAddress
- USHORT * pPriV6DnsAddress
- USHORT * pSecV6DnsAddress
- BYTE * pRATType
- BYTE * pAPNEnabled3GPP2
- ULONG * pPDNInactivTimeout3GPP2
- BYTE * pAPNClass3GPP2

### 8.812.1 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 byte 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 Byte 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 Byte 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 Byte 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(). |

### 8.812.2 Field Documentation

#### 8.812.2.1 BYTE* Profile3GPP2::pAllowLinger

#### 8.812.2.2 BYTE* Profile3GPP2::pAPNClass3GPP2

#### 8.812.2.3 BYTE* Profile3GPP2::pAPNEnabled3GPP2

#### 8.812.2.4 CHAR* Profile3GPP2::pApnString

#### 8.812.2.5 WORD* Profile3GPP2::pApnStringSize

#### 8.812.2.6 BYTE* Profile3GPP2::pAppPriority

#### 8.812.2.7 ULONG* Profile3GPP2::pAppType

#### 8.812.2.8 CHAR* Profile3GPP2::pAuthPassword

#### 8.812.2.9 WORD* Profile3GPP2::pAuthPasswordSize

#### 8.812.2.10 BYTE* Profile3GPP2::pAuthProtocol

#### 8.812.2.11 BYTE* Profile3GPP2::pAuthRetryCount

#### 8.812.2.12 USHORT* Profile3GPP2::pAuthTimeout

#### 8.812.2.13 BYTE* Profile3GPP2::pDataMode

#### 8.812.2.14 BYTE* Profile3GPP2::pDataRate

#### 8.812.2.15 USHORT* Profile3GPP2::pIpcpAckTimeout

#### 8.812.2.16 BYTE* Profile3GPP2::pIpcpCreqRetryCount

#### 8.812.2.17 BYTE* Profile3GPP2::pIsPcscfAddressNedded

#### 8.812.2.18 USHORT* Profile3GPP2::pLcpAckTimeout

#### 8.812.2.19 BYTE* Profile3GPP2::pLcpCreqRetryCount

#### 8.812.2.20 BYTE* Profile3GPP2::pNegoDnsSrvrPref

#### 8.812.2.21 ULONG* Profile3GPP2::pPDNInactivTimeout3GPP2

#### 8.812.2.22 BYTE* Profile3GPP2::pPdnType

#### 8.812.2.23 ULONG* Profile3GPP2::pPppSessCloseTimer1x

#### 8.812.2.24 ULONG* Profile3GPP2::pPppSessCloseTimerDO

#### 8.812.2.25 ULONG* Profile3GPP2::pPrimaryV4DnsAddress

#### 8.812.2.26 USHORT* Profile3GPP2::pPriV6DnsAddress

#### 8.812.2.27 BYTE* Profile3GPP2::pRATType

#### 8.812.2.28 ULONG* Profile3GPP2::pSecondaryV4DnsAddress

#### 8.812.2.29 USHORT* Profile3GPP2::pSecV6DnsAddress

#### 8.812.2.30 CHAR* Profile3GPP2::pUserId

#### 8.812.2.31 WORD* Profile3GPP2::pUserIdSize

## 8.813 ProfileIdentifier Struct Reference

### Data Fields

- BYTE profileType
- BYTE profileIndex

### 8.813.1 Detailed Description

This structure contains the Profile Identifier Information

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

**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 |

### 8.813.2 Field Documentation

#### 8.813.2.1 BYTE ProfileIdentifier::profileIndex

#### 8.813.2.2 BYTE ProfileIdentifier::profileType

## 8.814 protocolSubtypeElement Struct Reference

### Data Fields

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

### 8.814.1 Detailed Description

This structure contains Protocol Subtype Elements for Protocol Subtype List

**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 |

## 8.814.2 Field Documentation

### 8.814.2.1 WORD protocolSubtypeElement::AccessMac

### 8.814.2.2 WORD protocolSubtypeElement::AuthProt

### 8.814.2.3 WORD protocolSubtypeElement::ControlMac

### 8.814.2.4 WORD protocolSubtypeElement::EncryptProt

### 8.814.2.5 WORD protocolSubtypeElement::ForwardMac

### 8.814.2.6 WORD protocolSubtypeElement::IdleState

### 8.814.2.7 WORD protocolSubtypeElement::KeyExchange

### 8.814.2.8 WORD protocolSubtypeElement::MultDisc

### 8.814.2.9 WORD protocolSubtypeElement::PhysicalLayer

### 8.814.2.10 WORD protocolSubtypeElement::ReverseMac

#### 8.814.2.11 WORD protocolSubtypeElement::SecProt

#### 8.814.2.12 WORD protocolSubtypeElement::VirtStream

## 8.815 PSDetachReq Struct Reference

### Data Fields

- BYTE * pDetachAction

### 8.815.1 Detailed Description

This structure contains information about the SLQSSwiPSDetach request parameters.

**Parameters**

| | |
|---|---|
| *pDetachAction[IN]* | • Values<br>– 2- Initiates an immediate packet domain detach. |

### 8.815.2 Field Documentation

#### 8.815.2.1 BYTE* PSDetachReq::pDetachAction

## 8.816 PSMActiveTimerTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG activeTimer

### 8.816.1 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). |

### 8.816.2 Field Documentation

#### 8.816.2.1 ULONG PSMActiveTimerTlv::activeTimer

#### 8.816.2.2 BYTE PSMActiveTimerTlv::TlvPresent

## 8.817 PSMDurationDueToOOSTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG durationDueToOOS

### 8.817.1 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). |

### 8.817.2 Field Documentation

#### 8.817.2.1 ULONG PSMDurationDueToOOSTlv::durationDueToOOS

#### 8.817.2.2 BYTE PSMDurationDueToOOSTlv::TlvPresent

## 8.818 PSMDurationThresholdTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG durationThreshold

### 8.818.1 Detailed Description

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

**Parameters**

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

### 8.818.2 Field Documentation

#### 8.818.2.1 ULONG PSMDurationThresholdTlv::durationThreshold

#### 8.818.2.2 BYTE PSMDurationThresholdTlv::TlvPresent

## 8.819 PSMEarlyWakeupTimeTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG earlyWakeupTime

### 8.819.1 Detailed Description

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

**Parameters**

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

### 8.819.2 Field Documentation

#### 8.819.2.1 ULONG PSMEarlyWakeupTimeTlv::earlyWakeupTime

#### 8.819.2.2 BYTE PSMEarlyWakeupTimeTlv::TlvPresent

## 8.820 PSMEnableStateTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- BYTE enableState

### 8.820.1 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 |

### 8.820.2 Field Documentation

#### 8.820.2.1 BYTE PSMEnableStateTlv::enableState

#### 8.820.2.2 BYTE PSMEnableStateTlv::TlvPresent

## 8.821 PSMPeriodicUpdateTimerTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG periodicUpdateTimer

### 8.821.1 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). |

### 8.821.2 Field Documentation

#### 8.821.2.1 ULONG PSMPeriodicUpdateTimerTlv::periodicUpdateTimer

#### 8.821.2.2 BYTE PSMPeriodicUpdateTimerTlv::TlvPresent

## 8.822 PSMRandomizationWindowTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG randomizationWindow

### 8.822.1 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) |

### 8.822.2 Field Documentation

#### 8.822.2.1 ULONG PSMRandomizationWindowTlv::randomizationWindow

#### 8.822.2.2 BYTE PSMRandomizationWindowTlv::TlvPresent

## 8.823 qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- BYTE leapSeconds
- BYTE localTimeOffset
- BYTE daylightSavings

### 8.823.1 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) |

### 8.823.2 Field Documentation

#### 8.823.2.1 BYTE qaQmi3Gpp2TimeZone::daylightSavings

#### 8.823.2.2 BYTE qaQmi3Gpp2TimeZone::leapSeconds

#### 8.823.2.3 BYTE qaQmi3Gpp2TimeZone::localTimeOffset

## 8.824 qaQmiInterfaceInfo Struct Reference

### Data Fields

- BYTE qaQmiinstanceid
- eQaQMIService qaQmisvctype
- ULONG v4sessionId
- ULONG v6sessionId

### 8.824.1 Detailed Description

Structure used to store the service, interface and session information

**Parameters**

| | |
|---|---|
| *qaQmiinstanceid* | • The interface instance ID<br>– 0x00 - PDP instance ID 0<br>– 0x01 - PDP instance ID 1<br>– 0x02 - PDP instance ID 2 |
| *qaQmisvctype* | • The service type information. See eQaQMIService for more information |
| *v4sessionId* | • IPv4 QMI client session handle |
| *v6sessionId* | • IPv6 QMI client session handle |

### 8.824.2 Field Documentation

#### 8.824.2.1 BYTE qaQmiInterfaceInfo::qaQmiinstanceid

#### 8.824.2.2 eQaQMIService qaQmiInterfaceInfo::qaQmisvctype

#### 8.824.2.3 ULONG qaQmiInterfaceInfo::v4sessionId

#### 8.824.2.4 ULONG qaQmiInterfaceInfo::v6sessionId

## 8.825 qaQmiServingSystemParam Struct Reference

### Data Fields

- servSystem ServingSystem
- BYTE roamIndicatorVal
- dataSrvCapabilities DataSrvCapabilities
- currentPLMN CurrentPLMN
- WORD SystemID
- WORD NetworkID
- WORD BasestationID
- ULONG BasestationLatitude
- ULONG BasestationLongitude
- roamIndList RoamingIndicatorList
- BYTE defaultRoamInd
- qaQmi3Gpp2TimeZone Gpp2TimeZone
- BYTE CDMA_P_Rev
- BYTE GppTimeZone
- BYTE GppNetworkDSTAdjustment
- WORD Lac
- ULONG CellID
- BYTE concSvcInfo
- BYTE PRLInd
- BYTE DTMInd
- detailSvcInfo DetailedSvcInfo
- CDMASysInfoExt CDMASystemInfoExt

- BYTE hdrPersonality
- WORD trackAreaCode
- callBarStatus CallBarStatus

### 8.825.1 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 servSystem for more information |
| *roamIndicatorVal* | • Optional parameter indicating Roaming Indicator value<br>• Values:<br>– 0x00 - Roaming<br>– 0x01 - Home<br>– 0x02 - Flashing<br>– 0x03 and above - Operator defined values |
| *DataSrv-Capabilities* | • Optional parameter indicating Data services capability<br>• See dataSrvCapabilities for more information |
| *CurrentPLMN* | • Optional parameter indicating Current PLMN<br>• See currentPLMN for more information |
| *SystemID* | • Optional parameter indicating System ID |
| *NetworkID* | • Optional parameter indicating Network ID |
| *BaseStationID* | • Optional parameter indicating Base Station Identification Number |
| *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 |
| *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 |
| *Roaming-IndicatorList* | • Optional parameter indicating Roaming Indicator List<br>• See roamIndList for more information |

| | |
|---|---|
| *defaultRoamInd* | • Optional parameter indicating Default Roaming Indicator<br>• Values:<br>– 0x00 - Roaming<br>– 0x01 - Home |
| *Gpp2TimeZone* | • Optional parameter indicating 3GPP2 Time Zone<br>• See qaQmi3Gpp2TimeZone for more information |
| *CDMA_P_Rev* | • Optional parameter indicating CDMA P_Rev in use |
| *GppTimeZone* | • Optional parameter indicating Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min. (signed value). |
| *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 |
| *Lac* | • Optional parameter indicating 3GPP Location Area Code |
| *CellID* | • Optional parameter indicating 3GPP Cell ID |
| *concSvcInfo* | • Optional parameter indicating 3GPP2 concurrent service Info<br>• Values:<br>– 0x00 - Concurrent service not available<br>– 0x01 - Concurrent service available |
| *PRLInd* | • Optional parameter indicating 3GPP2 PRL Indicator<br>• Values:<br>– 0x00 - System not in PRL<br>– 0x01 - System is in PRL |
| *DTMInd* | • Optional parameter indicating Dual Transfer Mode Indication(GSM Only)<br>• Values:<br>– 0x00 - DTM not supported<br>– 0x01 - DTM supported |
| *DetailedSvcInfo* | • Optional parameter indicating Detailed service information<br>• See detailSvcInfo for more information |

| | |
|---|---|
| *CDMASystem-InfoExt* | • Optional parameter indicating CDMA System Info Ext<br>• See CDMASysInfoExt for more information |
| *hdrPersonality* | • Optional parameter indicating HDR Personality Information<br>• Values:<br>– 0x00 - Unknown<br>– 0x01 - HRPD<br>– 0x02 - eHRPD |
| *trackAreaCode* | • Optional parameter indicating Tracking area code information for LTE |
| *CallBarStatus* | • Optional parameter indicating Call Barring Status<br>• See callBarStatus for more information |

### 8.825.2 Field Documentation

#### 8.825.2.1 WORD qaQmiServingSystemParam::BasestationID

#### 8.825.2.2 ULONG qaQmiServingSystemParam::BasestationLatitude

#### 8.825.2.3 ULONG qaQmiServingSystemParam::BasestationLongitude

#### 8.825.2.4 callBarStatus qaQmiServingSystemParam::CallBarStatus

#### 8.825.2.5 BYTE qaQmiServingSystemParam::CDMA_P_Rev

#### 8.825.2.6 CDMASysInfoExt qaQmiServingSystemParam::CDMASystemInfoExt

#### 8.825.2.7 ULONG qaQmiServingSystemParam::CellID

#### 8.825.2.8 BYTE qaQmiServingSystemParam::concSvcInfo

#### 8.825.2.9 currentPLMN qaQmiServingSystemParam::CurrentPLMN

#### 8.825.2.10 dataSrvCapabilities qaQmiServingSystemParam::DataSrvCapabilities

#### 8.825.2.11 BYTE qaQmiServingSystemParam::defaultRoamInd

#### 8.825.2.12 detailSvcInfo qaQmiServingSystemParam::DetailedSvcInfo

#### 8.825.2.13 BYTE qaQmiServingSystemParam::DTMInd

#### 8.825.2.14 qaQmi3Gpp2TimeZone qaQmiServingSystemParam::Gpp2TimeZone

#### 8.825.2.15 BYTE qaQmiServingSystemParam::GppNetworkDSTAdjustment

#### 8.825.2.16 BYTE qaQmiServingSystemParam::GppTimeZone

#### 8.825.2.17 BYTE qaQmiServingSystemParam::hdrPersonality

#### 8.825.2.18 WORD qaQmiServingSystemParam::Lac

#### 8.825.2.19 WORD qaQmiServingSystemParam::NetworkID

#### 8.825.2.20 BYTE qaQmiServingSystemParam::PRLInd

#### 8.825.2.21 BYTE qaQmiServingSystemParam::roamIndicatorVal

#### 8.825.2.22 roamIndList qaQmiServingSystemParam::RoamingIndicatorList

#### 8.825.2.23 servSystem qaQmiServingSystemParam::ServingSystem

#### 8.825.2.24 WORD qaQmiServingSystemParam::SystemID

#### 8.825.2.25 WORD qaQmiServingSystemParam::trackAreaCode

## 8.826 QmiCbkCatEventStatusReportInd Struct Reference

### Data Fields

- BYTE event_Index
- struct CatCommonEventTlv CCETlv [11]

### 8.826.1 Field Documentation

#### 8.826.1.1 struct CatCommonEventTlv QmiCbkCatEventStatusReportInd::CCETlv[11]

#### 8.826.1.2 BYTE QmiCbkCatEventStatusReportInd::event_Index

## 8.827 QmiCbkLocBestAvailPosInd Struct Reference

### Data Fields

- ULONG status
- ULONG * pXid
- ULONGLONG * pLatitude
- ULONGLONG * pLongitude
- ULONG * pHorUncCircular
- ULONG * pAltitudeWrtEllipsoid
- ULONG * pVertUnc
- ULONGLONG * pTimestampUtc
- ULONG * pTimeUnc
- ULONG * pHorUncEllipseSemiMinor
- ULONG * pHorUncEllipseSemiMajor
- ULONG * pHorUncEllipseOrientAzimuth
- BYTE * pHorCirConf
- BYTE * pHorEllpConf
- ULONG * pHorReliability
- ULONG * pSpeedHorizontal
- ULONG * pSpeedUnc
- ULONG * pAltitudeWrtMeanSeaLevel

- BYTE * pVertConfidence
- ULONG * pVertReliability
- ULONG * pSpeedVertical
- ULONG * pSpeedVerticalUnc
- ULONG * pHeading
- ULONG * pHeadingUnc
- ULONG * pMagneticDeviation
- ULONG * pTechnologyMask
- precisionDilution * pPrecisionDilution
- gpsTime * pGpsTime
- ULONG * pTimeSrc
- sensorDataUsage * pSensorDataUsage
- svUsedforFix * pSvUsedforFix

### 8.827.1 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 |
| *xid* | Transaction ID that was specified in the Get Best Available Position request. |
| *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 |

| | |
|---|---|
| *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 |
| *pHorUncCircular* | • Horizontal position uncertainty.<br>• Units - Meters |
| *pAltitudeWrt-Ellipsoid* | • Altitude With Respect to WGS84 Ellipsoid.<br>• Units - Meters<br>• Range -500 to 15883 |
| *pVertUnc* | • Vertical uncertainty.<br>• Units - Meters |
| *pTimestampUtc* | • UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970 |
| *pTimeUnc* | • Time uncertainty.<br>• Units - Milliseconds |
| *pHorUncEllipse-SemiMinor* | • Semi-minor axis of horizontal elliptical uncertainty.<br>• Units - Meters |
| *pHorUncEllipse-SemiMajor* | • Semi-major axis of horizontal elliptical uncertainty.<br>• Units: Meters |
| *pHorUncEllipse-OrientAzimuth* | • Elliptical horizontal uncertainty azimuth of orientation.<br>• Units - Decimal degrees<br>• Range - 0 to 180 |
| *pHorCirConf* | • Horizontal circular uncertainty confidence<br>• Units: Precent<br>• Range: 0 to 99 |
| *pHorEllpConf* | • Horizontal elliptical uncertainty confidence<br>• Units: Precent<br>• Range: 0 to 99 |

| | |
|---|---|
| *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 |
| *pSpeed-Horizontal* | • Horizontal speed.<br>• Units - Meters/second |
| *pSpeedUnc* | • 3-D Speed uncertainty.<br>• Units - Meters/second. |
| *pAltitudeWrt-MeanSeaLevel* | • Altitude With Respect to Sea Level.<br>• Units - Meters |
| *pVertConfidence* | • Vertical uncertainty confidence.<br>• Units - Percentage<br>• Range 0 to 99 |
| *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 |
| *pSpeedVertical* | • Vertical speed.<br>• Units - Meters/second |
| *pSpeedVertical-Unc* | • Veritical speed<br>• Units: Meters/second |
| *pHeading* | • Heading.<br>• Units - Degree<br>• Range 0 to 359.999 |
| *pHeadingUnc* | • Heading uncertainty.<br>• Units - Degree<br>• Range 0 to 359.999 |

| | |
|---|---|
| *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. |
| *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 |
| *-pPrecision-Dilution* | • See precisionDilution for more information |
| *-pGpsTime* | • See gpsTime for more information |
| *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. |
| *-pSensorData-Usage* | • See sensorDataUsage for more information |
| *-pSvUsedforFix* | • See svUsedforFix for more information |

### 8.827.2 Field Documentation

#### 8.827.2.1 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtEllipsoid

#### 8.827.2.2 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtMeanSeaLevel

#### 8.827.2.3 gpsTime* QmiCbkLocBestAvailPosInd::pGpsTime

#### 8.827.2.4 ULONG* QmiCbkLocBestAvailPosInd::pHeading

#### 8.827.2.5 ULONG* QmiCbkLocBestAvailPosInd::pHeadingUnc

#### 8.827.2.6 BYTE* QmiCbkLocBestAvailPosInd::pHorCirConf

#### 8.827.2.7 BYTE* QmiCbkLocBestAvailPosInd::pHorEllpConf

#### 8.827.2.8 ULONG* QmiCbkLocBestAvailPosInd::pHorReliability

#### 8.827.2.9 ULONG* QmiCbkLocBestAvailPosInd::pHorUncCircular

#### 8.827.2.10 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseOrientAzimuth

#### 8.827.2.11 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMajor

#### 8.827.2.12 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMinor

#### 8.827.2.13 ULONGLONG* QmiCbkLocBestAvailPosInd::pLatitude

#### 8.827.2.14 ULONGLONG* QmiCbkLocBestAvailPosInd::pLongitude

#### 8.827.2.15 ULONG* QmiCbkLocBestAvailPosInd::pMagneticDeviation

#### 8.827.2.16 precisionDilution* QmiCbkLocBestAvailPosInd::pPrecisionDilution

#### 8.827.2.17 sensorDataUsage* QmiCbkLocBestAvailPosInd::pSensorDataUsage

#### 8.827.2.18 ULONG* QmiCbkLocBestAvailPosInd::pSpeedHorizontal

#### 8.827.2.19 ULONG* QmiCbkLocBestAvailPosInd::pSpeedUnc

#### 8.827.2.20 ULONG* QmiCbkLocBestAvailPosInd::pSpeedVertical

#### 8.827.2.21 ULONG* QmiCbkLocBestAvailPosInd::pSpeedVerticalUnc

#### 8.827.2.22 svUsedforFix* QmiCbkLocBestAvailPosInd::pSvUsedforFix

#### 8.827.2.23 ULONG* QmiCbkLocBestAvailPosInd::pTechnologyMask

#### 8.827.2.24 ULONG* QmiCbkLocBestAvailPosInd::pTimeSrc

#### 8.827.2.25 ULONGLONG* QmiCbkLocBestAvailPosInd::pTimestampUtc

#### 8.827.2.26 ULONG* QmiCbkLocBestAvailPosInd::pTimeUnc

#### 8.827.2.27 BYTE* QmiCbkLocBestAvailPosInd::pVertConfidence

#### 8.827.2.28 ULONG* QmiCbkLocBestAvailPosInd::pVertReliability

#### 8.827.2.29 ULONG* QmiCbkLocBestAvailPosInd::pVertUnc

#### 8.827.2.30 ULONG* QmiCbkLocBestAvailPosInd::pXid

#### 8.827.2.31 ULONG QmiCbkLocBestAvailPosInd::status

## 8.828 QmiCbkLocCradleMountInd Struct Reference

### Data Fields

- ULONG cradleMountConfigStatus

### 8.828.1 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 |

### 8.828.2 Field Documentation

#### 8.828.2.1 ULONG QmiCbkLocCradleMountInd::cradleMountConfigStatus

## 8.829 QmiCbkLocEngineStateInd Struct Reference

### Data Fields

- ULONG engineState

### 8.829.1 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 |

### 8.829.2 Field Documentation

#### 8.829.2.1 ULONG QmiCbkLocEngineStateInd::engineState

## 8.830 QmiCbkLocEventTimeSyncInd Struct Reference

### Data Fields

- ULONG timeSyncRefCounter

### 8.830.1 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. |

### 8.830.2 Field Documentation

#### 8.830.2.1 ULONG QmiCbkLocEventTimeSyncInd::timeSyncRefCounter

## 8.831 QmiCbkLocGetServerInd Struct Reference

### Data Fields

- ULONG serverStatus
- ULONG serverType
- IPv4Info * pIPv4AddrInfo
- IPv6Info * pIPv6AddrInfo
- urlAddr * pURL

### 8.831.1 Detailed Description

This structure contains LOC Get Server field.

**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 |
| *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 |
| *pIPv4AddrInfo* | • See IPv4Info for more information. |
| *pIPv6AddrInfo* | • See IPv6Info for more information. |
| *pURL* | • See urlAddr for more information. |

## 8.831.2 Field Documentation

### 8.831.2.1 IPv4Info* QmiCbkLocGetServerInd::pIPv4AddrInfo

### 8.831.2.2 IPv6Info* QmiCbkLocGetServerInd::pIPv6AddrInfo

### 8.831.2.3 urlAddr* QmiCbkLocGetServerInd::pURL

### 8.831.2.4 ULONG QmiCbkLocGetServerInd::serverStatus

### 8.831.2.5 ULONG QmiCbkLocGetServerInd::serverType

## 8.832 QmiCbkLocInjectPositionInd Struct Reference

### Data Fields

- ULONG status

### 8.832.1 Detailed Description

Contain the parameters passed for SetLocInjectPositionCallback by the device.

**Parameters**

| | |
|---|---|
| *status* | • UTC Position Injection Status<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 |

**Note**

None

### 8.832.2 Field Documentation

#### 8.832.2.1 ULONG QmiCbkLocInjectPositionInd::status

## 8.833 QmiCbkLocInjectSensorDataInd Struct Reference

### Data Fields

- ULONG injectSensorDataStatus
- ULONG * pOpaqueIdentifier
- BYTE * pAccelSamplesAccepted
- BYTE * pGyroSamplesAccepted
- BYTE * pAccelTempSamplesAccepted
- BYTE * pGyroTempSamplesAccepted

### 8.833.1 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 |
| *pOpaqueIdentifier* | • Sent in by the client echoed so the client can relate the indication to the request. |
| *pAccelSamplesAccepted* | • 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. |
| *pGyroSamplesAccepted* | • 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. |
| *pAccelTempSamplesAccepted* | • 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. |
| *pGyroTempSamplesAccepted* | • 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. |

### 8.833.2 Field Documentation

#### 8.833.2.1 ULONG QmiCbkLocInjectSensorDataInd::injectSensorDataStatus

#### 8.833.2.2 BYTE* QmiCbkLocInjectSensorDataInd::pAccelSamplesAccepted

#### 8.833.2.3 BYTE* QmiCbkLocInjectSensorDataInd::pAccelTempSamplesAccepted

#### 8.833.2.4 BYTE* QmiCbkLocInjectSensorDataInd::pGyroSamplesAccepted

#### 8.833.2.5 BYTE* QmiCbkLocInjectSensorDataInd::pGyroTempSamplesAccepted

#### 8.833.2.6 ULONG* QmiCbkLocInjectSensorDataInd::pOpaqueIdentifier

## 8.834 QmiCbkLocInjectTimeInd Struct Reference

### Data Fields

- ULONG injectTimeSyncStatus

### 8.834.1 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 |

### 8.834.2 Field Documentation

#### 8.834.2.1 ULONG QmiCbkLocInjectTimeInd::injectTimeSyncStatus

## 8.835 QmiCbkLocInjectUTCTimeInd Struct Reference

### Data Fields

- ULONG status

### 8.835.1 Detailed Description

Contain the parameters passed for SetLocInjectUTCTimeCallback by the device.

**Parameters**

| | |
|---|---|
| *status* | • Status of the UTC Time Injection 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 |

**Note**

None

### 8.835.2 Field Documentation

#### 8.835.2.1 ULONG QmiCbkLocInjectUTCTimeInd::status

## 8.836 QmiCbkLocPositionReportInd Struct Reference

### Data Fields

- ULONG sessionStatus
- BYTE sessionId
- ULONGLONG * pLatitude
- ULONGLONG * pLongitude
- ULONG * pHorUncCircular
- ULONG * pHorUncEllipseSemiMinor
- ULONG * pHorUncEllipseSemiMajor
- ULONG * pHorUncEllipseOrientAzimuth
- BYTE * pHorConfidence
- ULONG * pHorReliability
- ULONG * pSpeedHorizontal
- ULONG * pSpeedUnc
- ULONG * pAltitudeWrtEllipsoid
- ULONG * pAltitudeWrtMeanSeaLevel
- ULONG * pVertUnc
- BYTE * pVertConfidence
- ULONG * pVertReliability
- ULONG * pSpeedVertical
- ULONG * pHeading
- ULONG * pHeadingUnc
- ULONG * pMagneticDeviation
- ULONG * pTechnologyMask
- precisionDilution * pPrecisionDilution
- ULONGLONG * pTimestampUtc
- BYTE * pLeapSeconds
- gpsTime * pGpsTime
- ULONG * pTimeUnc

- ULONG * pTimeSrc
- sensorDataUsage * pSensorDataUsage
- ULONG * pFixId
- svUsedforFix * pSvUsedforFix
- BYTE * pAltitudeAssumed

### 8.836.1 Detailed Description

This structure contains Event Position Report

**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 |
| *sessionId* | • ID of the session that was specified in the Start request<br>• Range - 0 to 255 |
| *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 |
| *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 |
| *pHorUncCircular* | • Horizontal position uncertainty.<br>• Units - Meters |
| *pHorUncEllipse-SemiMinor* | • Semi-minor axis of horizontal elliptical uncertainty.<br>• Units - Meters |

| | |
|---|---|
| *pHorUncEllipse-SemiMajor* | • Semi-major axis of horizontal elliptical uncertainty.<br>• Units: Meters |
| *pHorUncEllipse-OrientAzimuth* | • Elliptical horizontal uncertainty azimuth of orientation.<br>• Units - Decimal degrees<br>• Range - 0 to 180 |
| *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 |
| *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 |
| *pSpeed-Horizontal* | • Horizontal speed.<br>• Units - Meters/second |
| *pSpeedUnc* | • 3-D Speed uncertainty.<br>• Units - Meters/second. |
| *pAltitudeWrt-Ellipsoid* | • Altitude With Respect to WGS84 Ellipsoid.<br>• Units - Meters<br>• Range -500 to 15883 |
| *pAltitudeWrt-MeanSeaLevel* | • Altitude With Respect to Sea Level.<br>• Units - Meters |
| *pVertUnc* | • Vertical uncertainty.<br>• Units - Meters |
| *pVertConfidence* | • Vertical uncertainty confidence.<br>• Units - Percentage<br>• Range 0 to 99 |

| | |
|---|---|
| *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 |
| *pSpeedVertical* | • Vertical speed.<br>• Units - Meters/second |
| *pHeading* | • Heading.<br>• Units - Degree<br>• Range 0 to 359.999 |
| *pHeadingUnc* | • Heading uncertainty.<br>• Units - Degree<br>• Range 0 to 359.999 |
| *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. |
| *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 |
| *-pPrecision-Dilution* | • See precisionDilution for more information |
| *pTimestampUtc* | • UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970 |
| *pLeapSeconds* | • Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds.<br>• Units - Seconds |
| *-pGpsTime* | • See gpsTime for more information |

| | |
|---|---|
| *pTimeUnc* | • Time uncertainty.<br>• Units - Milliseconds |
| *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. |
| *-pSensorData-Usage* | • See sensorDataUsage for more information |
| *pFixId* | • Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session. |
| *-pSvUsedforFix* | • See svUsedforFix for more information |
| *pAltitude-Assumed* | • Indicates whether altitude is assumed or calculated. |

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

## 8.836.2 Field Documentation

### 8.836.2.1 BYTE* QmiCbkLocPositionReportInd::pAltitudeAssumed

### 8.836.2.2 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtEllipsoid

### 8.836.2.3 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtMeanSeaLevel

8.836.2.4 **ULONG*** QmiCbkLocPositionReportInd::pFixId

8.836.2.5 **gpsTime*** QmiCbkLocPositionReportInd::pGpsTime

8.836.2.6 **ULONG*** QmiCbkLocPositionReportInd::pHeading

8.836.2.7 **ULONG*** QmiCbkLocPositionReportInd::pHeadingUnc

8.836.2.8 **BYTE*** QmiCbkLocPositionReportInd::pHorConfidence

8.836.2.9 **ULONG*** QmiCbkLocPositionReportInd::pHorReliability

8.836.2.10 **ULONG*** QmiCbkLocPositionReportInd::pHorUncCircular

8.836.2.11 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseOrientAzimuth

8.836.2.12 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMajor

8.836.2.13 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMinor

8.836.2.14 **ULONGLONG*** QmiCbkLocPositionReportInd::pLatitude

8.836.2.15 **BYTE*** QmiCbkLocPositionReportInd::pLeapSeconds

8.836.2.16 **ULONGLONG*** QmiCbkLocPositionReportInd::pLongitude

8.836.2.17 **ULONG*** QmiCbkLocPositionReportInd::pMagneticDeviation

8.836.2.18 **precisionDilution*** QmiCbkLocPositionReportInd::pPrecisionDilution

8.836.2.19 **sensorDataUsage*** QmiCbkLocPositionReportInd::pSensorDataUsage

8.836.2.20 **ULONG*** QmiCbkLocPositionReportInd::pSpeedHorizontal

8.836.2.21 **ULONG*** QmiCbkLocPositionReportInd::pSpeedUnc

8.836.2.22 **ULONG*** QmiCbkLocPositionReportInd::pSpeedVertical

8.836.2.23 **svUsedforFix*** QmiCbkLocPositionReportInd::pSvUsedforFix

8.836.2.24 **ULONG*** QmiCbkLocPositionReportInd::pTechnologyMask

8.836.2.25 **ULONG*** QmiCbkLocPositionReportInd::pTimeSrc

8.836.2.26 **ULONGLONG*** QmiCbkLocPositionReportInd::pTimestampUtc

8.836.2.27 **ULONG*** QmiCbkLocPositionReportInd::pTimeUnc

8.836.2.28 **BYTE*** QmiCbkLocPositionReportInd::pVertConfidence

8.836.2.29 **ULONG*** QmiCbkLocPositionReportInd::pVertReliability

8.836.2.30 **ULONG*** QmiCbkLocPositionReportInd::pVertUnc

8.836.2.31 **BYTE** QmiCbkLocPositionReportInd::sessionId

#### 8.836.2.32 ULONG QmiCbkLocPositionReportInd::sessionStatus

## 8.837 QmiCbkLocSensorStreamingInd Struct Reference

### Data Fields

- accelAcceptReady * pAccelAcceptReady
- gyroAcceptReady * pGyroAcceptReady
- accelTempAcceptReady * pAccelTempAcceptReady
- gyroTempAcceptReady * pGyroTempAcceptReady

### 8.837.1 Detailed Description

This structure contains LOC Event Sensor Streaming Ready Status

**Parameters**

| | |
|---|---|
| *-pAccelAccept-Ready* | • See accelAcceptReady for more information |
| *-pGyroAccept-Ready* | • See gyroAcceptReady for more information |
| *-pAccelTemp-AcceptReady* | • See accelTempAcceptReady for more information |
| *-pGyroTemp-AcceptReady* | • See gyroTempAcceptReady for more information |

### 8.837.2 Field Documentation

#### 8.837.2.1 accelAcceptReady* QmiCbkLocSensorStreamingInd::pAccelAcceptReady

#### 8.837.2.2 accelTempAcceptReady* QmiCbkLocSensorStreamingInd::pAccelTempAcceptReady

#### 8.837.2.3 gyroAcceptReady* QmiCbkLocSensorStreamingInd::pGyroAcceptReady

#### 8.837.2.4 gyroTempAcceptReady* QmiCbkLocSensorStreamingInd::pGyroTempAcceptReady

## 8.838 QmiCbkLocSetExtPowerConfigInd Struct Reference

### Data Fields

- ULONG status

### 8.838.1 Detailed Description

This structure contains LOC Set External Power Config Status

**Parameters**

| | |
|---|---|
| *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 |

### 8.838.2 Field Documentation

#### 8.838.2.1 ULONG QmiCbkLocSetExtPowerConfigInd::status

## 8.839 QmiCbkLocSetServerInd Struct Reference

### Data Fields

- ULONG serverStatus

### 8.839.1 Detailed Description

This structure contains LOC Set Server field.

**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 |

### 8.839.2 Field Documentation

#### 8.839.2.1 ULONG QmiCbkLocSetServerInd::serverStatus

## 8.840 QmiCbkNasLTECphyCaInfo Struct Reference

### Data Fields

- PhyCaAggScellIndType sPhyCaAggScellIndType
- PhyCaAggScellDlBw sPhyCaAggScellDlBw
- PhyCaAggScellInfo sPhyCaAggScellInfo
- PhyCaAggPcellInfo sPhyCaAggPcellInfo
- PhyCaAggScellIndex sPhyCaAggScellIndex

### 8.840.1 Detailed Description

Structure for storing the LTEC PHY CA indication parameters.

**Parameters**

| | |
|---|---|
| *pPhyCaAgg-ScellIndType* | • See PhyCaAggScellIndType for more information. |
| *sPhyCaAgg-ScellDlBw* | • See PhyCaAggScellDlBw for more information. |
| *sPhyCaAgg-ScellInfo* | • See PhyCaAggScellInfo for more information. |
| *sPhyCaAgg-PcellInfo* | • See PhyCaAggPcellInfo for more information. |
| *sPhyCaAgg-ScellIndex* | • See PhyCaAggScellIndex for more information. |

### 8.840.2 Field Documentation

#### 8.840.2.1 PhyCaAggPcellInfo QmiCbkNasLTECphyCaInfo::sPhyCaAggPcellInfo

#### 8.840.2.2 PhyCaAggScellDlBw QmiCbkNasLTECphyCaInfo::sPhyCaAggScellDlBw

#### 8.840.2.3 PhyCaAggScellIndex QmiCbkNasLTECphyCaInfo::sPhyCaAggScellIndex

#### 8.840.2.4 PhyCaAggScellIndType QmiCbkNasLTECphyCaInfo::sPhyCaAggScellIndType

#### 8.840.2.5 PhyCaAggScellInfo QmiCbkNasLTECphyCaInfo::sPhyCaAggScellInfo

## 8.841 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference

### Data Fields

- struct sessionInfoTlv SITlv

### 8.841.1 Field Documentation

#### 8.841.1.1 struct sessionInfoTlv QmiCbkSwiOmaDmEventStatusReportInd::SITlv

## 8.842 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference

### Data Fields

- struct sessionInfoTlvExt SITlv

### 8.842.1 Field Documentation

#### 8.842.1.1 struct sessionInfoTlvExt QmiCbkSwiOmaDmEventStatusReportIndExt::SITlv

## 8.843 QmiCbkTmdMitiLvlRptInd Struct Reference

### Data Fields

- _MitigationDevInfo MitigationDevInfo
- BYTE currentMitigationLvl

### 8.843.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

**Parameters**

| | |
|---|---|
| *MitigationDevInfo* | • See MitigationDevInfo for more information. |
| *currentMitigationLvl* | • Current Thermal Mitigation Level |

### 8.843.2 Field Documentation

#### 8.843.2.1 BYTE QmiCbkTmdMitiLvlRptInd::currentMitigationLvl

#### 8.843.2.2 _MitigationDevInfo QmiCbkTmdMitiLvlRptInd::MitigationDevInfo

## 8.844 QmiCbkWdsStatisticsIndState Struct Reference

### Data Fields

- DataULongTlv TxOkConutTlv
- DataULongTlv RxOkConutTlv
- DataULongLongTlv TxOkByteCountTlv
- DataULongLongTlv RxOkByteCountTlv
- DataULongTlv TxDropConutTlv
- DataULongTlv RxDropConutTlv

### 8.844.1 Detailed Description

WDS Pkt 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. |

### 8.844.2 Field Documentation

#### 8.844.2.1 DataULongTlv QmiCbkWdsStatisticsIndState::RxDropConutTlv

#### 8.844.2.2 DataULongLongTlv QmiCbkWdsStatisticsIndState::RxOkByteCountTlv

#### 8.844.2.3 DataULongTlv QmiCbkWdsStatisticsIndState::RxOkConutTlv

#### 8.844.2.4 DataULongTlv QmiCbkWdsStatisticsIndState::TxDropConutTlv

#### 8.844.2.5 DataULongLongTlv QmiCbkWdsStatisticsIndState::TxOkByteCountTlv

#### 8.844.2.6 DataULongTlv QmiCbkWdsStatisticsIndState::TxOkConutTlv

## 8.845 qmifwinfo_s Struct Reference

### Data Fields

- union {
  struct fwinfo_s g
  struct slqsfwinfo_s s
  } dev

### 8.845.1 Detailed Description

Top level structure for storing information about firmware images. union of structures depending on device type, MC77xx or MC83xx

**Parameters**

| | |
|---|---|
| *g* | - structure for MC83xx devices |
| *s* | - structure for devices with SPKG CWE file support |

- List of various Firmware Images Supported

```
Technology  Initials  Carrier                    Region              Network Technology
----------  --------  -------------------------  ------------------  --------------------
D3600       S         eGOBI_IMG_CAR_SPRINT       eGOBI_IMG_REG_NA    eGOBI_IMG_TECH_CDMA
D3600       V         eGOBI_IMG_CAR_VERIZON      eGOBI_IMG_REG_NA    eGOBI_IMG_TECH_CDMA
D3600       C         eGOBI_IMG_CAR_CHINA_TELECOM eGOBI_IMG_REG_ASIA  eGOBI_IMG_TECH_CDMA
D3600       G         eGOBI_IMG_CAR_GENERIC_CDMA eGOBI_IMG_REG_GLOBAL eGOBI_IMG_TECH_CDMA (item for Generic
D3600       H         eGOBI_IMG_CAR_GENERIC_CDMA eGOBI_IMG_REG_GLOBAL eGOBI_IMG_TECH_CDMA (item for Generic
D3200       V         eGOBI_IMG_CAR_VODAFONE     eGOBI_IMG_REG_NA    eGOBI_IMG_TECH_UMTS
D3200       A         eGOBI_IMG_CAR_ATT          eGOBI_IMG_REG_NA    eGOBI_IMG_TECH_UMTS
D3200       L         eGOBI_IMG_CAR_TMOBILE      eGOBI_IMG_REG_EU    eGOBI_IMG_TECH_UMTS
D3200       G         eGOBI_IMG_CAR_GENERIC      eGOBI_IMG_REG_GLOBAL eGOBI_IMG_TECH_UMTS
D3200       H         eGOBI_IMG_CAR_TELEFONICA   eGOBI_IMG_REG_EU    eGOBI_IMG_TECH_UMTS
D3200       I         eGOBI_IMG_CAR_TELCOM_ITALIA eGOBI_IMG_REG_NA   eGOBI_IMG_TECH_UMTS
D3200       O         eGOBI_IMG_CAR_ORANGE       eGOBI_IMG_REG_NA    eGOBI_IMG_TECH_UMTS
D3200       U         eGOBI_IMG_CAR_GENERIC      eGOBI_IMG_REG_GLOBAL eGOBI_IMG_TECH_UMTS
D3200       R         eGOBI_IMG_CAR_ROGERS       eGOBI_IMG_REG_NA    eGOBI_IMG_TECH_UMTS
D3600       A         eGOBI_IMG_CAR_AERIS        eGOBI_IMG_REG_NA    eGOBI_IMG_TECH_CDMA
```

**See Also**

fwinfo_s
slqsfwinfo_s

### 8.845.2 Field Documentation

#### 8.845.2.1 union { ... } qmifwinfo_s::dev

#### 8.845.2.2 struct fwinfo_s qmifwinfo_s::g

#### 8.845.2.3 struct slqsfwinfo_s qmifwinfo_s::s

## 8.846 QmiNas3GppNetworkInfo Struct Reference

### Data Fields

- WORD pMCC
- WORD pMNC
- ULONG pInUse
- ULONG pRoaming
- ULONG pForbidden
- ULONG pPreferred
- CHAR pDesription [255]

### 8.846.1 Detailed Description

This structure contains the PerformNetworkScan response parameters. This structure will hold the array of the netwok scan information.

**Parameters**

| | |
|---|---|
| *pMCC* | • Mobile Country Code |
| *pMNC* | • Mobile Networ Code |
| *pInUse* | • Is the Network the current serving Network<br>– 0 – Unknown<br>– 1 – Current serving network<br>– 2 – Not current serving network, available |
| *pRoaming* | • Home/Roam Status of the Network<br>– 0 – Unknown<br>– 1 – Home<br>– 2 – Roam |
| *pForbidden* | • Is the Network in the forbidden network list<br>– 0 – Unknown<br>– 1 – Forbidden<br>– 2 – Not Forbidden |
| *pPreferred* | • Is the Network in the Preferred network list<br>– 0 – Unknown<br>– 1 – Preferred<br>– 2 – Not Preferred |
| *pDesription* | • Network Name/Description |

### 8.846.2 Field Documentation

#### 8.846.2.1 CHAR QmiNas3GppNetworkInfo::pDesription[255]

#### 8.846.2.2 ULONG QmiNas3GppNetworkInfo::pForbidden

#### 8.846.2.3 ULONG QmiNas3GppNetworkInfo::pInUse

#### 8.846.2.4 WORD QmiNas3GppNetworkInfo::pMCC

#### 8.846.2.5 WORD QmiNas3GppNetworkInfo::pMNC

#### 8.846.2.6 ULONG QmiNas3GppNetworkInfo::pPreferred

#### 8.846.2.7 ULONG QmiNas3GppNetworkInfo::pRoaming

## 8.847 QmiNasGetRFBandInfoResp Struct Reference

### Data Fields

- struct qmTlvResult results
- BYTE * pInstancesSize
- struct RFBandInfoElements * pRFBandInfoElements
- RfDedicatedBandInfo * pRfDedicatedBandInfo
- RfBandInfoExtFormat * pRfBandInfoExtFormat
- RfBandwidthInfo * pRfBandwidthInfo
- LTEOperationMode * pLTEOperationMode

### 8.847.1 Field Documentation

#### 8.847.1.1 BYTE* QmiNasGetRFBandInfoResp::pInstancesSize

#### 8.847.1.2 LTEOperationMode* QmiNasGetRFBandInfoResp::pLTEOperationMode

#### 8.847.1.3 struct RFBandInfoElements* QmiNasGetRFBandInfoResp::pRFBandInfoElements

#### 8.847.1.4 RfBandInfoExtFormat* QmiNasGetRFBandInfoResp::pRfBandInfoExtFormat

#### 8.847.1.5 RfBandwidthInfo* QmiNasGetRFBandInfoResp::pRfBandwidthInfo

#### 8.847.1.6 RfDedicatedBandInfo* QmiNasGetRFBandInfoResp::pRfDedicatedBandInfo

#### 8.847.1.7 struct qmTlvResult QmiNasGetRFBandInfoResp::results

## 8.848 QmiNasPerformNetworkScanResp Struct Reference

### Data Fields

- struct qmTlvResult results
- BYTE * pInstanceSize
- struct QmiNas3GppNetworkInfo * pInstances

### 8.848.1 Field Documentation

#### 8.848.1.1 struct QmiNas3GppNetworkInfo* QmiNasPerformNetworkScanResp::pInstances

#### 8.848.1.2 BYTE* QmiNasPerformNetworkScanResp::pInstanceSize

#### 8.848.1.3 struct qmTlvResult QmiNasPerformNetworkScanResp::results

## 8.849 qmiSmsMessageList Struct Reference

### Data Fields

- uint32_t messageIndex
- uint32_t messageTag

### 8.849.1 Detailed Description

This structure contains SMS message list.

**Parameters**

| | |
|---|---|
| *messageIndex* | • Message index of each matched message |
| *messageTag* | • Messagetag |

### 8.849.2 Field Documentation

#### 8.849.2.1 uint32_t qmiSmsMessageList::messageIndex

#### 8.849.2.2 uint32_t qmiSmsMessageList::messageTag

## 8.850 qmiWDSDataBearerTechnology Struct Reference

### Data Fields

- uint8_t currentNetwork
- uint32_t ratMask
- uint32_t soMask

### 8.850.1 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]* | • Radio Access Technology (RAT) mask to indicate the type of technology (RAT mask value of zero indicates that this field is ignored)<br>– 0x8000 - NULL Bearer<br>– 0x0000 - DO_NOT_CARE CDMA RAT mask<br>– 0x01 - CDMA_1X<br>– 0x02 - EVDO_REV0<br>– 0x04 - EVDO_REVA UMTS RAT mask<br>– 0x01 - WCDMA<br>– 0x02 - GPRS<br>– 0x04 - HSDPA<br>– 0x08 - HSUPA<br>– 0x10 - EDGE<br>– 0x20 - LTE<br>– 0x40 - HSDPA+<br>– 0x80 - DC_HSDPA+<br>– 0xffffffff - Invalid data. |

| | |
|---|---|
| *soMask[OUT]* | • Service Option (SO) mask to indicate the SO or type of application (SO mask value of zero indicates that this field is ignored)<br>– 0x00 - DO_NOT_CARE CDMA 1X SO mask<br>– 0x01 - CDMA_1X_IS95<br>– 0x02 - CDMA_1X_IS2000<br>– 0x04 - CDMA_1X_IS2000_REL_A CDMA EV-DO Rev A SO mask<br>– 0x01 - EVDO_REVA_DPA<br>– 0x02 - EVDO_REVA_MFPA<br>– 0x04 - EVDO_REVA_EMPA<br>– 0x08 - EVDO_REVA_EMPA_EHRPD<br>– 0xffffffff - Invalid data. |

### 8.850.2 Field Documentation

#### 8.850.2.1 uint8_t qmiWDSDataBearerTechnology::currentNetwork

#### 8.850.2.2 uint32_t qmiWDSDataBearerTechnology::ratMask

#### 8.850.2.3 uint32_t qmiWDSDataBearerTechnology::soMask

## 8.851 QmiWdsIpAddressInfo Struct Reference

### Data Fields

- ULONG * pIPAddressV4
- USHORT * pIPAddressV6
- BYTE * pIPv6prefixlen

### 8.851.1 Detailed Description

**Parameters**

| | |
|---|---|
| *pIPAddressV4[-OUT]* | • Current IPv4 address<br>• default value of 0 if not reported by the device. |
| *pIPAddressV6[-OUT]* | • Current IPv6 address<br>Space for storing the 8 element array of type USHORT for the IPv6 address is allocated by the application.<br>The IP Address is stored in the user supplied buffer as follows:<br>User buffer:<br>[<U0>..<U7>]<br>IPv6 address from the network:<br>1234:2A01:....:5678<br>User buffer contents:<br>U0 corresponds to 1234<br>U1 corresponds to 2A01<br>– ----------— – -—<br>– ----------— – -—<br>U7 corresponds to 5678 |
| *pIPv6prefixlen[-OUT]* | • IPv6 prefix length in number of bits |

### 8.851.2 Field Documentation

#### 8.851.2.1 ULONG* QmiWdsIpAddressInfo::pIPAddressV4

#### 8.851.2.2 USHORT* QmiWdsIpAddressInfo::pIPAddressV6

#### 8.851.2.3 BYTE* QmiWdsIpAddressInfo::pIPv6prefixlen

## 8.852 qmiWdsRunTimeSettings Struct Reference

### Data Fields

- CHAR * pProfileName
- ULONG * pPDPType
- CHAR * pAPNName
- ULONG * pPrimaryDNSV4
- ULONG * pSecondaryDNSV4
- struct UMTSQoS * pUMTSGrantedQoS
- struct GPRSQoS * pGPRSGrantedQoS
- CHAR * pUsername
- ULONG * pAuthentication
- ULONG * pIPAddressV4
- struct ProfileIdentifier * pProfileID
- ULONG * pGWAddressV4
- ULONG * pSubnetMaskV4
- BYTE * pPCSCFAddrPCO
- struct PCSCFIPv4ServerAddressList * pServerAddrList
- struct PCSCFFQDNAddressList * pPCSCFFQDNAddrList
- USHORT * pPrimaryDNSV6

- USHORT * pSecondaryDNSV6
- ULONG * pMtu
- struct DomainNameList * pDomainList
- BYTE * pIPFamilyPreference
- BYTE * pIMCNflag
- WORD * pTechnology
- struct IPV6AddressInfo * pIPV6AddrInfo
- struct IPV6GWAddressInfo * pIPV6GWAddrInfo

### 8.852.1 Detailed Description

This structure contains the WdsRunTimeSettings Information

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

**Parameters**

| | |
|---|---|
| *pProfileName* | • Profile name<br>One or more bytes describing the profile. Description may be a user-defined name for the profile.<br>QMI_ERR_ARG_TOO_LONG is returned if profile_name is too long. |
| *pPDPType* | • PDP type<br>– 0 - PDP-IP (IPv4)<br>– 1 - PDP-PPP<br>– 2 - PDP-IPv6<br>– 3 - PDP-IPv4v6<br>– 0xffffffff - invalid |
| *pAPNName* | • Access point name<br>String parameter that is a logical name used to select the GGSN and external packet data network.<br>If value is NULL or omitted, then the subscription default value is requested.<br>QMI_ERR_ARG_TOO_LONG is returned if the APN name is too long. |
| *pPrimaryDNSV4* | • Primary DNS IPv4 Address |
| *pSecondaryDNSV4* | • Secondary DNS IPv4 Address |
| *pUMTSGrantedQoS* | • UMTS Granted QoS |
| *pGPRSGrantedQoS* | • GPRS Granted QoS |
| *pUsername* | • User name used during data network authentication |

| | |
|---|---|
| *pAuthentication* | • Authentication 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 |
| *pIPAddressV4* | • IPV4 Address assigned to the TE |
| *pProfileID* | • Profile Identifier |
| *pGWAddressV4* | • IPV4 Gateway Address |
| *pSubnetMaskV4* | • IPV4 Subnet Mask |
| *pPCSCFAddrPCO* | • PCSCF address using PCO values<br>– 1 - (TRUE) implies request PCSCF address using PCO<br>– 0 - (FALSE) implies do not request. This is the default value. |
| *pServerAddrList* | • P-CSCF IPv4 Server Address List |
| *pPCSCFFQDNAddrList* | • P-CSCF FQDN Address List |
| *pPrimaryDNSV6* | • Primary DNS IPv6 Address |
| *pSecondaryDNSV6* | • Secondary DNS IPv6 Address |
| *pMtu* | • actual (runtime) MTU size |
| *pDomainList* | • Domain-Name List |
| *pIPFamilyPreference* | • IP family<br>– 0x04 - IPV4 ADDR<br>– 0x06 - IPV6 ADDR |

| | |
|---|---|
| *pIMCNflag* | • IM CN Flag<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *pTechnology* | • Technology<br>– CDMA - 0x8001<br>– UMTS - 0x8004 |
| *pIPV6Address-Info* | • IPV6 Address Information |
| *pIPV6GW-AddressInfo* | • IPV6 Gateway Address Information |

### 8.852.2 Field Documentation

#### 8.852.2.1 CHAR* qmiWdsRunTimeSettings::pAPNName

#### 8.852.2.2 ULONG* qmiWdsRunTimeSettings::pAuthentication

#### 8.852.2.3 struct DomainNameList* qmiWdsRunTimeSettings::pDomainList

#### 8.852.2.4 struct GPRSQoS* qmiWdsRunTimeSettings::pGPRSGrantedQoS

#### 8.852.2.5 ULONG* qmiWdsRunTimeSettings::pGWAddressV4

#### 8.852.2.6 BYTE* qmiWdsRunTimeSettings::pIMCNflag

#### 8.852.2.7 ULONG* qmiWdsRunTimeSettings::pIPAddressV4

#### 8.852.2.8 BYTE* qmiWdsRunTimeSettings::pIPFamilyPreference

#### 8.852.2.9 struct IPV6AddressInfo* qmiWdsRunTimeSettings::pIPV6AddrInfo

#### 8.852.2.10 struct IPV6GWAddressInfo* qmiWdsRunTimeSettings::pIPV6GWAddrInfo

#### 8.852.2.11 ULONG* qmiWdsRunTimeSettings::pMtu

#### 8.852.2.12 BYTE* qmiWdsRunTimeSettings::pPCSCFAddrPCO

#### 8.852.2.13 struct PCSCFFQDNAddressList* qmiWdsRunTimeSettings::pPCSCFFQDNAddrList

#### 8.852.2.14 ULONG* qmiWdsRunTimeSettings::pPDPType

#### 8.852.2.15 ULONG* qmiWdsRunTimeSettings::pPrimaryDNSV4

#### 8.852.2.16 USHORT* qmiWdsRunTimeSettings::pPrimaryDNSV6

#### 8.852.2.17 struct ProfileIdentifier* qmiWdsRunTimeSettings::pProfileID

#### 8.852.2.18 CHAR* qmiWdsRunTimeSettings::pProfileName

#### 8.852.2.19 ULONG* qmiWdsRunTimeSettings::pSecondaryDNSV4

#### 8.852.2.20 USHORT* qmiWdsRunTimeSettings::pSecondaryDNSV6

#### 8.852.2.21 struct PCSCFIPv4ServerAddressList* qmiWdsRunTimeSettings::pServerAddrList

#### 8.852.2.22 ULONG* qmiWdsRunTimeSettings::pSubnetMaskV4

#### 8.852.2.23 WORD* qmiWdsRunTimeSettings::pTechnology

#### 8.852.2.24 struct UMTSQoS* qmiWdsRunTimeSettings::pUMTSGrantedQoS

#### 8.852.2.25 CHAR* qmiWdsRunTimeSettings::pUsername

## 8.853 qos_BindDataPortMuxID_t Struct Reference

### Data Fields

- uint8_t MuxID

### 8.853.1 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 |

### 8.853.2 Field Documentation

#### 8.853.2.1 uint8_t qos_BindDataPortMuxID_t::MuxID

## 8.854 qos_BindDataPortPeripheralEndPointID_t Struct Reference

### Data Fields

- uint32_t EndPointType
- uint32_t IfaceID

### 8.854.1 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. |

### 8.854.2 Field Documentation

#### 8.854.2.1 uint32_t qos_BindDataPortPeripheralEndPointID_t::EndPointType

#### 8.854.2.2 uint32_t qos_BindDataPortPeripheralEndPointID_t::IfaceID

## 8.855 qos_BindDataPortSIODataPort_t Struct Reference

### Data Fields

- uint16_t SIODataPort

### 8.855.1 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 |

### 8.855.2 Field Documentation

#### 8.855.2.1 uint16_t qos_BindDataPortSIODataPort_t::SIODataPort

## 8.856 QosClassID Struct Reference

### Data Fields

- BYTE QCI
- ULONG gDlBitRate
- ULONG maxDlBitRate
- ULONG gUlBitRate
- ULONG maxUlBitRate

### 8.856.1 Detailed Description

structure contains 3GPP LTE QoS parameters

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

**Parameters**

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

### 8.856.2 Field Documentation

#### 8.856.2.1 ULONG QosClassID::gDlBitRate

#### 8.856.2.2 ULONG QosClassID::gUlBitRate

#### 8.856.2.3 ULONG QosClassID::maxDlBitRate

#### 8.856.2.4 ULONG QosClassID::maxUlBitRate

#### 8.856.2.5 BYTE QosClassID::QCI

## 8.857 QosEventInfo Struct Reference

### Data Fields

- ULONG * pDataBearer
- ULONG * pPacketsCountTX
- ULONG * pPacketsCountRX
- ULONGLONG * pTotalBytesTX
- ULONGLONG * pTotalBytesRX

### 8.857.1 Detailed Description

Contains the WDS event information and information about the interface

**Parameters**

| | |
|---|---|
| *pQmiInterface-Info* | • See qaQmiInterfaceInfo for more information |
| *pDataBearer* | • Data bearer technology (NULL if not present)<br>– 0x00 - Indicates that this field is ignored<br>– 0x01 - CDMA 1X<br>– 0x02 - EV-DO Rev 0<br>– 0x03 - GPRS<br>– 0x04 - WCDMA<br>– 0x05 - EV-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 - EV-DO Rev A EHRPD<br>– 0x0C - HSDPA+ and WCDMA<br>– 0x0D - HSDPA+ and HSUPA<br>– 0x0E - DC_HSDPA+ and WCDMA<br>– 0x0F - DC_HSDPA+ and HSUPA<br>– 0x8000 - NULL Bearer<br>– 0xFF - Unknown Technology |
| *pDormancy-Status* | • Dormancy status (NULL if not present)<br>– 1 - traffic channel dormant<br>– 2 - traffic channel active |
| *pPacketsCount-TX* | • Packets transmitted without error (NULL if not present) |
| *pPacketsCount-RX* | • Packets received without error (NULL if not present) |
| *pTotalBytesTX* | • Bytes transmitted without error (NULL if not present) |
| *pTotalBytesRX* | • Bytes received without error (NULL if not present) |

## 8.857.2 Field Documentation

### 8.857.2.1 ULONG* QosEventInfo::pDataBearer

### 8.857.2.2 ULONG* QosEventInfo::pPacketsCountRX

### 8.857.2.3 ULONG* QosEventInfo::pPacketsCountTX

### 8.857.2.4 ULONGLONG* QosEventInfo::pTotalBytesRX

#### 8.857.2.5 ULONGLONG* QosEventInfo::pTotalBytesTX

## 8.858 QosFlowInfo Struct Reference

### Data Fields

- QosFlowInfoState * pQFlowState
- swiQosFlow * pTxQFlowGranted
- swiQosFlow * pRxQFlowGranted
- swiQosFilter * pTxQFilter [MAX_QOS_FILTER_TLV]
- swiQosFilter * pRxQFilter [MAX_QOS_FILTER_TLV]
- BYTE * pBearerID

### 8.858.1 Detailed Description

This structure contains QoS flow info

**Parameters**

| | |
|---|---|
| *pQFlowState* | • QoS flow state information, please check QosFlowInfoState for more information |
| *pTxQFlow-Granted* | • pointer to the Tx Qos flow granted, please check swiQosFlow for more information |
| *pRxQFlow-Granted* | • pointer to the Rx Qos flow granted |
| *pTxQFilter* | • pointer to the Tx Qos filter |
| *pRxQFilter* | • pointer to the Rx Qos flow |
| *pBearerID* | • pointer to 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. |

### 8.858.2 Field Documentation

#### 8.858.2.1 BYTE* QosFlowInfo::pBearerID

#### 8.858.2.2 QosFlowInfoState* QosFlowInfo::pQFlowState

#### 8.858.2.3 swiQosFilter* QosFlowInfo::pRxQFilter[MAX_QOS_FILTER_TLV]

#### 8.858.2.4 swiQosFlow* QosFlowInfo::pRxQFlowGranted

#### 8.858.2.5 swiQosFilter* QosFlowInfo::pTxQFilter[MAX_QOS_FILTER_TLV]

#### 8.858.2.6 swiQosFlow* QosFlowInfo::pTxQFlowGranted

## 8.859 QosFlowInfoState Struct Reference

### Data Fields

- ULONG id
- BYTE isNewFlow
- BYTE state

### 8.859.1 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 |

### 8.859.2 Field Documentation

#### 8.859.2.1 ULONG QosFlowInfoState::id

#### 8.859.2.2 BYTE QosFlowInfoState::isNewFlow

#### 8.859.2.3 BYTE QosFlowInfoState::state

## 8.860 QosMap Struct Reference

### Data Fields

- BYTE dscp
- ULONG qos_id
- BYTE state

### 8.860.1 Detailed Description

This structure contains the SLQSQosDumpMap Information

**Parameters**

| | |
|---|---|
| *dscp* | • Differential Service Code Point(DSCP) value |
| *qos_id* | • QoS identifier |
| *state* | • QoS Flow state |

### 8.860.2 Field Documentation

#### 8.860.2.1 BYTE QosMap::dscp

#### 8.860.2.2 ULONG QosMap::qos_id

#### 8.860.2.3 BYTE QosMap::state

## 8.861 RankIndicatorInd Struct Reference

### Data Fields

- WORD Count1
- WORD Count2

### 8.861.1 Field Documentation

#### 8.861.1.1 WORD RankIndicatorInd::Count1

#### 8.861.1.2 WORD RankIndicatorInd::Count2

## 8.862 readResult Struct Reference

### Data Fields

- WORD contentLen
- BYTE content [255+1]

### 8.862.1 Detailed Description

This structure contains the information for write operation.

**Parameters**

| | |
|---|---|
| *contentLen* | • Number of sets of content. |
| *content[MAX_DESCRIPTION_LENGTH]* | • Read content.<br>• The content is the sequence of bytes as read from the card. |

### 8.862.2 Field Documentation

#### 8.862.2.1 BYTE readResult::content[255+1]

#### 8.862.2.2 WORD readResult::contentLen

## 8.863 readTransparentInfo Struct Reference

### Data Fields

- WORD offset
- WORD length

### 8.863.1 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. |

### 8.863.2 Field Documentation

#### 8.863.2.1 WORD readTransparentInfo::length

#### 8.863.2.2 WORD readTransparentInfo::offset

## 8.864 redirNumInfo Struct Reference

### Data Fields

- BYTE PI
- BYTE SI
- BYTE numType
- BYTE numPlan
- BYTE reason
- BYTE numLen
- BYTE number [255]

### 8.864.1 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. |

### 8.864.2 Field Documentation

#### 8.864.2.1 BYTE redirNumInfo::number[255]

#### 8.864.2.2 BYTE redirNumInfo::numLen

#### 8.864.2.3 BYTE redirNumInfo::numPlan

#### 8.864.2.4 BYTE redirNumInfo::numType

#### 8.864.2.5 BYTE redirNumInfo::PI

#### 8.864.2.6 BYTE redirNumInfo::reason

#### 8.864.2.7 BYTE redirNumInfo::SI

## 8.865 registerRefresh Struct Reference

### Data Fields

- BYTE registerFlag
- BYTE voteForInit
- WORD numFiles
- fileInfo arrfileInfo [255]

### 8.865.1 Detailed Description

This structure contains paramaters 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 /ref fileInfo for more information |

### 8.865.2 Field Documentation

#### 8.865.2.1 fileInfo registerRefresh::arrfileInfo[255]

#### 8.865.2.2 WORD registerRefresh::numFiles

#### 8.865.2.3 BYTE registerRefresh::registerFlag

#### 8.865.2.4 BYTE registerRefresh::voteForInit

## 8.866 remainingRetries Struct Reference

### Data Fields

- BYTE verifyLeft
- BYTE unblockLeft

### 8.866.1 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.

### 8.866.2 Field Documentation

#### 8.866.2.1 BYTE remainingRetries::unblockLeft

#### 8.866.2.2 BYTE remainingRetries::verifyLeft

## 8.867 remotePartyName Struct Reference

### Data Fields

- BYTE namePI
- BYTE codingScheme
- BYTE nameLen
- BYTE callerName [255]

### 8.867.1 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_DESCRIPTION_LENGTH]* | • Name in ASCII, NULL ending. |

### 8.867.2 Field Documentation

#### 8.867.2.1 BYTE remotePartyName::callerName[255]

#### 8.867.2.2 BYTE remotePartyName::codingScheme

#### 8.867.2.3 BYTE remotePartyName::nameLen

#### 8.867.2.4 BYTE remotePartyName::namePI

## 8.868 remotePartyNum Struct Reference

### Data Fields

- BYTE presentationInd
- BYTE numLen
- BYTE remPartyNumber [81]

### 8.868.1 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. |
| *remParty-Number[MAX_-CALL_NO_LEN]* | • Array of numbers in ASCII, NULL ending. |

### 8.868.2 Field Documentation

#### 8.868.2.1 BYTE remotePartyNum::numLen

#### 8.868.2.2 BYTE remotePartyNum::presentationInd

#### 8.868.2.3 BYTE remotePartyNum::remPartyNumber[81]

## 8.869 ReqFieldsList Struct Reference

### Data Fields

- BYTE requestFieldsLen
- BYTE requestFields [256]

### 8.869.1 Detailed Description

This structure contains the Supported Request Fields List Information

**Parameters**

| | |
|---|---|
| *requestFields-Len* | • Number of sets of the request fields. |
| *requestFields* | • Describes which optional field IDs are supported in QMI Request.<br>• Array of uint8 is a bitmask where each bit represents a field ID.<br>• Field 0-15 are mandatory, First Bit represents field ID 16,<br>• Starting with the LSB, bit 0 represents Field ID 16, bit 1 represents ID 17. |

### 8.869.2 Field Documentation

#### 8.869.2.1 BYTE ReqFieldsList::requestFields[256]

#### 8.869.2.2 BYTE ReqFieldsList::requestFieldsLen

## 8.870 RespFieldsList Struct Reference

### Data Fields

- BYTE responseFieldsLen
- BYTE responseFields [256]

### 8.870.1 Detailed Description

This structure contains the Supported Response Fields List Information

**Parameters**

| | |
|---|---|
| *responseFields-Len* | • Number of sets of the response fields. |
| *responseFields* | • Describes which optional field IDs are supported in QMI Response.<br>• Format is same as request field. |

### 8.870.2 Field Documentation

#### 8.870.2.1 BYTE RespFieldsList::responseFields[256]

#### 8.870.2.2 BYTE RespFieldsList::responseFieldsLen

## 8.871 RFBandInfoElements Struct Reference

### Data Fields

- BYTE radioInterface
- WORD activeBandClass
- WORD activeChannel
- uint8_t radioInterface
- uint16_t activeBandClass
- uint16_t activeChannel

### 8.871.1 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) |

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) |

### 8.871.2 Field Documentation

#### 8.871.2.1 WORD RFBandInfoElements::activeBandClass

#### 8.871.2.2 uint16_t RFBandInfoElements::activeBandClass

#### 8.871.2.3 WORD RFBandInfoElements::activeChannel

#### 8.871.2.4 uint16_t RFBandInfoElements::activeChannel

#### 8.871.2.5 BYTE RFBandInfoElements::radioInterface

#### 8.871.2.6 uint8_t RFBandInfoElements::radioInterface

## 8.872 RfBandInfoExtFormat Struct Reference

### Data Fields

- BYTE TlvPresent
- BYTE * pInstancesSize
- struct RfBandInfoExtFormatElements * pRfBandInfoExtFormatParam

### 8.872.1 Detailed Description

This structure contains the 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 RfBandInfoExtFormatElements for more information |

### 8.872.2 Field Documentation

#### 8.872.2.1 BYTE* RfBandInfoExtFormat::pInstancesSize

#### 8.872.2.2 struct RfBandInfoExtFormatElements* RfBandInfoExtFormat::pRfBandInfoExtFormatParam

#### 8.872.2.3 BYTE RfBandInfoExtFormat::TlvPresent

## 8.873 RfBandInfoExtFormatElements Struct Reference

### Data Fields

- BYTE radioInterface
- WORD activeBand
- ULONG activeChannel

### 8.873.1 Detailed Description

This structure contains the RfBandInfoExtFormatElements response parameters.

**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) |

### 8.873.2 Field Documentation

#### 8.873.2.1 WORD RfBandInfoExtFormatElements::activeBand

#### 8.873.2.2 ULONG RfBandInfoExtFormatElements::activeChannel

#### 8.873.2.3 BYTE RfBandInfoExtFormatElements::radioInterface

## 8.874 RfBandInfoList Struct Reference

### Data Fields

- BYTE * pInstanceSize
- struct RFBandInfoElements * pRFBandInfoParam

### 8.874.1 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 RFBandInfoElements for more information |

### 8.874.2 Field Documentation

#### 8.874.2.1 BYTE* RfBandInfoList::pInstanceSize

#### 8.874.2.2 struct RFBandInfoElements* RfBandInfoList::pRFBandInfoParam

## 8.875 RfBandwidthInfo Struct Reference

### Data Fields

- BYTE TlvPresent
- BYTE * pInstancesSize
- struct RfBandwidthInfoElements * pRfBandwidthInfoParam

### 8.875.1 Detailed Description

This structure contains the 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. |
| *pRfBandwidthInfoParam* | • See RfBandwidthInfoElements for more information |

### 8.875.2 Field Documentation

#### 8.875.2.1 BYTE* RfBandwidthInfo::pInstancesSize

#### 8.875.2.2 struct RfBandwidthInfoElements* RfBandwidthInfo::pRfBandwidthInfoParam

#### 8.875.2.3 BYTE RfBandwidthInfo::TlvPresent

## 8.876 RfBandwidthInfoElements Struct Reference

### Data Fields

- BYTE radioInterface
- ULONG bandwidth

### 8.876.1 Detailed Description

This structure contains the RfBandwidthInfoElements response 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 |

### 8.876.2 Field Documentation

#### 8.876.2.1 ULONG RfBandwidthInfoElements::bandwidth

#### 8.876.2.2 BYTE RfBandwidthInfoElements::radioInterface

## 8.877 RfDedicatedBandInfo Struct Reference

### Data Fields

- BYTE TlvPresent
- BYTE * pInstancesSize
- struct RfDedicatedBandInfoElements * pRfDedicatedBandInfoParam

### 8.877.1 Detailed Description

This structure contains the 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 RfDedicatedBandInfoElements for more information |

### 8.877.2 Field Documentation

#### 8.877.2.1 BYTE* RfDedicatedBandInfo::pInstancesSize

#### 8.877.2.2 struct RfDedicatedBandInfoElements* RfDedicatedBandInfo::pRfDedicatedBandInfoParam

#### 8.877.2.3 BYTE RfDedicatedBandInfo::TlvPresent

## 8.878 RfDedicatedBandInfoElements Struct Reference

### Data Fields

- BYTE radioInterface
- WORD dedicatedBand

### 8.878.1 Detailed Description

This structure contains the RfDedicatedBandInfoElements response parameters.

**Parameters**

| | |
|---|---|
| *radioInterface* | • Radio interface technology<br>– See Tables for Radio Interface |
| *dedicatedBand* | • Active Band Class<br>– See Tables for Band Classes |

### 8.878.2 Field Documentation

#### 8.878.2.1 WORD RfDedicatedBandInfoElements::dedicatedBand

#### 8.878.2.2 BYTE RfDedicatedBandInfoElements::radioInterface

## 8.879 rmTrasnferStaticsReq Struct Reference

### Data Fields

- uint8_t bResetStatistics
- uint32_t ulMask

### 8.879.1 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 |

### 8.879.2 Field Documentation

#### 8.879.2.1 uint8_t rmTrasnferStaticsReq::bResetStatistics

#### 8.879.2.2 uint32_t rmTrasnferStaticsReq::ulMask

## 8.880 roamIndList Struct Reference

### Data Fields

- BYTE numInstances
- BYTE radioInterface [0x0A]
- BYTE roamIndicator [0x0A]

### 8.880.1 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 |

### 8.880.2 Field Documentation

#### 8.880.2.1 BYTE roamIndList::numInstances

#### 8.880.2.2 BYTE roamIndList::radioInterface[0x0A]

#### 8.880.2.3 BYTE roamIndList::roamIndicator[0x0A]

## 8.881 RoamingInfo Struct Reference

### Data Fields

- BYTE TlvPresent
- BYTE roaming_ind

### 8.881.1 Field Documentation

#### 8.881.1.1 BYTE RoamingInfo::roaming_ind

#### 8.881.1.2 BYTE RoamingInfo::TlvPresent

## 8.882 roamTimer Struct Reference

### Data Fields

- BYTE namID
- ULONG roamTimerValue

#### 8.882.1 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. |

#### 8.882.2 Field Documentation

##### 8.882.2.1 BYTE roamTimer::namID

##### 8.882.2.2 ULONG roamTimer::roamTimerValue

## 8.883 RSRPThresh Struct Reference

### Data Fields

- BYTE RSRPThresListLen
- SHORT * pRSRPThresList

#### 8.883.1 Detailed Description

This structure contains RSRP threshold related parameters.

**Parameters**

| | |
|---|---|
| *RSRPThresList-Len* | • Length of the LTE 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 |

### 8.883.2 Field Documentation

#### 8.883.2.1 SHORT* RSRPThresh::pRSRPThresList

#### 8.883.2.2 BYTE RSRPThresh::RSRPThresListLen

## 8.884 rsrqInformation Struct Reference

### Data Fields

- INT8 rsrq
- BYTE radioIf

### 8.884.1 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 |

### 8.884.2 Field Documentation

#### 8.884.2.1 BYTE rsrqInformation::radioIf

#### 8.884.2.2 INT8 rsrqInformation::rsrq

## 8.885 RSRQThresh Struct Reference

### Data Fields

- BYTE RSRQThresListLen
- INT8 * pRSRQThresList

### 8.885.1 Detailed Description

This structure contains RSRQ threshold related parameters.

**Parameters**

| | |
|---|---|
| *RSRQThresList-Len* | • Length of the LTE 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 |

### 8.885.2 Field Documentation

#### 8.885.2.1 INT8* RSRQThresh::pRSRQThresList

#### 8.885.2.2 BYTE RSRQThresh::RSRQThresListLen

## 8.886 RSSIThresh Struct Reference

### Data Fields

- BYTE RSSIThresListLen
- INT8 * pRSSIThresList

### 8.886.1 Detailed Description

This structure contains RSSI threshold related parameters.

**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. |

### 8.886.2 Field Documentation

#### 8.886.2.1 INT8* RSSIThresh::pRSSIThresList

#### 8.886.2.2 BYTE RSSIThresh::RSSIThresListLen

## 8.887 RXAGCList Struct Reference

### Data Fields

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

### 8.887.1 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 |

### 8.887.2 Field Documentation

#### 8.887.2.1 WORD* RXAGCList::pRXAIG

#### 8.887.2.2 WORD* RXAGCList::pRXComprSlope

#### 8.887.2.3 WORD* RXAGCList::pRXComprThres

#### 8.887.2.4 WORD* RXAGCList::pRXExpSlope

#### 8.887.2.5 WORD* RXAGCList::pRXExpThres

#### 8.887.2.6 WORD* RXAGCList::pRXStaticGain

## 8.888 RXAVCList Struct Reference

### Data Fields

- WORD * pAVRXAVCSens
- WORD * pAVRXAVCHeadroom

### 8.888.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXAVCLIST.

**Parameters**

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

### 8.888.2 Field Documentation

#### 8.888.2.1 WORD* RXAVCList::pAVRXAVCHeadroom

#### 8.888.2.2 WORD* RXAVCList::pAVRXAVCSens

## 8.889 rxInfo Struct Reference

### Data Fields

- BYTE isRadioTuned

- INT32 rxPower
- INT32 ecio
- INT32 rscp
- INT32 rsrp
- ULONG phase

### 8.889.1 Detailed Description

This structure contains the Rx Information.

**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. |

### 8.889.2 Field Documentation

#### 8.889.2.1 INT32 rxInfo::ecio

#### 8.889.2.2 BYTE rxInfo::isRadioTuned

#### 8.889.2.3 ULONG rxInfo::phase

#### 8.889.2.4 INT32 rxInfo::rscp

#### 8.889.2.5 INT32 rxInfo::rsrp

#### 8.889.2.6 INT32 rxInfo::rxPower

## 8.890 RXPCMIIRFltr Struct Reference

### Data Fields

- WORD * pFlag
- WORD * pStageCnt
- BYTE * pStage0Val
- BYTE * pStage1Val
- BYTE * pStage2Val
- BYTE * pStage3Val
- BYTE * pStage4Val

### 8.890.1 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 |

### 8.890.2 Field Documentation

#### 8.890.2.1 WORD* RXPCMIIRFltr::pFlag

#### 8.890.2.2 BYTE* RXPCMIIRFltr::pStage0Val

#### 8.890.2.3 BYTE* RXPCMIIRFltr::pStage1Val

#### 8.890.2.4 BYTE* RXPCMIIRFltr::pStage2Val

#### 8.890.2.5 BYTE* RXPCMIIRFltr::pStage3Val

#### 8.890.2.6 BYTE* RXPCMIIRFltr::pStage4Val

#### 8.890.2.7 WORD* RXPCMIIRFltr::pStageCnt

## 8.891 RxSigInfo Struct Reference

### Data Fields

- BYTE rxChainIndex
- BYTE isRadioTuned
- INT32 rxPower
- INT32 rsrp

### 8.891.1 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 |

### 8.891.2 Field Documentation

#### 8.891.2.1 BYTE RxSigInfo::isRadioTuned

#### 8.891.2.2 INT32 RxSigInfo::rsrp

#### 8.891.2.3 BYTE RxSigInfo::rxChainIndex

#### 8.891.2.4 INT32 RxSigInfo::rxPower

## 8.892 rxSignalStrengthListElement Struct Reference

### Data Fields

- SHORT rxSignalStrength
- BYTE radioIf

### 8.892.1 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 - cdma2000Â® 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 element of the RSSI list always contains the current Signal strength and Radio Interface.

### 8.892.2 Field Documentation

#### 8.892.2.1 BYTE rxSignalStrengthListElement::radioIf

#### 8.892.2.2 SHORT rxSignalStrengthListElement::rxSignalStrength

## 8.893 sApnExtraParams Struct Reference

### Data Fields

- ULONG apnId
- BYTE ambr_ul
- BYTE ambr_dl
- BYTE ambr_ul_ext
- BYTE ambr_dl_ext
- BYTE ambr_ul_ext2
- BYTE ambr_dl_ext2

### 8.893.1 Detailed Description

This structure contains the Extra Apn Params

**Parameters**

| | |
|---|---|
| *apnId* | • APN id<br>• ID identifying the APN that the client would like to query the AMBR params |
| *ambr_ul* | • APN AMBR uplink<br>• APN AMBR uplink values from 1 kbps to 8640 kbps |

| | |
|---|---|
| *ambr_dl* | • APN AMBR downlink<br>• APN AMBR downlink values from 1 kbps to 8640 kbps |
| *ambr_ul_ext* | • Extended APN AMBR uplink<br>• APN AMBR uplink values from 8700 kbps to 256 Mbps |
| *ambr_dl_ext* | • Extended APN AMBR downlink<br>• APN AMBR downlink values from 8700 kbps to 256 Mbps |
| *ambr_ul_ext2* | • Second extended APN AMBR uplink<br>• APN AMBR uplink values from 256 Mbps to 65280 Mbps |
| *ambr_dl_ext2* | • Second extended APN AMBR downlink<br>• APN AMBR downlink values from 256 Mbps to 65280 Mbps |

### 8.893.2 Field Documentation

#### 8.893.2.1 BYTE sApnExtraParams::ambr_dl

#### 8.893.2.2 BYTE sApnExtraParams::ambr_dl_ext

#### 8.893.2.3 BYTE sApnExtraParams::ambr_dl_ext2

#### 8.893.2.4 BYTE sApnExtraParams::ambr_ul

#### 8.893.2.5 BYTE sApnExtraParams::ambr_ul_ext

#### 8.893.2.6 BYTE sApnExtraParams::ambr_ul_ext2

#### 8.893.2.7 ULONG sApnExtraParams::apnId

## 8.894 satelliteInfo Struct Reference

### Data Fields

- BYTE svListLen
- ULONG validMask
- ULONG system
- WORD gnssSvId
- BYTE healthStatus
- ULONG svStatus
- BYTE svInfoMask
- FLOAT elevation
- FLOAT azimuth
- FLOAT snr

### 8.894.1 Detailed Description

Contain fields in struct 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 |

**Note**

None

### 8.894.2 Field Documentation

#### 8.894.2.1 FLOAT satelliteInfo::azimuth

#### 8.894.2.2 FLOAT satelliteInfo::elevation

#### 8.894.2.3 WORD satelliteInfo::gnssSvId

#### 8.894.2.4 BYTE satelliteInfo::healthStatus

#### 8.894.2.5 FLOAT satelliteInfo::snr

#### 8.894.2.6 BYTE satelliteInfo::svInfoMask

#### 8.894.2.7 BYTE satelliteInfo::svListLen

#### 8.894.2.8 ULONG satelliteInfo::svStatus

#### 8.894.2.9 ULONG satelliteInfo::system

#### 8.894.2.10 ULONG satelliteInfo::validMask

## 8.895 SccRxInfo Struct Reference

### Data Fields

- INT32 rsrq
- SHORT snr
- BYTE numInstances
- RxSigInfo sigInfo [255]
- BYTE TlvPresent

### 8.895.1 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 RxSigInfo for more information |
| *TlvPresent* | • Tlv Present. |

### 8.895.2 Field Documentation

#### 8.895.2.1 BYTE SccRxInfo::numInstances

#### 8.895.2.2 INT32 SccRxInfo::rsrq

#### 8.895.2.3 RxSigInfo SccRxInfo::sigInfo[255]

#### 8.895.2.4 SHORT SccRxInfo::snr

#### 8.895.2.5 BYTE SccRxInfo::TlvPresent

## 8.896 sensorData Struct Reference

### Data Fields

- ULONG timeOfFirstSample
- BYTE flags
- BYTE sensorDataLen
- WORD timeOffset [64]
- ULONG xAxis [64]
- ULONG yAxis [64]
- ULONG zAxis [64]

### 8.896.1 Detailed Description

This structure specifies information regarding the 3-Axis Sensor Data.

**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 |

### 8.896.2 Field Documentation

#### 8.896.2.1 BYTE sensorData::flags

#### 8.896.2.2 BYTE sensorData::sensorDataLen

#### 8.896.2.3 ULONG sensorData::timeOfFirstSample

#### 8.896.2.4 WORD sensorData::timeOffset[64]

#### 8.896.2.5 ULONG sensorData::xAxis[64]

#### 8.896.2.6 ULONG sensorData::yAxis[64]

#### 8.896.2.7 ULONG sensorData::zAxis[64]

## 8.897 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]

### 8.897.1 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 |

### 8.897.2 Field Documentation

#### 8.897.2.1 uint8_t sensorData_t::flags

#### 8.897.2.2 uint8_t sensorData_t::sensorDataLen

#### 8.897.2.3 uint32_t sensorData_t::timeOfFirstSample

#### 8.897.2.4 uint16_t sensorData_t::timeOffset[64]

#### 8.897.2.5 uint32_t sensorData_t::xAxis[64]

#### 8.897.2.6 uint32_t sensorData_t::yAxis[64]

#### 8.897.2.7 uint32_t sensorData_t::zAxis[64]

## 8.898 sensorDataUsage_s Struct Reference

### Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

### 8.898.1 Detailed Description

This structure contains Sensor Data Usage info.

**Parameters**

| | |
|---|---|
| *usageMask* | • Specifies which sensors were used in calculating the position in the position report. |

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

**Parameters**

| | |
|---|---|
| *aidingIndicator-Mask* | |

  - Specifies which results were aided by sensors.

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

### 8.898.2 Field Documentation

#### 8.898.2.1 ULONG sensorDataUsage_s::aidingIndicatorMask

#### 8.898.2.2 ULONG sensorDataUsage_s::usageMask

## 8.899 serialNumbersInfo Struct Reference

### Data Fields

- BYTE esnSize
- CHAR * pESNString
- BYTE imeiSize
- CHAR * pIMEIString
- BYTE meidSize
- CHAR * pMEIDString
- BYTE imeiSvnSize
- CHAR * pImeiSvnString

### 8.899.1 Detailed Description

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity) and MEID (Mobile Equipment Identifier).

**Parameters**

| | |
|---|---|
| *esnSize* | • The maximum number of characters (including NULL terminator) that the ESN string array can contain |
| *pESNString[OUT]* | • NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed |
| *imeiSize* | • The maximum number of characters (including NULL terminator) that the IMEI string array can contain |
| *pIMEIString[OUT]* | • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed |
| *meidSize* | • The maximum number of characters (including NULL terminator) that the MEID string array can contain |
| *pMEIDString[OUT]* | • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed |
| *imeiSvnSize* | • The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain. |
| *pImeiSvnString[OUT]* | • NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed. |

### 8.899.2 Field Documentation

#### 8.899.2.1 BYTE serialNumbersInfo::esnSize

#### 8.899.2.2 BYTE serialNumbersInfo::imeiSize

#### 8.899.2.3 BYTE serialNumbersInfo::imeiSvnSize

#### 8.899.2.4 BYTE serialNumbersInfo::meidSize

#### 8.899.2.5 CHAR* serialNumbersInfo::pESNString

#### 8.899.2.6 CHAR* serialNumbersInfo::pIMEIString

#### 8.899.2.7 CHAR* serialNumbersInfo::pImeiSvnString

#### 8.899.2.8 CHAR* serialNumbersInfo::pMEIDString

## 8.900 serviceProviderName Struct Reference

### Data Fields

- BYTE displayCondition
- BYTE spnLength
- BYTE spn [255]

### 8.900.1 Detailed Description

This structure contains Service Provider Name as defined in 3GPP TS 31.102 (Section 4.2.12) from multiple sources.

**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. |

### 8.900.2 Field Documentation

#### 8.900.2.1 BYTE serviceProviderName::displayCondition

#### 8.900.2.2 BYTE serviceProviderName::spn[255]

#### 8.900.2.3 BYTE serviceProviderName::spnLength

## 8.901 ServingSystemInfo Struct Reference

### Data Fields

- BYTE registrationState
- BYTE csAttachState
- BYTE psAttachState
- BYTE selectedNetwork
- BYTE radioInterfaceNo
- BYTE radioInterfaceList [255]
- BYTE hdrPersonality

### 8.901.1 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

### 8.901.2 Field Documentation

#### 8.901.2.1 BYTE ServingSystemInfo::csAttachState

#### 8.901.2.2 BYTE ServingSystemInfo::hdrPersonality

#### 8.901.2.3 BYTE ServingSystemInfo::psAttachState

#### 8.901.2.4 BYTE ServingSystemInfo::radioInterfaceList[255]

#### 8.901.2.5 BYTE ServingSystemInfo::radioInterfaceNo

#### 8.901.2.6 BYTE ServingSystemInfo::registrationState

#### 8.901.2.7 BYTE ServingSystemInfo::selectedNetwork

## 8.902 servSystem Struct Reference

### Data Fields

- BYTE regState
- BYTE csAttachState
- BYTE psAttachState
- BYTE selNetwork
- BYTE numRadioInterfaces
- BYTE radioInterface [0x0A]

### 8.902.1 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(s) modem discovered<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 |

### 8.902.2 Field Documentation

#### 8.902.2.1 BYTE servSystem::csAttachState

#### 8.902.2.2 BYTE servSystem::numRadioInterfaces

#### 8.902.2.3 BYTE servSystem::psAttachState

#### 8.902.2.4 BYTE servSystem::radioInterface[0x0A]

#### 8.902.2.5 BYTE servSystem::regState

#### 8.902.2.6 BYTE servSystem::selNetwork

## 8.903 sessionInfo Union Reference

### Data Fields

- struct omaDmFotaTlv omaDmFota
- struct omaDmConfigTlv omaDmConfig
- struct omaDmNotificationsTlv omaDmNotifications

### 8.903.1 Detailed Description

This union sessionInfo consist of omaDmFotaTlv, omaDmConfigTlv and omaDmNotificationsTlv, out of which one will be unpacked against pEventFields.

### 8.903.2 Field Documentation

#### 8.903.2.1 struct omaDmConfigTlv sessionInfo::omaDmConfig

#### 8.903.2.2 struct omaDmFotaTlv sessionInfo::omaDmFota

#### 8.903.2.3 struct omaDmNotificationsTlv sessionInfo::omaDmNotifications

## 8.904 sessionInfoExt Union Reference

### Data Fields

- struct omaDmFotaTlvExt omaDmFota
- struct omaDmConfigTlvExt omaDmConfig

### 8.904.1 Detailed Description

This union sessionInfo consist of omaDmFotaTlv and omaDmConfigTlv, out of which one will be unpacked against pEventFields.

### 8.904.2 Field Documentation

#### 8.904.2.1 struct omaDmConfigTlvExt sessionInfoExt::omaDmConfig

#### 8.904.2.2 struct omaDmFotaTlvExt sessionInfoExt::omaDmFota

## 8.905 sessionInfoTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformation sessionInfo

### 8.905.1 Detailed Description

Structure used to store all sessionInfo(Union)TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *sessionType* | - sessiontype Value |

### 8.905.2 Field Documentation

#### 8.905.2.1 sessionInformation sessionInfoTlv::sessionInfo

#### 8.905.2.2 ULONG sessionInfoTlv::sessionType

#### 8.905.2.3 BYTE sessionInfoTlv::TlvPresent

## 8.906 sessionInfoTlvExt Struct Reference

### Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformationExt sessionInfo

### 8.906.1 Detailed Description

Structure used to store all sessionInfo(Union)TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *sessionType* | - sessiontype Value |

### 8.906.2 Field Documentation

#### 8.906.2.1 sessionInformationExt sessionInfoTlvExt::sessionInfo

#### 8.906.2.2 ULONG sessionInfoTlvExt::sessionType

#### 8.906.2.3 BYTE sessionInfoTlvExt::TlvPresent

## 8.907 SetAudioPathConfigReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE * pECMode
- BYTE * pNSEnable
- WORD * pTXGain
- WORD * pDTMFTXGain
- WORD * pCodecSTGain
- TXPCMIIRFltr * pTXPCMIIRFltr
- RXPCMIIRFltr * pRXPCMIIRFltr
- BYTE * pRXAVCAGCSwitch
- BYTE * pTXAVCSwitch
- RXAGCList * pRXAGCList
- RXAVCList * pRXAVCList
- TXAGCList * pTXAGCList

### 8.907.1 Detailed Description

This structure contains the SLQSSetAudioPathConfig request parameters.

**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 TXPCMIIRFltr for more information |
| *pRXPCMIIRFltr* | [Optional]<br>• See RXPCMIIRFltr for more information |
| *pRXAVCAGC-Switch* | [Optional]<br>• RX AVC/AGC Switch |
| *pTXAVCSwitch* | [Optional]<br>• TX AVC Switch |
| *pRXAGCList* | [Optional]<br>• See RXAGCList for more information |
| *pRXAVCList* | [Optional]<br>• See RXAVCList for more information |
| *pTXAGCList* | [Optional]<br>• See TXAGCList for more information |

## 8.907.2 Field Documentation

### 8.907.2.1 WORD* SetAudioPathConfigReq::pCodecSTGain

#### 8.907.2.2 WORD* SetAudioPathConfigReq::pDTMFTXGain

#### 8.907.2.3 BYTE* SetAudioPathConfigReq::pECMode

#### 8.907.2.4 BYTE* SetAudioPathConfigReq::pNSEnable

#### 8.907.2.5 BYTE SetAudioPathConfigReq::Profile

#### 8.907.2.6 RXAGCList* SetAudioPathConfigReq::pRXAGCList

#### 8.907.2.7 BYTE* SetAudioPathConfigReq::pRXAVCAGCSwitch

#### 8.907.2.8 RXAVCList* SetAudioPathConfigReq::pRXAVCList

#### 8.907.2.9 RXPCMIIRFltr* SetAudioPathConfigReq::pRXPCMIIRFltr

#### 8.907.2.10 TXAGCList* SetAudioPathConfigReq::pTXAGCList

#### 8.907.2.11 BYTE* SetAudioPathConfigReq::pTXAVCSwitch

#### 8.907.2.12 WORD* SetAudioPathConfigReq::pTXGain

#### 8.907.2.13 TXPCMIIRFltr* SetAudioPathConfigReq::pTXPCMIIRFltr

## 8.908 SetAudioProfileReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE EarMute
- BYTE MicMute
- BYTE Generator
- BYTE Volume

### 8.908.1 Detailed Description

This structure contains the SLQSSetAudioProfile request parameters.

**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 |

### 8.908.2 Field Documentation

#### 8.908.2.1 BYTE SetAudioProfileReq::EarMute

#### 8.908.2.2 BYTE SetAudioProfileReq::Generator

#### 8.908.2.3 BYTE SetAudioProfileReq::MicMute

#### 8.908.2.4 BYTE SetAudioProfileReq::Profile

#### 8.908.2.5 BYTE SetAudioProfileReq::Volume

## 8.909 SetAudioVolTLBConfigReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Volume
- BYTE Item
- WORD VolValue

### 8.909.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig request parameters

**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 |

### 8.909.2 Field Documentation

#### 8.909.2.1 BYTE SetAudioVolTLBConfigReq::Generator

#### 8.909.2.2 BYTE SetAudioVolTLBConfigReq::Item

#### 8.909.2.3 BYTE SetAudioVolTLBConfigReq::Profile

#### 8.909.2.4 BYTE SetAudioVolTLBConfigReq::Volume

#### 8.909.2.5 WORD SetAudioVolTLBConfigReq::VolValue

## 8.910 SetAudioVolTLBConfigResp Struct Reference

### Data Fields

- WORD ResCode

### 8.910.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig response parameters.

**Parameters**

| | |
|---|---|
| *ResCode* | • Result of requested item |

### 8.910.2 Field Documentation

#### 8.910.2.1 WORD SetAudioVolTLBConfigResp::ResCode

## 8.911 setCustomSettingV2 Struct Reference

### Data Fields

- CHAR cust_id [64+1]
- WORD value_length
- BYTE cust_value [8+1]

### 8.911.1 Detailed Description

This structure contains customization settings set to modem

**Parameters**

| | |
|---|---|
| *cust_id[IN]* | • Customization ID (Maximum 64 bytes) |
| *value_length[IN]* | • length of cust_value field |
| *cust_value[IN]* | • Customization Setting Value (Maximum 8 bytes) |

### 8.911.2 Field Documentation

#### 8.911.2.1 CHAR setCustomSettingV2::cust_id[64+1]

#### 8.911.2.2 BYTE setCustomSettingV2::cust_value[8+1]

#### 8.911.2.3 WORD setCustomSettingV2::value_length

## 8.912 setDyingGaspCfg Struct Reference

### Data Fields

- BYTE * pDestSMSNum
- BYTE * pDestSMSContent

### 8.912.1 Detailed Description

This struture contains the TLV required to get the Dysing GASP Config.

**Parameters**

| | |
|---|---|
| *pDestSMSNum[OUT]* | • SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.<br>• Optional parameter. |
| *pDestSMSContent[OUT]* | • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars.<br>• Optional parameter. |

### 8.912.2 Field Documentation

#### 8.912.2.1 BYTE* setDyingGaspCfg::pDestSMSContent

#### 8.912.2.2 BYTE* setDyingGaspCfg::pDestSMSNum

## 8.913 SetIMSSMSConfigReq Struct Reference

### Data Fields

- BYTE * pSMSFormat
- BYTE * pSMSOverIPNwInd
- BYTE * pPhoneCtxtURILen
- BYTE * pPhoneCtxtURI

### 8.913.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig request 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 |

### 8.913.2 Field Documentation

#### 8.913.2.1 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURI

#### 8.913.2.2 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURILen

#### 8.913.2.3 BYTE* SetIMSSMSConfigReq::pSMSFormat

#### 8.913.2.4 BYTE* SetIMSSMSConfigReq::pSMSOverIPNwInd

## 8.914 SetIMSSMSConfigResp Struct Reference

### Data Fields

- BYTE * pSettingResp

### 8.914.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig response 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 |

### 8.914.2 Field Documentation

#### 8.914.2.1 BYTE* SetIMSSMSConfigResp::pSettingResp

## 8.915 SetIMSUserConfigReq Struct Reference

### Data Fields

- BYTE * pIMSDomainLen
- BYTE * pIMSDomain

### 8.915.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig request parameters.

**Parameters**

| | |
|---|---|
| *pIMSDomainLen* | • Length of IMS Domain Name to follow |
| *pIMSDomain* | • IMS domain name |

### 8.915.2 Field Documentation

#### 8.915.2.1 BYTE* SetIMSUserConfigReq::pIMSDomain

#### 8.915.2.2 BYTE* SetIMSUserConfigReq::pIMSDomainLen

## 8.916 SetIMSUserConfigResp Struct Reference

### Data Fields

- BYTE * pSettingResp

### 8.916.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig response 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 |

### 8.916.2 Field Documentation

#### 8.916.2.1 BYTE* SetIMSUserConfigResp::pSettingResp

## 8.917 SetIMSVoIPConfigReq Struct Reference

### Data Fields

- WORD * pSessionExpiryTimer
- WORD * pMinSessionExpiryTimer
- BYTE * pAmrWbEnable
- BYTE * pScrAmrEnable
- BYTE * pScrAmrWbEnable
- BYTE * pAmrMode
- WORD * pAmrWBMode
- BYTE * pAmrOctetAligned
- BYTE * pAmrWBOctetAligned
- WORD * pRingingTimer
- WORD * pRingBackTimer
- WORD * pRTPRTCPInactTimer

### 8.917.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig request 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. |

### 8.917.2 Field Documentation

#### 8.917.2.1 BYTE* SetIMSVoIPConfigReq::pAmrMode

#### 8.917.2.2 BYTE* SetIMSVoIPConfigReq::pAmrOctetAligned

#### 8.917.2.3 BYTE* SetIMSVoIPConfigReq::pAmrWbEnable

#### 8.917.2.4 WORD* SetIMSVoIPConfigReq::pAmrWBMode

#### 8.917.2.5 BYTE* SetIMSVoIPConfigReq::pAmrWBOctetAligned

#### 8.917.2.6 WORD* SetIMSVoIPConfigReq::pMinSessionExpiryTimer

#### 8.917.2.7 WORD* SetIMSVoIPConfigReq::pRingBackTimer

#### 8.917.2.8 WORD* SetIMSVoIPConfigReq::pRingingTimer

#### 8.917.2.9 WORD* SetIMSVoIPConfigReq::pRTPRTCPInactTimer

#### 8.917.2.10 BYTE* SetIMSVoIPConfigReq::pScrAmrEnable

#### 8.917.2.11 BYTE* SetIMSVoIPConfigReq::pScrAmrWbEnable

#### 8.917.2.12 WORD* SetIMSVoIPConfigReq::pSessionExpiryTimer

## 8.918 SetIMSVoIPConfigResp Struct Reference

### Data Fields

- BYTE * pSettingResp

### 8.918.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig response 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 |

### 8.918.2 Field Documentation

#### 8.918.2.1 BYTE* SetIMSVoIPConfigResp::pSettingResp

## 8.919 SetM2MAudioAVCFGReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE Device
- BYTE PIFACEId
- PCMparams * pPCMParams

### 8.919.1 Detailed Description

This structure contains the SLQSSetM2MAudioAVCFG request 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 PCMparams for more information |

### 8.919.2 Field Documentation

#### 8.919.2.1 BYTE SetM2MAudioAVCFGReq::Device

#### 8.919.2.2 BYTE SetM2MAudioAVCFGReq::PIFACEId

#### 8.919.2.3 PCMparams* SetM2MAudioAVCFGReq::pPCMParams

#### 8.919.2.4 BYTE SetM2MAudioAVCFGReq::Profile

## 8.920 SetM2MAudioLPBKReq Struct Reference

### Data Fields

- BYTE Enable

### 8.920.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK request parameters.

**Parameters**

| | |
|---|---|
| *Enable* | • Operation to be performed<br>– 0 - stop<br>– 1 - VOCODER loop<br>– 2 - internal codec loop |

### 8.920.2 Field Documentation

#### 8.920.2.1 BYTE SetM2MAudioLPBKReq::Enable

## 8.921 SetM2MAudioProfileReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE * pEarMute
- BYTE * pMicMute
- BYTE * pGenerator
- BYTE * pVolume
- BYTE * pCwtMute

### 8.921.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile request 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 |

### 8.921.2 Field Documentation

#### 8.921.2.1 BYTE* SetM2MAudioProfileReq::pCwtMute

#### 8.921.2.2 BYTE* SetM2MAudioProfileReq::pEarMute

#### 8.921.2.3 BYTE* SetM2MAudioProfileReq::pGenerator

#### 8.921.2.4 BYTE* SetM2MAudioProfileReq::pMicMute

#### 8.921.2.5 BYTE SetM2MAudioProfileReq::Profile

#### 8.921.2.6 BYTE* SetM2MAudioProfileReq::pVolume

## 8.922 SetM2MAudioVolumeReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Level

### 8.922.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile request parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |
| *Generator* | • Generator<br>– 0 - voice |

| | |
|---|---|
| *Level* | • Audio volume level<br>– 0-5 |

### 8.922.2 Field Documentation

#### 8.922.2.1 BYTE SetM2MAudioVolumeReq::Generator

#### 8.922.2.2 BYTE SetM2MAudioVolumeReq::Level

#### 8.922.2.3 BYTE SetM2MAudioVolumeReq::Profile

## 8.923 SetM2MAVMuteReq Struct Reference

### Data Fields

- BYTE Profile
- BYTE EarMute
- BYTE MicMute
- BYTE * pCwtMute

### 8.923.1 Detailed Description

This structure contains the SLQSSetM2MAVMute request 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 |

### 8.923.2 Field Documentation

#### 8.923.2.1 BYTE SetM2MAVMuteReq::EarMute

#### 8.923.2.2 BYTE SetM2MAVMuteReq::MicMute

#### 8.923.2.3 BYTE* SetM2MAVMuteReq::pCwtMute

#### 8.923.2.4 BYTE SetM2MAVMuteReq::Profile

## 8.924 SetM2MSpkrGainReq Struct Reference

### Data Fields

- BYTE Profile
- WORD Value

### 8.924.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain request parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |
| *Value* | • RX speakerphone gain<br>– 0x0 - 0x7fff |

### 8.924.2 Field Documentation

#### 8.924.2.1 BYTE SetM2MSpkrGainReq::Profile

#### 8.924.2.2 WORD SetM2MSpkrGainReq::Value

## 8.925 setPINProtection Struct Reference

### Data Fields

- BYTE pinID
- BYTE pinOperation
- BYTE pinLength
- BYTE pinValue [255]

### 8.925.1 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. |

### 8.925.2 Field Documentation

#### 8.925.2.1 BYTE setPINProtection::pinID

#### 8.925.2.2 BYTE setPINProtection::pinLength

#### 8.925.2.3 BYTE setPINProtection::pinOperation

#### 8.925.2.4 BYTE setPINProtection::pinValue[255]

## 8.926 SetRegMgrConfigReq Struct Reference

### Data Fields

- WORD * pPriCSCFPort
- BYTE * pCSCFPortNameLen
- BYTE * pCSCFPortName
- BYTE * pIMSTestMode

### 8.926.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig request 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 |

### 8.926.2 Field Documentation

#### 8.926.2.1 BYTE* SetRegMgrConfigReq::pCSCFPortName

#### 8.926.2.2 BYTE* SetRegMgrConfigReq::pCSCFPortNameLen

#### 8.926.2.3 BYTE* SetRegMgrConfigReq::pIMSTestMode

#### 8.926.2.4 WORD* SetRegMgrConfigReq::pPriCSCFPort

## 8.927 SetRegMgrConfigResp Struct Reference

### Data Fields

- BYTE * pSettingResp

### 8.927.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig response 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 |

### 8.927.2 Field Documentation

#### 8.927.2.1 BYTE* SetRegMgrConfigResp::pSettingResp

## 8.928 setSignalStrengthInfo Struct Reference

### Data Fields

- CDMARSSIThresh * pCDMARSSIThresh
- WORD * pCDMARSSIDelta
- CDMAECIOThresh * pCDMAECIOThresh
- WORD * pCDMAECIODelta
- HDRRSSIThresh * pHDRRSSIThresh
- WORD * pHDRRSSIDelta

- HDRECIOThresh * pHDRECIOThresh
- WORD * pHDRECIODelta
- HDRSINRThreshold * pHDRSINRThresh
- WORD * pHDRSINRDelta
- HDRIOThresh * pHDRIOThresh
- WORD * pHDRIODelta
- GSMRSSIThresh * pGSMRSSIThresh
- WORD * pGSMRSSIDelta
- WCDMARSSIThresh * pWCDMARSSIThresh
- WORD * pWCDMARSSIDelta
- WCDMAECIOThresh * pWCDMAECIOThresh
- WORD * pWCDMAECIODelta
- LTERSSIThresh * pLTERSSIThresh
- WORD * pLTERSSIDelta
- LTESNRThreshold * pLTESNRThresh
- WORD * pLTESNRDelta
- LTERSRQThresh * pLTERSRQThresh
- WORD * pLTERSRQDelta
- LTERSRPThresh * pLTERSRPThresh
- WORD * pLTERSRPDelta
- LTESigRptConfig * pLTESigRptConfig
- TDSCDMARSCPThresh * pTDSCDMARSCPThresh
- WORD * pTDSCDMARSCPDelta
- TDSCDMARSSIThresh * pTDSCDMARSSIThresh
- ULONG * pTDSCDMARSSIDelta
- TDSCDMAECIOThresh * pTDSCDMAECIOThresh
- ULONG * pTDSCDMAECIODelta
- TDSCDMASINRThresh * pTDSCDMASINRThresh
- ULONG * pTDSCDMASINRDelta

### 8.928.1 Detailed Description

This structure contains the Signal Strength reporting thresholds Item information.

**Parameters**

| | |
|---|---|
| *pCDMARSSI-Thresh* | • CDMA RSSI threshold List<br>• See CDMARSSIThresh for more details |
| *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<br>• See CDMAECIOThresh for more details |
| *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<br>• See HDRRSSIThresh for more details |
| *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<br>• See HDRECIOThresh for more details |
| *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<br>• See HDRSINRThreshold for more details |
| *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<br>• See HDRIOThresh for more details |
| *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 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 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<br>• See WCDMAECIOThresh for more details |

| | |
|---|---|
| *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<br>• See LTERSSIThresh for more details |
| *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<br>• See LTESNRThreshold for more details |
| *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<br>• See LTERSRQThresh for more details |
| *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<br>• See LTERSRPThresh for more details |
| *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<br>• See LTESigRptConfig for more details |
| *pTDSCDMARS-CPThresh* | • TDSCDMA RSCP Threshold List<br>• See TDSCDMARSCPThresh for more details |
| *pTDSCDMARS-CPDelta* | • RSCP delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pTDSCDMARS-SIThresh* | • TDSCDMA RSSI Threshold List<br>• See TDSCDMARSSIThresh for more details |

| | |
|---|---|
| *pTDSCDMARSSIDelta* | • RSSI delta (in dBm) used by TD-SCDMA. |
| *pTDSCDMAECIOThresh* | • TDSCDMA ECIO Threshold List<br>• See TDSCDMAECIOThresh for more details |
| *pTDSCDMAECIODelta* | • ECIO delta (in dB) used by TD-SCDMA |
| *pTDSCDMASINRThresh* | • TDSCDMA SINR Threshold List<br>• See TDSCDMASINRThresh for more details |
| *pTDSCDMASINRDelta* | • SINR delta (in dB) used by TD-SCDMA. |

## 8.928.2 Field Documentation

### 8.928.2.1 WORD* setSignalStrengthInfo::pCDMAECIODelta

### 8.928.2.2 CDMAECIOThresh* setSignalStrengthInfo::pCDMAECIOThresh

### 8.928.2.3 WORD* setSignalStrengthInfo::pCDMARSSIDelta

### 8.928.2.4 CDMARSSIThresh* setSignalStrengthInfo::pCDMARSSIThresh

### 8.928.2.5 WORD* setSignalStrengthInfo::pGSMRSSIDelta

### 8.928.2.6 GSMRSSIThresh* setSignalStrengthInfo::pGSMRSSIThresh

### 8.928.2.7 WORD* setSignalStrengthInfo::pHDRECIODelta

### 8.928.2.8 HDRECIOThresh* setSignalStrengthInfo::pHDRECIOThresh

### 8.928.2.9 WORD* setSignalStrengthInfo::pHDRIODelta

### 8.928.2.10 HDRIOThresh* setSignalStrengthInfo::pHDRIOThresh

### 8.928.2.11 WORD* setSignalStrengthInfo::pHDRRSSIDelta

### 8.928.2.12 HDRRSSIThresh* setSignalStrengthInfo::pHDRRSSIThresh

### 8.928.2.13 WORD* setSignalStrengthInfo::pHDRSINRDelta

### 8.928.2.14 HDRSINRThreshold* setSignalStrengthInfo::pHDRSINRThresh

### 8.928.2.15 WORD* setSignalStrengthInfo::pLTERSRPDelta

### 8.928.2.16 LTERSRPThresh* setSignalStrengthInfo::pLTERSRPThresh

### 8.928.2.17 WORD* setSignalStrengthInfo::pLTERSRQDelta

#### 8.928.2.18 LTERSRQThresh∗ setSignalStrengthInfo::pLTERSRQThresh

#### 8.928.2.19 WORD∗ setSignalStrengthInfo::pLTERSSIDelta

#### 8.928.2.20 LTERSSIThresh∗ setSignalStrengthInfo::pLTERSSIThresh

#### 8.928.2.21 LTESigRptConfig∗ setSignalStrengthInfo::pLTESigRptConfig

#### 8.928.2.22 WORD∗ setSignalStrengthInfo::pLTESNRDelta

#### 8.928.2.23 LTESNRThreshold∗ setSignalStrengthInfo::pLTESNRThresh

#### 8.928.2.24 ULONG∗ setSignalStrengthInfo::pTDSCDMAECIODelta

#### 8.928.2.25 TDSCDMAECIOThresh∗ setSignalStrengthInfo::pTDSCDMAECIOThresh

#### 8.928.2.26 WORD∗ setSignalStrengthInfo::pTDSCDMARSCPDelta

#### 8.928.2.27 TDSCDMARSCPThresh∗ setSignalStrengthInfo::pTDSCDMARSCPThresh

#### 8.928.2.28 ULONG∗ setSignalStrengthInfo::pTDSCDMARSSIDelta

#### 8.928.2.29 TDSCDMARSSIThresh∗ setSignalStrengthInfo::pTDSCDMARSSIThresh

#### 8.928.2.30 ULONG∗ setSignalStrengthInfo::pTDSCDMASINRDelta

#### 8.928.2.31 TDSCDMASINRThresh∗ setSignalStrengthInfo::pTDSCDMASINRThresh

#### 8.928.2.32 WORD∗ setSignalStrengthInfo::pWCDMAECIODelta

#### 8.928.2.33 WCDMAECIOThresh∗ setSignalStrengthInfo::pWCDMAECIOThresh

#### 8.928.2.34 WORD∗ setSignalStrengthInfo::pWCDMARSSIDelta

#### 8.928.2.35 WCDMARSSIThresh∗ setSignalStrengthInfo::pWCDMARSSIThresh

## 8.929 SetSIPConfigReq Struct Reference

### Data Fields

- WORD ∗ pSIPLocalPort
- ULONG ∗ pTimerSIPReg
- ULONG ∗ pSubscribeTimer
- ULONG ∗ pTimerT1
- ULONG ∗ pTimerT2
- ULONG ∗ pTimerTf
- BYTE ∗ pSigCompEnabled

### 8.929.1 Detailed Description

This structure contains the SLQSSetSIPConfig request 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 |

### 8.929.2 Field Documentation

#### 8.929.2.1 BYTE* SetSIPConfigReq::pSigCompEnabled

#### 8.929.2.2 WORD* SetSIPConfigReq::pSIPLocalPort

#### 8.929.2.3 ULONG* SetSIPConfigReq::pSubscribeTimer

#### 8.929.2.4 ULONG* SetSIPConfigReq::pTimerSIPReg

#### 8.929.2.5 ULONG* SetSIPConfigReq::pTimerT1

#### 8.929.2.6 ULONG* SetSIPConfigReq::pTimerT2

#### 8.929.2.7 ULONG* SetSIPConfigReq::pTimerTf

## 8.930 SetSIPConfigResp Struct Reference

### Data Fields

- BYTE * pSettingResp

### 8.930.1 Detailed Description

This structure contains the SLQSSetSIPConfig response 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 |

### 8.930.2 Field Documentation

#### 8.930.2.1 BYTE* SetSIPConfigResp::pSettingResp

## 8.931 sGetDeviceSeriesResult Struct Reference

### Data Fields

- enum eGobiDeviceSeries eDevice
- ULONG uResult

### 8.931.1 Detailed Description

This structure contains the Device Series

**Parameters**

| | |
|---|---|
| *eGobiDeviceSeries* | • The number of device in the device series |
| *uResult* | -eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise |

### 8.931.2 Field Documentation

#### 8.931.2.1 enum eGobiDeviceSeries sGetDeviceSeriesResult::eDevice

#### 8.931.2.2 ULONG sGetDeviceSeriesResult::uResult

## 8.932 sidNid Struct Reference

### Data Fields

- WORD nid
- WORD sid

### 8.932.1 Detailed Description

This structure contains the parameters for SidNid Information

**Parameters**

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

### 8.932.2 Field Documentation

#### 8.932.2.1 WORD sidNid::nid

#### 8.932.2.2 WORD sidNid::sid

## 8.933 sigInfo Struct Reference

### Data Fields

- RSSIThresh * pRSSIThresh
- ECIOThresh * pECIOThresh
- HDRSINRThresh * pHDRSINRThresh
- LTESNRThresh * pLTESNRThresh
- IOThresh * pIOThresh
- RSRQThresh * pRSRQThresh
- RSRPThresh * pRSRPThresh
- LTESigRptCfg * pLTESigRptCfg
- TDSCDMASINRCONFThresh * pTDSCDMASINRCONFThresh

### 8.933.1 Detailed Description

This structure contains the 3gpp Configuration Item information.

**Parameters**

| | |
|---|---|
| *pRSSIThresh* | • RSSI threshold List<br>• See RSSIThresh for more details |
| *pECIOThresh* | • ECIO Threshold List<br>• See ECIOThresh for more details |
| *pHDRSINR-Thresh* | • HDR SINR Threshold List<br>• See HDRSINRThresh for more details |
| *pLTESNR-Thresh* | • LTE SNR Threshold List<br>• See LTESNRThresh for more details |
| *pIOThresh* | • IO Threshold List<br>• See IOThresh for more details |
| *pRSRQThresh* | • RSRQ Threshold List<br>• See RSRQThresh for more details |
| *pRSRPThresh* | • RSRP Threshold List<br>• See RSRPThresh for more details |

| | |
|---|---|
| *pLTESigRptCfg* | • LTE signal report config<br>• See LTESigRptCfg for more details |
| *pTDSCDMASIN-RCONFThresh* | • TD-SCDMA SINR Threshold List<br>• See TDSCDMASINRCONFThresh for more details |

### 8.933.2 Field Documentation

#### 8.933.2.1 ECIOThresh* sigInfo::pECIOThresh

#### 8.933.2.2 HDRSINRThresh* sigInfo::pHDRSINRThresh

#### 8.933.2.3 IOThresh* sigInfo::pIOThresh

#### 8.933.2.4 LTESigRptCfg* sigInfo::pLTESigRptCfg

#### 8.933.2.5 LTESNRThresh* sigInfo::pLTESNRThresh

#### 8.933.2.6 RSRPThresh* sigInfo::pRSRPThresh

#### 8.933.2.7 RSRQThresh* sigInfo::pRSRQThresh

#### 8.933.2.8 RSSIThresh* sigInfo::pRSSIThresh

#### 8.933.2.9 TDSCDMASINRCONFThresh* sigInfo::pTDSCDMASINRCONFThresh

## 8.934 signalInfo Struct Reference

### Data Fields

- BYTE signalType
- BYTE alertPitch
- BYTE signal

### 8.934.1 Detailed Description

This structure contains Signal Information

**Parameters**

| | |
|---|---|
| *signalType* | • Call identifier for the call. |
| *alertPitch* | • Signal Information |
| *signal* | • Caller ID Information |

### 8.934.2 Field Documentation

#### 8.934.2.1 BYTE signalInfo::alertPitch

#### 8.934.2.2 BYTE signalInfo::signal

#### 8.934.2.3 BYTE signalInfo::signalType

## 8.935 SignalStrengthDataType Struct Reference

### Data Fields

- BYTE thresholdsSize
- INT8 thresholds [5]

### 8.935.1 Field Documentation

#### 8.935.1.1 INT8 SignalStrengthDataType::thresholds[5]

#### 8.935.1.2 BYTE SignalStrengthDataType::thresholdsSize

## 8.936 slot_t Struct Reference

### Data Fields

- uint32_t uPhyCardStatus
- uint32_t uPhySlotStatus
- uint8_t bLogicalSlot
- uint8_t bICCIDLength
- uint8_t bICCID [255]

### 8.936.1 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. |

### 8.936.2 Field Documentation

#### 8.936.2.1 uint8_t slot_t::bICCID[255]

#### 8.936.2.2 uint8_t slot_t::bICCIDLength

#### 8.936.2.3 uint8_t slot_t::bLogicalSlot

#### 8.936.2.4 uint32_t slot_t::uPhyCardStatus

#### 8.936.2.5 uint32_t slot_t::uPhySlotStatus

## 8.937 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]

### 8.937.1 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. |

## 8.937.2 Field Documentation

### 8.937.2.1 appStats slotInf::AppStatus[10]

### 8.937.2.2 uint8_t slotInf::cardState

### 8.937.2.3 uint8_t slotInf::errorState

### 8.937.2.4 uint8_t slotInf::numApp

#### 8.937.2.5 uint8_t slotInf::upinRetries

#### 8.937.2.6 uint8_t slotInf::upinState

#### 8.937.2.7 uint8_t slotInf::upukRetries

## 8.938 slotInfo Struct Reference

### Data Fields

- BYTE cardState
- BYTE upinState
- BYTE upinRetries
- BYTE upukRetries
- BYTE errorState
- BYTE numApp
- appStatus AppStatus [10]

### 8.938.1 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 appStatus for more information. |

### 8.938.2 Field Documentation

#### 8.938.2.1 appStatus slotInfo::AppStatus[10]

#### 8.938.2.2 BYTE slotInfo::cardState

#### 8.938.2.3 BYTE slotInfo::errorState

#### 8.938.2.4 BYTE slotInfo::numApp

#### 8.938.2.5 BYTE slotInfo::upinRetries

#### 8.938.2.6 BYTE slotInfo::upinState

#### 8.938.2.7 BYTE slotInfo::upukRetries

## 8.939 slots_t Struct Reference

### Data Fields

- slot_t uimSlotStatus [255]

### 8.939.1 Detailed Description

This structure contains slots status.

**Parameters**

| | |
|---|---|
| *uimSlotStatus* | • See slot_t |

### 8.939.2 Field Documentation

#### 8.939.2.1 slot_t slots_t::uimSlotStatus[255]

## 8.940 slqsautoconnect Struct Reference

### Data Fields

- BOOL action
- BYTE acsetting
- BYTE acroamsetting

### 8.940.1 Detailed Description

structure contains autoconnect settings parameters

**Parameters**

| | |
|---|---|
| *action* | • 0 - get autoconnect settings<br>• 1 - set autoconnect settings |
| *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 |

### 8.940.2 Field Documentation

#### 8.940.2.1 BYTE slqsautoconnect::acroamsetting

#### 8.940.2.2 BYTE slqsautoconnect::acsetting

#### 8.940.2.3 BOOL slqsautoconnect::action

## 8.941 SLQSDeleteProfileParams Struct Reference

### Data Fields

- BYTE profileType

- BYTE profileIndex

### 8.941.1 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) |

### 8.941.2 Field Documentation

#### 8.941.2.1 BYTE SLQSDeleteProfileParams::profileIndex

#### 8.941.2.2 BYTE SLQSDeleteProfileParams::profileType

## 8.942 slqsfwinfo_s 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]

### 8.942.1 Detailed Description

SPKG CWE firmware image info structure

**Parameters**

| | |
|---|---|
| *modelid_str* | • device model identifier string |
| *bootversion_str* | • firmware boot version string |
| *appversion_str* | • firmware application version string |

| | |
|---|---|
| *sku_str* | • SKU(PRI) string |
| *packageid_str* | • package identifier string<br>• deprecated on EM/MC74xx(9x30) devices |
| *carrier_str* | • carrier string<br>• See qaGobiApiTableCarrierCodes.h for carrier codes |
| *priversion_str* | • PRI version string |
| *cur_carr_name* | • Current PRI Carrier Name |
| *cur_carr_rev* | • Current PRI Carrier Revision |

### 8.942.2 Field Documentation

#### 8.942.2.1 CHAR slqsfwinfo_s::appversion_str[85]

#### 8.942.2.2 CHAR slqsfwinfo_s::bootversion_str[85]

#### 8.942.2.3 CHAR slqsfwinfo_s::carrier_str[20]

#### 8.942.2.4 CHAR slqsfwinfo_s::cur_carr_name[17]

#### 8.942.2.5 CHAR slqsfwinfo_s::cur_carr_rev[13]

#### 8.942.2.6 CHAR slqsfwinfo_s::modelid_str[20]

#### 8.942.2.7 CHAR slqsfwinfo_s::packageid_str[85]

#### 8.942.2.8 CHAR slqsfwinfo_s::priversion_str[16]

#### 8.942.2.9 CHAR slqsfwinfo_s::sku_str[15]

## 8.943 SlqsNas3GppNetworkInfo Struct Reference

### Data Fields

- WORD MCC
- WORD MNC
- ULONG InUse
- ULONG Roaming
- ULONG Forbidden
- ULONG Preferred
- CHAR Description [255]

### 8.943.1 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. |

### 8.943.2 Field Documentation

#### 8.943.2.1 CHAR SlqsNas3GppNetworkInfo::Description[255]

#### 8.943.2.2 ULONG SlqsNas3GppNetworkInfo::Forbidden

#### 8.943.2.3 ULONG SlqsNas3GppNetworkInfo::InUse

#### 8.943.2.4 WORD SlqsNas3GppNetworkInfo::MCC

#### 8.943.2.5 WORD SlqsNas3GppNetworkInfo::MNC

#### 8.943.2.6 ULONG SlqsNas3GppNetworkInfo::Preferred

#### 8.943.2.7 ULONG SlqsNas3GppNetworkInfo::Roaming

## 8.944 SlqsNasPcsDigit Struct Reference

### Data Fields

- WORD MCC
- WORD MNC
- BYTE includes_pcs_digit

### 8.944.1 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 |

### 8.944.2 Field Documentation

#### 8.944.2.1 BYTE SlqsNasPcsDigit::includes_pcs_digit

#### 8.944.2.2 WORD SlqsNasPcsDigit::MCC

#### 8.944.2.3 WORD SlqsNasPcsDigit::MNC

## 8.945 slqssendasyncsmsparams_s Struct Reference

### Data Fields

- ULONG messageFormat
- ULONG messageSize
- BYTE * pMessage
- BYTE * pForceOnDC
- BYTE * pServiceOption
- BYTE * pFollowOnDC
- BYTE * pLinktimer
- BYTE * pSmsOnIms
- BYTE * pRetryMessage
- ULONG * pRetryMessageId
- ULONG * pUserData

### 8.945.1 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. |
| *pRetryMessage-Id* | • 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. |

### 8.945.2 Field Documentation

#### 8.945.2.1 ULONG slqssendasyncsmsparams_s::messageFormat

#### 8.945.2.2 ULONG slqssendasyncsmsparams_s::messageSize

#### 8.945.2.3 BYTE* slqssendasyncsmsparams_s::pFollowOnDC

#### 8.945.2.4 BYTE* slqssendasyncsmsparams_s::pForceOnDC

#### 8.945.2.5 BYTE* slqssendasyncsmsparams_s::pLinktimer

#### 8.945.2.6 BYTE* slqssendasyncsmsparams_s::pMessage

#### 8.945.2.7 BYTE* slqssendasyncsmsparams_s::pRetryMessage

#### 8.945.2.8 ULONG* slqssendasyncsmsparams_s::pRetryMessageId

#### 8.945.2.9 BYTE* slqssendasyncsmsparams_s::pServiceOption

#### 8.945.2.10 BYTE* slqssendasyncsmsparams_s::pSmsOnIms

#### 8.945.2.11 ULONG* slqssendasyncsmsparams_s::pUserData

## 8.946 slqssendsmsparams_s Struct Reference

### Data Fields

- ULONG messageFormat
- ULONG messageSize
- BYTE * pMessage
- USHORT messageID
- ULONG messageFailureCode
- BYTE * pLinktimer
- BYTE * pSmsOnIms

### 8.946.1 Detailed Description

This structure contains SMS parameters

**Parameters**

| | |
|---|---|
| *messageFormat[IN]* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *messageSize[IN]* | • The length of the message contents in bytes |
| *pMessage[IN]* | • The message contents in PDU format contains SMS header and payload message |
| *pMessageID[OUT]* | • message reference ID |
| *pMessageFailureCode[OUT]* | • (Optional) Message Failure Code<br>• If cause code is not provided, then value will be 0xFFFFFFFF |
| *pSmsOnIms[IN]* | • (Optional) SMS on IMS<br>• 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 |
| *pLinktimer[IN]* | • (Optional) Link Timer<br>• Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow |

### 8.946.2 Field Documentation

#### 8.946.2.1 ULONG slqssendsmsparams_s::messageFailureCode

#### 8.946.2.2 ULONG slqssendsmsparams_s::messageFormat

#### 8.946.2.3 USHORT slqssendsmsparams_s::messageID

#### 8.946.2.4 ULONG slqssendsmsparams_s::messageSize

#### 8.946.2.5 BYTE* slqssendsmsparams_s::pLinktimer

#### 8.946.2.6 BYTE* slqssendsmsparams_s::pMessage

#### 8.946.2.7 BYTE* slqssendsmsparams_s::pSmsOnIms

## 8.947 slqsSessionStateInfo Struct Reference

### Data Fields

- qaQmiInterfaceInfo * pQmiInterfaceInfo
- ULONG reconfiguration_required
- ULONG state
- ULONG sessionEndReason

### 8.947.1 Detailed Description

Contains the session state information and information about the interface

**Parameters**

| | |
|---|---|
| *pQmiInterface-Info* | • See qaQmiInterfaceInfo for more information |
| *state* | • Current Link Status<br>– 1 Disconnected<br>– 2 Connected<br>– 3 Suspended (Unsupported)<br>– 4 Authenticating |
| *reconfiguration_-required* | • Indicates if host needs to be reconfigured<br>– 0 No need to reconfigure<br>– 1 Reconfiguration required |
| *sessionEnd-Reason* | • See qaGobiApiTableCallEndReasons.h for Call End Reason |

### 8.947.2 Field Documentation

#### 8.947.2.1 qaQmiInterfaceInfo* slqsSessionStateInfo::pQmiInterfaceInfo

#### 8.947.2.2 ULONG slqsSessionStateInfo::reconfiguration_required

#### 8.947.2.3 ULONG slqsSessionStateInfo::sessionEndReason

#### 8.947.2.4 ULONG slqsSessionStateInfo::state

## 8.948 slqsSignalStrengthInfo Struct Reference

### Data Fields

- USHORT signalStrengthReqMask
- USHORT rxSignalStrengthListLen
- struct rxSignalStrengthListElement rxSignalStrengthList [18]
- USHORT ecioListLen
- struct ecioListElement ecioList [18]

- INT32 Io
- BYTE sinr
- USHORT errorRateListLen
- struct errorRateListElement errorRateList [18]
- struct rsrqInformation rsrqInfo
- SHORT lteSnr
- SHORT lteRsrp

### 8.948.1 Detailed Description

This structure contains the Signal Strength Information

**Parameters**

| | |
|---|---|
| *signalStrength-ReqMask[IN]* | • 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 |
| *rxSignal-StrengthList-Len[OUT]* | • Number of elements in Receive Signal Strength List |
| *rxSignal-StrengthList[OU-T]* | • See rxSignalStrengthListElement for more information |
| *ecioListLen[OU-T]* | • Number of elements in ECIO List |

| | |
|---|---|
| *ecioList[OUT]* | • See ecioListElement for more information |
| *Io[OUT]* | • Received Io in dBm; IO is only applicable for 1xEV-DO |
| *sinr[OUT]* | • 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 |
| *errorRateList-Len[OUT]* | • Number of elements in Error Rate List |
| *errorRateList[OUT]* | • See errorRateListElement for more information |
| *rsrqInfo[OUT]* | • See rsrqInformation for more information |
| *ltesnr[OUT]* | • 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 |
| *ltersrp[OUT]* | • 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 |

## 8.948.2 Field Documentation

### 8.948.2.1 struct ecioListElement slqsSignalStrengthInfo::ecioList[18]

### 8.948.2.2 USHORT slqsSignalStrengthInfo::ecioListLen

### 8.948.2.3 struct errorRateListElement slqsSignalStrengthInfo::errorRateList[18]

### 8.948.2.4 USHORT slqsSignalStrengthInfo::errorRateListLen

### 8.948.2.5 INT32 slqsSignalStrengthInfo::Io

### 8.948.2.6 SHORT slqsSignalStrengthInfo::ltersrp

### 8.948.2.7 SHORT slqsSignalStrengthInfo::ltesnr

### 8.948.2.8 struct rsrqInformation slqsSignalStrengthInfo::rsrqInfo

### 8.948.2.9 struct rxSignalStrengthListElement slqsSignalStrengthInfo::rxSignalStrengthList[18]

### 8.948.2.10 USHORT slqsSignalStrengthInfo::rxSignalStrengthListLen

#### 8.948.2.11 USHORT slqsSignalStrengthInfo::signalStrengthReqMask

#### 8.948.2.12 BYTE slqsSignalStrengthInfo::sinr

## 8.949 SLQSSignalStrengthsIndReq Struct Reference

### Data Fields

- BYTE rxSignalStrengthDelta
- BYTE ecioDelta
- BYTE ioDelta
- BYTE sinrDelta
- BYTE rsrqDelta
- BYTE ecioThresholdListLen
- SHORT ecioThresholdList [10]
- BYTE sinrThresholdListLen
- BYTE sinrThresholdList [5]
- WORD lteSnrDelta
- BYTE lteRsrpDelta

### 8.949.1 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. |

**Note**

None

### 8.949.2 Field Documentation

#### 8.949.2.1 BYTE SLQSSignalStrengthsIndReq::ecioDelta

#### 8.949.2.2 SHORT SLQSSignalStrengthsIndReq::ecioThresholdList[10]

#### 8.949.2.3 BYTE SLQSSignalStrengthsIndReq::ecioThresholdListLen

#### 8.949.2.4 BYTE SLQSSignalStrengthsIndReq::ioDelta

#### 8.949.2.5 BYTE SLQSSignalStrengthsIndReq::lteRsrpDelta

#### 8.949.2.6 WORD SLQSSignalStrengthsIndReq::lteSnrDelta

#### 8.949.2.7 BYTE SLQSSignalStrengthsIndReq::rsrqDelta

#### 8.949.2.8 BYTE SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

#### 8.949.2.9 BYTE SLQSSignalStrengthsIndReq::sinrDelta

#### 8.949.2.10 BYTE SLQSSignalStrengthsIndReq::sinrThresholdList[5]

#### 8.949.2.11 BYTE SLQSSignalStrengthsIndReq::sinrThresholdListLen

# 8.950 SLQSSignalStrengthsInformation Struct Reference

## Data Fields

- struct rxSignalStrengthListElement rxSignalStrengthInfo
- struct ecioListElement ecioInfo
- ULONG io
- BYTE sinr
- struct errorRateListElement errorRateInfo
- struct rsrqInformation rsrqInfo
- struct lteSnrinformation lteSnrinfo
- struct lteRsrpinformation lteRsrpinfo

### 8.950.1 Detailed Description

Structure for Received Signal Strength Information.

**Parameters**

| | |
|---|---|
| *rxSignal-StrengthInfo* | • See rxSignalStrengthListElement for more information. |
| *ecioInfo* | • See ecioListElement for more information. |
| *io* | • Received IO in dBm; IO is only applicable for 1xEV-DO. |
| *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 |
| *errorRateInfo* | • See errorRateListElement for more information. |
| *rsrqInfo* | • See rsrqInformation for more information. |
| *lteSnrinfo* | • See lteSnrinformation for more information. |
| *lteRsrpinfo* | • See lteRsrpinformation for more information. |

**Note**

None

### 8.950.2 Field Documentation

#### 8.950.2.1 struct ecioListElement SLQSSignalStrengthsInformation::ecioInfo

#### 8.950.2.2 struct errorRateListElement SLQSSignalStrengthsInformation::errorRateInfo

#### 8.950.2.3 ULONG SLQSSignalStrengthsInformation::io

#### 8.950.2.4 struct lteRsrpinformation SLQSSignalStrengthsInformation::lteRsrpinfo

#### 8.950.2.5 struct lteSnrinformation SLQSSignalStrengthsInformation::lteSnrinfo

#### 8.950.2.6 struct rsrqInformation SLQSSignalStrengthsInformation::rsrqInfo

#### 8.950.2.7 struct rxSignalStrengthListElement SLQSSignalStrengthsInformation::rxSignalStrengthInfo

#### 8.950.2.8 BYTE SLQSSignalStrengthsInformation::sinr

## 8.951 sIqsWdsEventInfo Struct Reference

### Data Fields

- qaQmiInterfaceInfo * pQmiInterfaceInfo
- ULONG * pDormancyStatus
- ULONG * pDataBearer
- ULONG * pPacketsCountTX
- ULONG * pPacketsCountRX
- ULONGLONG * pTotalBytesTX
- ULONGLONG * pTotalBytesRX

### 8.951.1 Detailed Description

Contains the WDS event information and information about the interface

**Parameters**

| | |
|---|---|
| *pQmiInterface-Info* | • See qaQmiInterfaceInfo for more information |
| *pDataBearer,-* | Data bearer technology (NULL if not present)<br>• 0x00 - Indicates that this field is ignored<br>• 0x01 - CDMA 1X<br>• 0x02 - EV-DO Rev 0<br>• 0x03 - GPRS<br>• 0x04 - WCDMA<br>• 0x05 - EV-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 - EV-DO Rev A EHRPD<br>• 0x0C - HSDPA+ and WCDMA<br>• 0x0D - HSDPA+ and HSUPA<br>• 0x0E - DC_HSDPA+ and WCDMA<br>• 0x0F - DC_HSDPA+ and HSUPA<br>• 0x8000 - NULL Bearer<br>• 0xFF - Unknown Technology |
| *pDormancy-Status* | • Dormancy status (NULL if not present)<br>– 1 - traffic channel dormant<br>– 2 - traffic channel active |
| *pPacketsCount-TX* | • Packets transmitted without error (NULL if not present) |
| *pPacketsCount-RX* | • Packets received without error (NULL if not present) |
| *pTotalBytesTX* | • Bytes transmitted without error (NULL if not present) |
| *pTotalBytesRX* | • Bytes received without error (NULL if not present) |

## 8.951.2 Field Documentation

### 8.951.2.1 ULONG* slqsWdsEventInfo::pDataBearer

### 8.951.2.2 ULONG* slqsWdsEventInfo::pDormancyStatus

### 8.951.2.3 ULONG* slqsWdsEventInfo::pPacketsCountRX

### 8.951.2.4 ULONG* slqsWdsEventInfo::pPacketsCountTX

#### 8.951.2.5 qaQmiInterfaceInfo* slqsWdsEventInfo::pQmiInterfaceInfo

#### 8.951.2.6 ULONGLONG* slqsWdsEventInfo::pTotalBytesRX

#### 8.951.2.7 ULONGLONG* slqsWdsEventInfo::pTotalBytesTX

## 8.952 sms_BroadcastConfig Struct Reference

### Data Fields

- uint16_t fromServiceId
- uint16_t toServiceId
- uint8_t selected

### 8.952.1 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 |

### 8.952.2 Field Documentation

#### 8.952.2.1 uint16_t sms_BroadcastConfig::fromServiceId

#### 8.952.2.2 uint8_t sms_BroadcastConfig::selected

#### 8.952.2.3 uint16_t sms_BroadcastConfig::toServiceId

## 8.953 sms_CDMABroadcastConfig Struct Reference

### Data Fields

- uint16_t serviceCategory
- uint16_t language
- uint8_t selected

### 8.953.1 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 |

### 8.953.2 Field Documentation

#### 8.953.2.1 uint16_t sms_CDMABroadcastConfig::language

#### 8.953.2.2 uint8_t sms_CDMABroadcastConfig::selected

#### 8.953.2.3 uint16_t sms_CDMABroadcastConfig::serviceCategory

## 8.954 sms_getIndicationReg Struct Reference

### Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

### 8.954.1 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<br>• returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from 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>• returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |

| | |
|---|---|
| *pRegCallStat-InfoEvt* | - Optional parameter indicating registration status of call status information events<br>- Values:<br>  – 0x00 - Disabled<br>  – 0x01 - Enabled<br>- returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |

### 8.954.2 Field Documentation

#### 8.954.2.1 uint8_t* sms_getIndicationReg::pRegCallStatInfoEvt

#### 8.954.2.2 uint8_t* sms_getIndicationReg::pRegTransLayerInfoEvt

#### 8.954.2.3 uint8_t* sms_getIndicationReg::pRegTransNWRegInfoEvt

## 8.955 sms_getMsgWaitingInfo Struct Reference

### Data Fields

- uint8_t numInstances
- sms_messageWaitingInfoContent msgWaitInfo [0xFF]

### 8.955.1 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. |

### 8.955.2 Field Documentation

#### 8.955.2.1 sms_messageWaitingInfoContent sms_getMsgWaitingInfo::msgWaitInfo[0xFF]

#### 8.955.2.2 uint8_t sms_getMsgWaitingInfo::numInstances

## 8.956 sms_getTransLayerInfo Struct Reference

### Data Fields

- uint8_t * pRegInd
- sms_transLayerInfo * pTransLayerInfo

### 8.956.1 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.<br>• NULL pointer - Invalid data. |
| *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.<br>• NULL pointer - Invalid data. |

### 8.956.2 Field Documentation

#### 8.956.2.1 uint8_t* sms_getTransLayerInfo::pRegInd

#### 8.956.2.2 sms_transLayerInfo* sms_getTransLayerInfo::pTransLayerInfo

## 8.957 sms_getTransNWRegInfo Struct Reference

### Data Fields

- uint8_t * pRegStatus

### 8.957.1 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. |

### 8.957.2 Field Documentation

#### 8.957.2.1 uint8_t* sms_getTransNWRegInfo::pRegStatus

## 8.958 sms_maxStorageSizeReq Struct Reference

### Data Fields

- uint8_t storageType
- uint8_t * pMessageMode

### 8.958.1 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

### 8.958.2 Field Documentation

#### 8.958.2.1 uint8_t* sms_maxStorageSizeReq::pMessageMode

#### 8.958.2.2 uint8_t sms_maxStorageSizeReq::storageType

## 8.959 sms_maxStorageSizeResp Struct Reference

### Data Fields

- uint32_t maxStorageSize
- uint32_t freeSlots

### 8.959.1 Detailed Description

This structure contains get store max size response parameters

**Parameters**

| | |
|---|---|
| *maxStorageSize* | -<br>• Memory Store Size |

| | |
|---|---|
| *freeSlots* | - 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. |

### 8.959.2 Field Documentation

#### 8.959.2.1 uint32_t sms_maxStorageSizeResp::freeSlots

#### 8.959.2.2 uint32_t sms_maxStorageSizeResp::maxStorageSize

## 8.960 sms_messageWaitingInfoContent Struct Reference

### Data Fields

- uint8_t msgType
- uint8_t activeInd
- uint8_t msgCount

### 8.960.1 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 |

### 8.960.2 Field Documentation

#### 8.960.2.1 uint8_t sms_messageWaitingInfoContent::activeInd

#### 8.960.2.2 uint8_t sms_messageWaitingInfoContent::msgCount

#### 8.960.2.3 uint8_t sms_messageWaitingInfoContent::msgType

## 8.961 sms_msgProtocolResp Struct Reference

### Data Fields

- uint8_t msgProtocol

### 8.961.1 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 |

### 8.961.2 Field Documentation

#### 8.961.2.1 uint8_t sms_msgProtocolResp::msgProtocol

## 8.962 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference

### Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_CDMABroadcastConfig CDMABroadcastConfig [0x05]

### 8.962.1 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 |

### 8.962.2 Field Documentation

#### 8.962.2.1 uint8_t sms_qaQmi3GPP2BroadcastCfgInfo::activated_ind

#### 8.962.2.2 sms_CDMABroadcastConfig sms_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]

#### 8.962.2.3 uint16_t sms_qaQmi3GPP2BroadcastCfgInfo::num_instances

## 8.963 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference

### Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_BroadcastConfig broadcastConfig [0x05]

### 8.963.1 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 |

### 8.963.2 Field Documentation

#### 8.963.2.1 uint8_t sms_qaQmi3GPPBroadcastCfgInfo::activated_ind

#### 8.963.2.2 sms_BroadcastConfig sms_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

#### 8.963.2.3 uint16_t sms_qaQmi3GPPBroadcastCfgInfo::num_instances

## 8.964 sms_routeEntry Struct Reference

### Data Fields

- uint8_t messageType
- uint8_t messageClass

- uint8_t routeStorage
- uint8_t receiptAction

### 8.964.1 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) |

### 8.964.2 Field Documentation

#### 8.964.2.1 uint8_t sms_routeEntry::messageClass

#### 8.964.2.2 uint8_t sms_routeEntry::messageType

#### 8.964.2.3 uint8_t sms_routeEntry::receiptAction

#### 8.964.2.4 uint8_t sms_routeEntry::routeStorage

## 8.965 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

## 8.965.1 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. |
| *pRetryMessage-Id* | • 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. |

### 8.965.2 Field Documentation

#### 8.965.2.1 uint32_t sms_sendAsyncsmsParams::messageFormat

#### 8.965.2.2 uint32_t sms_sendAsyncsmsParams::messageSize

#### 8.965.2.3 uint8_t* sms_sendAsyncsmsParams::pFollowOnDC

#### 8.965.2.4 uint8_t* sms_sendAsyncsmsParams::pForceOnDC

#### 8.965.2.5 uint8_t* sms_sendAsyncsmsParams::pLinktimer

#### 8.965.2.6 uint8_t* sms_sendAsyncsmsParams::pMessage

#### 8.965.2.7 uint8_t* sms_sendAsyncsmsParams::pRetryMessage

#### 8.965.2.8 uint32_t* sms_sendAsyncsmsParams::pRetryMessageId

#### 8.965.2.9 uint8_t* sms_sendAsyncsmsParams::pServiceOption

#### 8.965.2.10 uint8_t* sms_sendAsyncsmsParams::pSmsOnIms

#### 8.965.2.11 uint32_t* sms_sendAsyncsmsParams::pUserData

## 8.966 sms_setIndicationReg Struct Reference

### Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

### 8.966.1 Detailed Description

This structure contains Indication Register request parameters

**Parameters**

| | |
|---|---|
| *pRegTransLayerInfoEvt* | - 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 |
| *pRegTransNWRegInfoEvt* | - 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 |
| *pRegCallStatInfoEvt* | - 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 |

### 8.966.2 Field Documentation

#### 8.966.2.1 uint8_t* sms_setIndicationReg::pRegCallStatInfoEvt

#### 8.966.2.2 uint8_t* sms_setIndicationReg::pRegTransLayerInfoEvt

#### 8.966.2.3 uint8_t* sms_setIndicationReg::pRegTransNWRegInfoEvt

## 8.967 sms_setRoutesReq Struct Reference

### Data Fields

- uint16_t numOfRoutes

- sms_routeEntry routeList [0x0A]
- uint8_t * pTransferStatusReport

### 8.967.1 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) |

### 8.967.2 Field Documentation

#### 8.967.2.1 uint16_t sms_setRoutesReq::numOfRoutes

#### 8.967.2.2 uint8_t* sms_setRoutesReq::pTransferStatusReport

#### 8.967.2.3 sms_routeEntry sms_setRoutesReq::routeList[0x0A]

## 8.968 sms_transLayerInfo Struct Reference

### Data Fields

- uint8_t TransType
- uint8_t TransCap

### 8.968.1 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 |

### 8.968.2 Field Documentation

#### 8.968.2.1 uint8_t sms_transLayerInfo::TransCap

#### 8.968.2.2 uint8_t sms_transLayerInfo::TransType

## 8.969 SMSAsyncRawSend_s Struct Reference

### Data Fields

- WORD sendStatus
- WORD messageID
- WORD causeCode
- BYTE errorClass
- WORD RPCause
- BYTE TPCause
- BYTE msgDelFailureType
- BYTE msgDelFailureCause
- BYTE alphaIDLen
- BYTE * pAlphaID
- ULONG userData

### 8.969.1 Detailed Description

This structure contains SMS parameters

**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 |
| *messageID* | • Unique ID assigned by WMS for non-retry messages. |
| *causeCode* | • WMS cause code |
| *errorClass* | • Error Class<br>• Values:<br>– 0x00 - ERROR_CLASS_TEMPORARY<br>– 0x01 - ERROR_CLASS_PERMANENT |
| *RPCause* | • GW RP cause |
| *TPCause* | • GW TP Cause |

| | |
|---|---|
| *msgDelFailure-Type* | • Message delivery failure type<br>• Values:<br>– 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY<br>– 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT |
| *msgDelFailure-Cause* | • Message delivery failure cause<br>• Values:<br>– 0x00 - WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL |
| *alphaIDLen* | • Number of sets of the pAlphaID |
| *pAlphaID* | • Alpha ID |
| *userData* | • Identifies the request associated with this indication. |

### 8.969.2 Field Documentation

#### 8.969.2.1 BYTE SMSAsyncRawSend_s::alphaIDLen

#### 8.969.2.2 WORD SMSAsyncRawSend_s::causeCode

#### 8.969.2.3 BYTE SMSAsyncRawSend_s::errorClass

#### 8.969.2.4 WORD SMSAsyncRawSend_s::messageID

#### 8.969.2.5 BYTE SMSAsyncRawSend_s::msgDelFailureCause

#### 8.969.2.6 BYTE SMSAsyncRawSend_s::msgDelFailureType

#### 8.969.2.7 BYTE* SMSAsyncRawSend_s::pAlphaID

#### 8.969.2.8 WORD SMSAsyncRawSend_s::RPCause

#### 8.969.2.9 WORD SMSAsyncRawSend_s::sendStatus

#### 8.969.2.10 BYTE SMSAsyncRawSend_s::TPCause

#### 8.969.2.11 ULONG SMSAsyncRawSend_s::userData

## 8.970 SMSCAddress Struct Reference

### Data Fields

- BYTE length
- BYTE data [256]

### 8.970.1 Detailed Description

This structure holds SMSC information

**Parameters**

| | |
|---|---|
| *length* | • Number of sets of following element |
| *data* | • SMSC address |

### 8.970.2 Field Documentation

#### 8.970.2.1 BYTE SMSCAddress::data[256]

#### 8.970.2.2 BYTE SMSCAddress::length

## 8.971 sMSCAddressInfo Struct Reference

### Data Fields

- uint8_t length
- uint8_t data [256]

### 8.971.1 Detailed Description

This structure contains SMSC information

**Parameters**

| | |
|---|---|
| *length* | • Number of sets of following element |
| *data* | • SMSC address |

### 8.971.2 Field Documentation

#### 8.971.2.1 uint8_t sMSCAddressInfo::data[256]

#### 8.971.2.2 uint8_t sMSCAddressInfo::length

## 8.972 sMSCAddressTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSCAddressInfo SMSCInfo

### 8.972.1 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 |

### 8.972.2 Field Documentation

#### 8.972.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

#### 8.972.2.2 uint8_t sMSCAddressTlv::TlvPresent

## 8.973 SMSEtwsMessage Struct Reference

### Data Fields

- BYTE notificationType
- WORD length
- BYTE data [1254]

### 8.973.1 Detailed Description

This structure holds information related earthquake and Tsunami warning system

**Parameters**

| | |
|---|---|
| *notificationType* | • Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS |
| *length* | • Number of sets of following elements |
| *data* | • Raw message data |

### 8.973.2 Field Documentation

#### 8.973.2.1 BYTE SMSEtwsMessage::data[1254]

#### 8.973.2.2 WORD SMSEtwsMessage::length

#### 8.973.2.3 BYTE SMSEtwsMessage::notificationType

## 8.974 sMSEtwsMessageInfo Struct Reference

### Data Fields

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

### 8.974.1 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 |

### 8.974.2 Field Documentation

#### 8.974.2.1 uint8_t sMSEtwsMessageInfo::data[1254]

#### 8.974.2.2 uint16_t sMSEtwsMessageInfo::length

#### 8.974.2.3 uint8_t sMSEtwsMessageInfo::notificationType

## 8.975 sMSEtwsMessageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSEtwsMessageInfo EtwsMessageInfo

### 8.975.1 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 |

### 8.975.2 Field Documentation

#### 8.975.2.1 sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo

#### 8.975.2.2 uint8_t sMSEtwsMessageTlv::TlvPresent

## 8.976 SMSEtwsPlmn Struct Reference

### Data Fields

- WORD mobileCountryCode
- WORD mobileNetworkCode

### 8.976.1 Detailed Description

This structure holds information related ETWS PLMN

**Parameters**

| | |
|---|---|
| *mobileCountry-Code* | • 16 bit representation of MCC value range : 0 -999 |
| *mobileNetwork-Code* | • 16 bit representation of MNC value range : 0 -999 |

### 8.976.2 Field Documentation

#### 8.976.2.1 WORD SMSEtwsPlmn::mobileCountryCode

#### 8.976.2.2 WORD SMSEtwsPlmn::mobileNetworkCode

## 8.977 sMSEtwsPlmnInfo Struct Reference

### Data Fields

- uint16_t mobileCountryCode
- uint16_t mobileNetworkCode

### 8.977.1 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 |

### 8.977.2 Field Documentation

#### 8.977.2.1 uint16_t sMSEtwsPlmnInfo::mobileCountryCode

#### 8.977.2.2 uint16_t sMSEtwsPlmnInfo::mobileNetworkCode

## 8.978 SMSEventInfo_s Struct Reference

### Data Fields

- BYTE smsEventType
- SMSMTMessageInfo * pMTMessageInfo
- SMSTransferRouteMTMessageInfo * pTransferRouteMTMessageInfo
- SMSMessageModeInfo * pMessageModeInfo
- SMSEtwsMessageInfo * pEtwsMessageInfo
- SMSEtwsPlmnInfo * pEtwsPlmnInfo
- SMSCAddressInfo * pSMSCAddressInfo
- SMSOnIMSInfo * pSMSOnIMSInfo

### 8.978.1 Detailed Description

This structure will hold the information related to received SMS events

**Parameters**

| | |
|---|---|
| *smsEventType* | • Type of the SMS events that are received. This is a bit map of SMSEventType. Only the parameters (which follows) related to the events received would be filled, and the rest of the parameters would be NULL |
| *pMTMessage-Info* | • pointer to the SMSMTMessageInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pTransferRoute-MTMessageInfo* | • pointer to the SMSTransferRouteMTMessageInfo structure<br>. NULL, if this event is not present in the smsEventType parameter |
| *pMessageMode-Info* | • pointer to the SMSMessageModeInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pEtwsMessage-Info* | • pointer to the SMSEtwsMessageInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pEtwsPlmnInfo* | • pointer to the SMSEtwsPlmnInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pSMSCAddress-Info* | • pointer to the SMSCAddressInfo structure<br>NULL, if this event is not present in the smsEventType parameter |

| | |
|---|---|
| *pSMSOnIMSInfo* | • pointer to the SMSOnIMSInfo structure<br>NULL, if this event is not present in the smsEventType parameter Note: None |

### 8.978.2 Field Documentation

#### 8.978.2.1 SMSEtwsMessageInfo∗ SMSEventInfo_s::pEtwsMessageInfo

#### 8.978.2.2 SMSEtwsPlmnInfo∗ SMSEventInfo_s::pEtwsPlmnInfo

#### 8.978.2.3 SMSMessageModeInfo∗ SMSEventInfo_s::pMessageModeInfo

#### 8.978.2.4 SMSMTMessageInfo∗ SMSEventInfo_s::pMTMessageInfo

#### 8.978.2.5 SMSCAddressInfo∗ SMSEventInfo_s::pSMSCAddressInfo

#### 8.978.2.6 SMSOnIMSInfo∗ SMSEventInfo_s::pSMSOnIMSInfo

#### 8.978.2.7 SMSTransferRouteMTMessageInfo∗ SMSEventInfo_s::pTransferRouteMTMessageInfo

#### 8.978.2.8 BYTE SMSEventInfo_s::smsEventType

## 8.979 smsMaxStorageSizeReq Struct Reference

### Data Fields

- BYTE storageType
- BYTE ∗ pMessageMode

### 8.979.1 Detailed Description

This structure contains get store max size resquest 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

### 8.979.2 Field Documentation

#### 8.979.2.1 BYTE* smsMaxStorageSizeReq::pMessageMode

#### 8.979.2.2 BYTE smsMaxStorageSizeReq::storageType

## 8.980 smsMaxStorageSizeResp Struct Reference

### Data Fields

- ULONG maxStorageSize
- ULONG freeSlots

### 8.980.1 Detailed Description

This structure contains get store max size response parameters

**Parameters**

| | |
|---|---|
| *maxStorageSize* | - • Memory Store Size |
| *freeSlots* | - • Optional parameter indicating how much Memory is available • function SLQSSmsGetMaxStorageSize() returns a default value 0xFFFFFFFF for parameter values if no response is received from the device. |

### 8.980.2 Field Documentation

#### 8.980.2.1 ULONG smsMaxStorageSizeResp::freeSlots

#### 8.980.2.2 ULONG smsMaxStorageSizeResp::maxStorageSize

## 8.981 SMSMemoryInfo Struct Reference

### Data Fields

- BYTE storageType
- BYTE messageMode

### 8.981.1 Detailed Description

This structure holds information related to memory

**Parameters**

| | |
|---|---|
| *storageType* | • Indicates the type of memory storage 0x00 - STORAGE_TYPE_UIM 0x01 - STORAGE_TYPE-_NV |
| *messageMode* | • Indicates the type of memory mode 0x00 - MESSAGE_MODE_CDMA - CDMA 0x01 - MESSA-GE_MODE_GW - GW |

### 8.981.2 Field Documentation

#### 8.981.2.1 BYTE SMSMemoryInfo::messageMode

#### 8.981.2.2 BYTE SMSMemoryInfo::storageType

## 8.982 SMSMessageMode Struct Reference

### Data Fields

- BYTE messageMode

### 8.982.1 Detailed Description

This structure holds information related to message mode

**Parameters**

| | |
|---|---|
| *messageMode* | • Message mode 0x00 - CDMA 0x01 - GW |

### 8.982.2 Field Documentation

#### 8.982.2.1 BYTE SMSMessageMode::messageMode

## 8.983 sMSMessageModeInfo Struct Reference

### Data Fields

- uint8_t messageMode

### 8.983.1 Detailed Description

This structure contains message mode information.

**Parameters**

| | |
|---|---|
| *messageMode* | Message Mode |

### 8.983.2 Field Documentation

#### 8.983.2.1 uint8_t sMSMessageModeInfo::messageMode

## 8.984 smsMsgprotocolResp Struct Reference

### Data Fields

- BYTE msgProtocol

### 8.984.1 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 |

### 8.984.2 Field Documentation

#### 8.984.2.1 BYTE smsMsgprotocolResp::msgProtocol

## 8.985 SMSMTMessage Struct Reference

### Data Fields

- ULONG storageType
- ULONG messageIndex

### 8.985.1 Detailed Description

This structure holds information related to MT SMS

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type for the new message<br>0 - UIM 1 - NV |
| *messageIndex* | • Index of the new message |

### 8.985.2 Field Documentation

#### 8.985.2.1 ULONG SMSMTMessage::messageIndex

#### 8.985.2.2 ULONG SMSMTMessage::storageType

## 8.986 sMSMTMessageInfo Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageIndex

### 8.986.1 Detailed Description

This structure contains MT message information.

**Parameters**

| | |
|---|---|
| *storageType* | memory storage 0x00-UIM 0x01-NV |
| *messageIndex* | MT Message index |

### 8.986.2 Field Documentation

#### 8.986.2.1 uint32_t sMSMTMessageInfo::messageIndex

#### 8.986.2.2 uint32_t sMSMTMessageInfo::storageType

## 8.987 SMSOnIMS Struct Reference

### Data Fields

- BYTE smsOnIMS

### 8.987.1 Detailed Description

This structure holds information related to message mode

**Parameters**

| | |
|---|---|
| *smsOnIMS* | • Indicates whether the message is received from IMS 0x00 - Message is not received from IMS 0x01 - Message is received from IMS 0x02-0xFF - Reserved Note: In multiple modem solutions, this TLV may be used to help the client determine with which modem to communicate. This TLV may not be supported on all implementations. |

### 8.987.2 Field Documentation

#### 8.987.2.1 BYTE SMSOnIMS::smsOnIMS

## 8.988 sMSOnIMSInfo Struct Reference

### Data Fields

- uint8_t smsOnIMS

### 8.988.1 Detailed Description

This structure contains SMS on IMS information

**Parameters**

| | |
|---|---|
| *smsOnIMS* | SMS on IMS |

### 8.988.2 Field Documentation

#### 8.988.2.1 uint8_t sMSOnIMSInfo::smsOnIMS

## 8.989 sMSOnIMSTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSOnIMSInfo IMSInfo

### 8.989.1 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 |

### 8.989.2 Field Documentation

#### 8.989.2.1 sMSOnIMSInfo sMSOnIMSTlv::IMSInfo

#### 8.989.2.2 uint8_t sMSOnIMSTlv::TlvPresent

## 8.990 smsRouteEntry Struct Reference

### Data Fields

- BYTE messageType
- BYTE messageClass
- BYTE routeStorage
- BYTE receiptAction

### 8.990.1 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) |

### 8.990.2 Field Documentation

#### 8.990.2.1 BYTE smsRouteEntry::messageClass

#### 8.990.2.2 BYTE smsRouteEntry::messageType

#### 8.990.2.3 BYTE smsRouteEntry::receiptAction

#### 8.990.2.4 BYTE smsRouteEntry::routeStorage

## 8.991 smsSetRoutesReq Struct Reference

### Data Fields

- WORD numOfRoutes
- smsRouteEntry routeList [0x0A]
- BYTE * pTransferStatusReport

### 8.991.1 Detailed Description

This structure contains SMS route request parameters

**Parameters**

| | |
|---|---|
| *numOfRoutes* | - • Number of sets of the following element |
| *routeList* | - • Array containing the set of smsRouteEntry |
| *pTransferStatus-Report* | - • 0x01 - Status reoprt are transferred to the client (optional) |

### 8.991.2 Field Documentation

#### 8.991.2.1 WORD smsSetRoutesReq::numOfRoutes

#### 8.991.2.2 BYTE* smsSetRoutesReq::pTransferStatusReport

#### 8.991.2.3 smsRouteEntry smsSetRoutesReq::routeList[0x0A]

## 8.992 SMSTransferRouteMTMessage Struct Reference

### Data Fields

- BYTE ackIndicator
- ULONG transactionID
- BYTE format
- WORD length
- BYTE data [256]

### 8.992.1 Detailed Description

This structure holds information related to transfer route MT SMS

**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 |

### 8.992.2 Field Documentation

#### 8.992.2.1 BYTE SMSTransferRouteMTMessage::ackIndicator

#### 8.992.2.2 BYTE SMSTransferRouteMTMessage::data[256]

#### 8.992.2.3 BYTE SMSTransferRouteMTMessage::format

#### 8.992.2.4 WORD SMSTransferRouteMTMessage::length

#### 8.992.2.5 ULONG SMSTransferRouteMTMessage::transactionID

## 8.993 sMSTransferRouteMTMessageInfo Struct Reference

### Data Fields

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

### 8.993.1 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 |

### 8.993.2 Field Documentation

#### 8.993.2.1 uint8_t sMSTransferRouteMTMessageInfo::ackIndicator

#### 8.993.2.2 uint8_t sMSTransferRouteMTMessageInfo::data[256]

#### 8.993.2.3 uint8_t sMSTransferRouteMTMessageInfo::format

#### 8.993.2.4 uint16_t sMSTransferRouteMTMessageInfo::length

#### 8.993.2.5 uint32_t sMSTransferRouteMTMessageInfo::transactionID

## 8.994 sQosFlowStat Struct Reference

### Data Fields

- ULONG bearerId
- ULONG tx_pkt
- ULONG tx_pkt_drp
- ULONGLONG tx_bytes
- ULONGLONG tx_bytes_drp

### 8.994.1 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 |

### 8.994.2 Field Documentation

#### 8.994.2.1 ULONG sQosFlowStat::bearerId

#### 8.994.2.2 ULONGLONG sQosFlowStat::tx_bytes

#### 8.994.2.3 ULONGLONG sQosFlowStat::tx_bytes_drp

#### 8.994.2.4 ULONG sQosFlowStat::tx_pkt

#### 8.994.2.5 ULONG sQosFlowStat::tx_pkt_drp

## 8.995 sQosStat Struct Reference

### Data Fields

- ULONG apnId

- ULONG total_tx_pkt
- ULONG total_tx_pkt_drp
- ULONG total_rx_pkt
- ULONGLONG total_tx_bytes
- ULONGLONG total_tx_bytes_drp
- ULONGLONG total_rx_bytes
- ULONG numQosFlow
- sQosFlowStat qosFlow [(10)]

### 8.995.1 Detailed Description

This structure contains the Data statistic per QoS flow

**Parameters**

| | |
|---|---|
| *apnId* | • APN id<br>• ID identifing the connected APN that the client would like to query the data statistic for |
| *total_tx_pkt* | • sum of all packets sent |
| *total_tx_pkt_drp* | • sum of all(TX) packets dropped |
| *total_rx_pkt* | • sum of all packets received |
| *total_tx_bytes* | • sum of all bytes sent |
| *toal_tx_bytes_drp* | • sum of all(TX) bytes dropped |
| *total_rx_bytes* | • number of received bytes for the QoS flow ID |
| *numQosFlow* | • pointer to number of QoS flow Stat |
| *qosFlow[MAX_QOS_SPEC_PER_APN]* | • Data statistic per QoS flow<br>• See sQosFlowStat for more information<br>• See MAX_QOS_SPEC_PER_APN for more information |

### 8.995.2 Field Documentation

#### 8.995.2.1 ULONG sQosStat::apnId

#### 8.995.2.2 ULONG sQosStat::numQosFlow

#### 8.995.2.3 sQosFlowStat sQosStat::qosFlow[(10)]

#### 8.995.2.4 ULONGLONG sQosStat::total_rx_bytes

#### 8.995.2.5 ULONG sQosStat::total_rx_pkt

#### 8.995.2.6 ULONGLONG sQosStat::total_tx_bytes

#### 8.995.2.7 ULONGLONG sQosStat::total_tx_bytes_drp

#### 8.995.2.8 ULONG sQosStat::total_tx_pkt

#### 8.995.2.9 ULONG sQosStat::total_tx_pkt_drp

## 8.996 SrvStatusInfo Struct Reference

### Data Fields

- BYTE srvStatus
- BYTE isPrefDataPath

### 8.996.1 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 |

### 8.996.2 Field Documentation

#### 8.996.2.1 BYTE SrvStatusInfo::isPrefDataPath

#### 8.996.2.2 BYTE SrvStatusInfo::srvStatus

## 8.997 ssdatasession_params Struct Reference

### Data Fields

- BOOL action

- BYTE instanceId
- ULONG * pTechnology
- ULONG * pProfileId3GPP
- ULONG * pProfileId3GPP2
- ULONG sessionId
- ULONG failureReason
- ULONG failureReasonv4
- ULONG failureReasonv6
- ULONG rcv4
- ULONG rcv6
- ULONG v4sessionId
- ULONG v6sessionId
- BYTE ipfamily
- ULONG * pAuthentication
- CHAR * pUsername
- CHAR * pPassword
- ULONG verbFailReasonType
- ULONG verbFailReason

### 8.997.1 Detailed Description

This structure contains the start/stop data session params Information

**Parameters**

| | |
|---|---|
| *action* | • 1 - Start Session<br>• 0 - Stop Session |
| *instanceId* | • PDP Instance.<br>• Instance ID corresponding to the session ID |
| *pTechnology* | • Indicates the technology preference (optional)<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. |
| *pProfileId3GPP* | • configured 3GPP profile identifier |
| *pProfileId3GPP2* | • configured 3GPP2 profile identifier |
| *sessionId[IN\|OUT]* | • [IN] - Passed session ID when stopping the data session<br>• [OUT] - Assigned session ID when starting a data session |
| *failureReason* | • Reason data session failed to be established<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason |

| | |
|---|---|
| *failureReasonv4* | • Reason v4 data session failed to be established<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason |
| *failureReasonv6* | • Reason v6 data session failed to be established<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason |
| *rc4* | • v4 result code<br>• See qmerrno.h |
| *rc6* | • v6 result code<br>• See qmerrno.h |
| *v4sessionId* | • Do not modify - used for internal management of data sessions<br>• Non zero value indicates that a session is active |
| *v6sessionId* | • Do not modify - used for internal management of data sessions<br>• Non zero value indicates that a session is active |
| *ipfamily* | • 4 for an IPv4 data session<br>• 6 for an IPv6 data session<br>• 7 for an IPv4v6 data session |
| *pAuthentication* | • Authentication type, it can be PAP or CHAP |
| *pUsername* | • username for authentication process |
| *pPassword* | • password for authentication process |
| *verbFailReason-Type* | • Parameter describing type of verbose failure reason<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason Type |
| *verbFailReason* | • Verbose reason explaining why call failed. Depends on verbFailReasonType parameter<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason |

### 8.997.2 Field Documentation

#### 8.997.2.1 BOOL ssdatasession_params::action

#### 8.997.2.2 ULONG ssdatasession_params::failureReason

#### 8.997.2.3 ULONG ssdatasession_params::failureReasonv4

#### 8.997.2.4 ULONG ssdatasession_params::failureReasonv6

#### 8.997.2.5 BYTE ssdatasession_params::instanceId

#### 8.997.2.6 BYTE ssdatasession_params::ipfamily

#### 8.997.2.7 ULONG* ssdatasession_params::pAuthentication

#### 8.997.2.8 CHAR* ssdatasession_params::pPassword

#### 8.997.2.9 ULONG* ssdatasession_params::pProfileId3GPP

#### 8.997.2.10 ULONG* ssdatasession_params::pProfileId3GPP2

#### 8.997.2.11 ULONG* ssdatasession_params::pTechnology

#### 8.997.2.12 CHAR* ssdatasession_params::pUsername

#### 8.997.2.13 ULONG ssdatasession_params::rcv4

#### 8.997.2.14 ULONG ssdatasession_params::rcv6

#### 8.997.2.15 ULONG ssdatasession_params::sessionId

#### 8.997.2.16 ULONG ssdatasession_params::v4sessionId

#### 8.997.2.17 ULONG ssdatasession_params::v6sessionId

#### 8.997.2.18 ULONG ssdatasession_params::verbFailReason

#### 8.997.2.19 ULONG ssdatasession_params::verbFailReasonType

## 8.998 SupportedMsgList Struct Reference

### Data Fields

- WORD supportedMsgLen
- BYTE supportedMsgs [256]

### 8.998.1 Detailed Description

This structure contains the Supported Messages List Information

**Parameters**

| | |
|---|---|
| *supportedMsg-Len* | • Number of sets of the supported messages |
| *supportedMsgs* | • Array of uint8 is a bitmask where each bit represents a message ID.<br>• Starting with the LSB, bit 0 represents message ID 0, bit 1 represents message ID 1. |

### 8.998.2 Field Documentation

#### 8.998.2.1 WORD SupportedMsgList::supportedMsgLen

#### 8.998.2.2 BYTE SupportedMsgList::supportedMsgs[256]

## 8.999 SUPSInfo Struct Reference

### Data Fields

- BYTE svcType
- BYTE isModByCC

### 8.999.1 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 |

### 8.999.2 Field Documentation

#### 8.999.2.1 BYTE SUPSInfo::isModByCC

#### 8.999.2.2 BYTE SUPSInfo::svcType

## 8.1000 SV Struct Reference

### Data Fields

- WORD id
- ULONG system
- BYTE mask

### 8.1000.1 Detailed Description

This structure contains the Delete SV Info

**Parameters**

| | |
|---|---|
| *id* | • 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 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 |

### 8.1000.2 Field Documentation

#### 8.1000.2.1 WORD SV::id

#### 8.1000.2.2 BYTE SV::mask

#### 8.1000.2.3 ULONG SV::system

## 8.1001 SVInfo Struct Reference

### Data Fields

- BYTE len
- SV * pSV

### 8.1001.1 Detailed Description

This structure contains the elements of Delete SV Info

**Parameters**

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

### 8.1001.2 Field Documentation

#### 8.1001.2.1 BYTE SVInfo::len

#### 8.1001.2.2 SV* SVInfo::pSV

## 8.1002 svUsedforFix_s Struct Reference

### Data Fields

- BYTE gnssSvUsedList_len
- WORD gnssSvUsedList [255]

### 8.1002.1 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 |

### 8.1002.2 Field Documentation

#### 8.1002.2.1 WORD svUsedforFix_s::gnssSvUsedList[255]

#### 8.1002.2.2 BYTE svUsedforFix_s::gnssSvUsedList_len

## 8.1003 SWI_STRUCT_CarrierImage Struct Reference

### Data Fields

- ULONG m_nCarrierId
- ULONG m_nFolderId
- ULONG m_nStorage
- BYTE m_FwImageId [16]
- BYTE m_FwBuildId [100]
- BYTE m_PriImageId [16]
- BYTE m_PriBuildId [100]

## 8.1003.1 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 |

## 8.1003.2 Field Documentation

### 8.1003.2.1 BYTE SWI_STRUCT_CarrierImage::m_FwBuildId[100]

### 8.1003.2.2 BYTE SWI_STRUCT_CarrierImage::m_FwImageId[16]

### 8.1003.2.3 ULONG SWI_STRUCT_CarrierImage::m_nCarrierId

### 8.1003.2.4 ULONG SWI_STRUCT_CarrierImage::m_nFolderId

### 8.1003.2.5 ULONG SWI_STRUCT_CarrierImage::m_nStorage

#### 8.1003.2.6 BYTE SWI_STRUCT_CarrierImage::m_PriBuildId[100]

#### 8.1003.2.7 BYTE SWI_STRUCT_CarrierImage::m_PriImageId[16]

## 8.1004 swidms_ehrpdMTUSizeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t ehrpdMTUSize

### 8.1004.1 Detailed Description

This structure contains the parameters for EHRPD MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *ehrpdMTUSize* | EHRPD size<br>• Size of EHRPD MTU |

### 8.1004.2 Field Documentation

#### 8.1004.2.1 uint16_t swidms_ehrpdMTUSizeTlv::ehrpdMTUSize

#### 8.1004.2.2 uint8_t swidms_ehrpdMTUSizeTlv::TlvPresent

## 8.1005 swidms_hrpdMTUSizeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t hrpdMTUSize

### 8.1005.1 Detailed Description

This structure contains the parameters for HRPD MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *hrpdMTUSize* | HRPD size<br>• Size of HRPD MTU |

### 8.1005.2 Field Documentation

#### 8.1005.2.1 uint16_t swidms_hrpdMTUSizeTlv::hrpdMTUSize

#### 8.1005.2.2 uint8_t swidms_hrpdMTUSizeTlv::TlvPresent

## 8.1006 swidms_mtuSize3gppTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t MTUSize3gpp

### 8.1006.1 Detailed Description

This structure contains the parameters for 3GPP MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *MTUSize3gpp* | 3GPP MTU size<br>• Size of 3GPP MTU |

### 8.1006.2 Field Documentation

#### 8.1006.2.1 uint16_t swidms_mtuSize3gppTlv::MTUSize3gpp

#### 8.1006.2.2 uint8_t swidms_mtuSize3gppTlv::TlvPresent

## 8.1007 swidms_usbMTUSizeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t UsbMTUSize

### 8.1007.1 Detailed Description

This structure contains the parameters for USB MTU Size

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *UsbMTUSize* | USB MTU size<br>• Size of USB MTU |

### 8.1007.2 Field Documentation

#### 8.1007.2.1 uint8_t swidms_usbMTUSizeTlv::TlvPresent

#### 8.1007.2.2 uint16_t swidms_usbMTUSizeTlv::UsbMTUSize

## 8.1008 SwiDmsGetMTUResp Struct Reference

### Data Fields

- mtuSize3gppTlv * pMTUSize3gpp
- hrpdMTUSizeTlv * pHrpdMTUSize
- ehrpdMTUSizeTlv * pEhrpdMTUSize
- usbMTUSizeTlv * pUsbMTUSize

### 8.1008.1 Detailed Description

This structure contains the Get MTU Response parameter.

**Parameters**

| | |
|---|---|
| *pMTUSize3gpp* | [Optional]<br>• See mtuSize3gppTlv for more information |
| *pHrpdMTUSize* | [Optional]<br>• See hrpdMTUSizeTlv for more information |
| *pEhrpdMTUSize* | [Optional]<br>• See ehrpdMTUSizeTlv for more information |
| *pUsbMTUSize* | [Optional]<br>• See usbMTUSizeTlv for more information |

### 8.1008.2 Field Documentation

#### 8.1008.2.1 ehrpdMTUSizeTlv* SwiDmsGetMTUResp::pEhrpdMTUSize

#### 8.1008.2.2 hrpdMTUSizeTlv* SwiDmsGetMTUResp::pHrpdMTUSize

#### 8.1008.2.3 mtuSize3gppTlv* SwiDmsGetMTUResp::pMTUSize3gpp

#### 8.1008.2.4 usbMTUSizeTlv* SwiDmsGetMTUResp::pUsbMTUSize

## 8.1009 SwiDmsSLQSGetUsbNetNum Struct Reference

### Data Fields

- BYTE usbNetNum

### 8.1009.1 Detailed Description

This structure contains QMAP configuration

**Parameters**

| | |
|---|---|
| *usbNetNum* | • Number of usb net<br>• Range is 0-8 |

### 8.1009.2 Field Documentation

#### 8.1009.2.1 BYTE SwiDmsSLQSGetUsbNetNum::usbNetNum

## 8.1010 SwiLocGetAutoStartResp Struct Reference

### Data Fields

- BYTE function
- BOOL function_reported
- BYTE fix_type
- BOOL fix_type_reported
- BYTE max_time
- BOOL max_time_reported
- ULONG max_dist
- BOOL max_dist_reported
- ULONG fix_rate
- BOOL fix_rate_reported

### 8.1010.1 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 |
| *function_reported* | • 0 - not reported by modem<br>• 1 - 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 |
| *fix_type_reported* | • 0 - not reported by modem<br>• 1 - reported by modem |

| | |
|---|---|
| *max_time* | • Maximum time allowed for the receiver to get a fix in seconds<br>• Valid range: 1-255 |
| *max_time_reported* | • 0 - not reported by modem<br>• 1 - reported by modem |
| *max_dist* | • Maximum uncertainty of a fix measured by distance in meters<br>• Valid range: 1 - 4294967280 |
| *max_dist_reported* | • 0 - not reported by modem<br>• 1 - reported by modem |
| *fix_rate* | • Time between fixes in seconds<br>• Valid range: 1–65535 |
| *fix_rate_reported* | • 0 - not reported by modem<br>• 1 - reported by modem |

### 8.1010.2 Field Documentation

#### 8.1010.2.1 ULONG SwiLocGetAutoStartResp::fix_rate

#### 8.1010.2.2 BOOL SwiLocGetAutoStartResp::fix_rate_reported

#### 8.1010.2.3 BYTE SwiLocGetAutoStartResp::fix_type

#### 8.1010.2.4 BOOL SwiLocGetAutoStartResp::fix_type_reported

#### 8.1010.2.5 BYTE SwiLocGetAutoStartResp::function

#### 8.1010.2.6 BOOL SwiLocGetAutoStartResp::function_reported

#### 8.1010.2.7 ULONG SwiLocGetAutoStartResp::max_dist

#### 8.1010.2.8 BOOL SwiLocGetAutoStartResp::max_dist_reported

#### 8.1010.2.9 BYTE SwiLocGetAutoStartResp::max_time

#### 8.1010.2.10 BOOL SwiLocGetAutoStartResp::max_time_reported

## 8.1011 SwiLocSetAutoStartReq Struct Reference

### Data Fields

- BYTE function
- BOOL set_function

- BYTE fix_type
- BOOL set_fix_type
- BYTE max_time
- BOOL set_max_time
- ULONG max_dist
- BOOL set_max_dist
- ULONG fix_rate
- BOOL set_fix_rate

### 8.1011.1 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>• 1 - 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>• 1 - 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>• 1 - 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>• 1 - 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>• 1 - set to modem |

### 8.1011.2 Field Documentation

#### 8.1011.2.1 ULONG SwiLocSetAutoStartReq::fix_rate

#### 8.1011.2.2 BYTE SwiLocSetAutoStartReq::fix_type

#### 8.1011.2.3 BYTE SwiLocSetAutoStartReq::function

#### 8.1011.2.4 ULONG SwiLocSetAutoStartReq::max_dist

#### 8.1011.2.5 BYTE SwiLocSetAutoStartReq::max_time

#### 8.1011.2.6 BOOL SwiLocSetAutoStartReq::set_fix_rate

#### 8.1011.2.7 BOOL SwiLocSetAutoStartReq::set_fix_type

#### 8.1011.2.8 BOOL SwiLocSetAutoStartReq::set_function

#### 8.1011.2.9 BOOL SwiLocSetAutoStartReq::set_max_dist

#### 8.1011.2.10 BOOL SwiLocSetAutoStartReq::set_max_time

## 8.1012 swiModemStatusResp Struct Reference

### Data Fields

- CommInfo commonInfo
- LTEInfo * pLTEInfo

### 8.1012.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

**Parameters**

| | |
|---|---|
| *commonInfo* | (mandatory)<br>• See CommInfo for more information |
| *pLTEInfo* | (optional)<br>• See LTEInfo for more information |

### 8.1012.2 Field Documentation

#### 8.1012.2.1 CommInfo swiModemStatusResp::commonInfo

#### 8.1012.2.2 LTEInfo* swiModemStatusResp::pLTEInfo

## 8.1013 SwiOTAMsg_s Struct Reference

### Data Fields

- ULONG type
- WORD data_len
- BYTE data [2048]
- LteNasReleaseInfo * pLteNasRelInfo
- ULONGLONG * pTime

### 8.1013.1 Detailed Description

This structure contains OTA message

**Parameters**

| | |
|---|---|
| *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* | • OTA Message Content |
| *pLteNasRelInfo* | • LTE NAS Release Info<br>• see LteNasReleaseInfo for details |
| *pTime* | • Seconds in local time since Jan. 6th 1980 00:00:00 UTC |

### 8.1013.2 Field Documentation

#### 8.1013.2.1 BYTE SwiOTAMsg_s::data[2048]

#### 8.1013.2.2 WORD SwiOTAMsg_s::data_len

#### 8.1013.2.3 LteNasReleaseInfo* SwiOTAMsg_s::pLteNasRelInfo

#### 8.1013.2.4 ULONGLONG* SwiOTAMsg_s::pTime

#### 8.1013.2.5 ULONG SwiOTAMsg_s::type

## 8.1014 swiPDPRuntimeSettingsReq Struct Reference

### Data Fields

- BYTE contextId
- BYTE contextType

### 8.1014.1 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 |

### 8.1014.2 Field Documentation

#### 8.1014.2.1 BYTE swiPDPRuntimeSettingsReq::contextId

#### 8.1014.2.2 BYTE swiPDPRuntimeSettingsReq::contextType

## 8.1015 swiPDPRuntimeSettingsResp Struct Reference

### Data Fields

- BYTE * pContextId
- BYTE * pBearerId
- CHAR * pAPNName
- ULONG * pIPv4Address
- ULONG * pIPv4GWAddress
- ULONG * pPrDNSIPv4Address
- ULONG * pSeDNSIPv4Address
- struct IPV6AddressInfo * pIPv6Address
- struct IPV6AddressInfo * pIPv6GWAddress
- WORD * pPrDNSIPv6Address
- WORD * pSeDNSIPv6Address
- ULONG * pPrPCSCFIPv4Address
- ULONG * pSePCSCFIPv4Address
- WORD * pPrPCSCFIPv6Address
- WORD * pSePCSCFIPv6Address

### 8.1015.1 Detailed Description

This structure contains the response parameters retrieved by the API SLQSWdsSwiPDPRuntimeSettings

**Parameters**

| | |
|---|---|
| *pContextId* | (optional)<br>• Context Identifier<br>– 0xFF - Not Available |
| *pBearerId* | (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 |
| *pAPNName* | (optional)<br>• APN name associated with the context id<br>– NULL terminated by default. |
| *pIPv4Address* | (optional)<br>• IPv4 Address<br>– 0xFFFF - Not Available |
| *pIPv4GW-Address* | (optional)<br>• IPv4 Gateway Address<br>– 0xFFFF - Not Available |
| *pPrDNSIPv4-Address* | (optional)<br>• Primary DNS IPv4 Address<br>– 0xFFFF - Not Available |
| *pSeDNSIPv4-Address* | (optional)<br>• Secondary DNS IPv4 Address<br>– 0xFFFF - Not Available |
| *pIPv6Address* | (optional)<br>• IPv6 Address<br>• See IPV6AddressInfo for more information |
| *pIPv6GW-Address* | (optional)<br>• IPv6 Gateway Address<br>• See IPV6AddressInfo for more information |
| *pPrDNSIPv6-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 |
| *pSeDNSIPv6-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 |
| *pPrPCSCFIPv4-Address* | (optional)<br>• Primary PCSCF IPv4 Address |

| | |
|---|---|
| *pSePCSCFIPv4-Address* | (optional)<br>• Secondary PCSCF IPv4 Address |
| *pPrPCSCFIPv6-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 |
| *pSePCSCFIPv6-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 |

**Note**

Parameters which are mentioned as NULL will be ignored.

### 8.1015.2 Field Documentation

#### 8.1015.2.1 CHAR* swiPDPRuntimeSettingsResp::pAPNName

#### 8.1015.2.2 BYTE* swiPDPRuntimeSettingsResp::pBearerId

#### 8.1015.2.3 BYTE* swiPDPRuntimeSettingsResp::pContextId

#### 8.1015.2.4 ULONG* swiPDPRuntimeSettingsResp::pIPv4Address

#### 8.1015.2.5 ULONG* swiPDPRuntimeSettingsResp::pIPv4GWAddress

#### 8.1015.2.6 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6Address

#### 8.1015.2.7 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6GWAddress

#### 8.1015.2.8 ULONG* swiPDPRuntimeSettingsResp::pPrDNSIPv4Address

#### 8.1015.2.9 WORD* swiPDPRuntimeSettingsResp::pPrDNSIPv6Address

#### 8.1015.2.10 ULONG* swiPDPRuntimeSettingsResp::pPrPCSCFIPv4Address

#### 8.1015.2.11 WORD* swiPDPRuntimeSettingsResp::pPrPCSCFIPv6Address

#### 8.1015.2.12 ULONG* swiPDPRuntimeSettingsResp::pSeDNSIPv4Address

#### 8.1015.2.13 WORD* swiPDPRuntimeSettingsResp::pSeDNSIPv6Address

#### 8.1015.2.14 ULONG* swiPDPRuntimeSettingsResp::pSePCSCFIPv4Address

#### 8.1015.2.15 WORD* swiPDPRuntimeSettingsResp::pSePCSCFIPv6Address

## 8.1016 swiQosFilter Struct Reference

### Data Fields

- BYTE index

- BYTE version
- IPv4Addr * pIPv4SrcAddr
- IPv4Addr * pIPv4DstAddr
- BYTE * pNxtHdrProto
- Tos * pTos
- IPv6Addr * pIPv6SrcAddr
- IPv6Addr * pIPv6DstAddr
- IPv6TrafCls * pIPv6TrafCls
- ULONG * pIPv6Label
- Port * pTCPSrcPort
- Port * pTCPDstPort
- Port * pUDPSrcPort
- Port * pUDPDstPort
- ULONG * pEspSpi
- WORD * pPrecedence
- WORD * pId
- Port * pTranSrcPort
- Port * pTranDstPort

### 8.1016.1 Detailed Description

This structure contains the QoS Filter Request

**Parameters**

| | |
|---|---|
| *index* | IP filter index Integer that uniquely identifies each filter instance This TLV must be present in the request |
| *version* | 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 |
| *pIPv4SrcAddr* | IPv4 filter soruce address See IPv4Addr for more information<br>• Implemented only for unsolicited indication |
| *pIPv4DstAddr* | IPv4 filter destination address See IPv4Addr for more information<br>• Implemented only for unsolicited indication |
| *pNxtHdrProto* | 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. |
| *pTos* | IPv4 filter type of service See Tos for more information |
| *pIPv6SrcAddr* | IPv6 filter soruce address See IPv6Addr for more information<br>• Implemented only for unsolicited indication |
| *pIPv6DstAddr* | IPv6 filter destination address See IPv6Addr for more information<br>• Implemented only for unsolicited indication |

| | |
|---|---|
| *pIPv6TrafCls* | IPv6 filter traffic class See IPv6TrafCls for more information |
| *pIPv6Label* | IPv6 flow label Packet matches the IPv6 flow label filter if: ( ∗pIPv6Label == flow label in the IPv6 header)<br>• Implemented only for unsolicited indication |
| *pTCPSrcPort* | TCP filter source port filter See Port for more information<br>• Implemented only for unsolicited indication |
| *pTCPDstPort* | TCP filter destination port filter See Port for more information<br>• Implemented only for unsolicited indication |
| *pUDPSrcPort* | UDP filter source port filter See Port for more information<br>• Implemented only for unsolicited indication |
| *pUDPDstPort* | UDP filter destination port filter See Port for more information<br>• Implemented only for unsolicited indication |
| *pEspSpi* | 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 |
| *pPrecedence* | 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 |
| *pId* | 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 |
| *pTranSrcPort* | Transport protocolfilter source port See Port for more information<br>• Implemented only for unsolicited indication |
| *pUDPDstPort* | Transport protocol filter destination port See Port for more information<br>• Implemented only for unsolicited indication |

## 8.1016.2 Field Documentation

### 8.1016.2.1 BYTE swiQosFilter::index

### 8.1016.2.2 ULONG∗ swiQosFilter::pEspSpi

### 8.1016.2.3 WORD∗ swiQosFilter::pId

### 8.1016.2.4 IPv4Addr∗ swiQosFilter::pIPv4DstAddr

### 8.1016.2.5 IPv4Addr∗ swiQosFilter::pIPv4SrcAddr

### 8.1016.2.6 IPv6Addr∗ swiQosFilter::pIPv6DstAddr

### 8.1016.2.7 ULONG∗ swiQosFilter::pIPv6Label

### 8.1016.2.8 IPv6Addr∗ swiQosFilter::pIPv6SrcAddr

### 8.1016.2.9 IPv6TrafCls∗ swiQosFilter::pIPv6TrafCls

#### 8.1016.2.10 BYTE* swiQosFilter::pNxtHdrProto

#### 8.1016.2.11 WORD* swiQosFilter::pPrecedence

#### 8.1016.2.12 Port* swiQosFilter::pTCPDstPort

#### 8.1016.2.13 Port* swiQosFilter::pTCPSrcPort

#### 8.1016.2.14 Tos* swiQosFilter::pTos

#### 8.1016.2.15 Port* swiQosFilter::pTranDstPort

#### 8.1016.2.16 Port* swiQosFilter::pTranSrcPort

#### 8.1016.2.17 Port* swiQosFilter::pUDPDstPort

#### 8.1016.2.18 Port* swiQosFilter::pUDPSrcPort

#### 8.1016.2.19 BYTE swiQosFilter::version

## 8.1017 swiQosFlow Struct Reference

### Data Fields

- BYTE index
- WORD * pProfileId3GPP2
- BYTE * p3GPP2Pri
- BYTE * pTrafficClass
- dataRate * pDataRate
- tokenBucket * pTokenBucket
- ULONG * pLatency
- ULONG * pJitter
- pktErrRate * pPktErrRate
- ULONG * pMinPolicedPktSz
- ULONG * pMaxAllowedPktSz
- WORD * p3GPPResResidualBER
- BYTE * p3GPPTraHdlPri
- BYTE * p3GPPImCn
- BYTE * p3GPPSigInd
- BYTE * pLteQci

### 8.1017.1 Detailed Description

This structure contains the QoS Flow Request

**Parameters**

| | |
|---|---|
| *index* | • 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 |

| | |
|---|---|
| *pProfileId3GPP2* | • 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 |
| *p3GPP2Pri* | • 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 |
| *pTrafficClass* | • IP flow traffic class<br>• Integer that designates the requested traffic class:<br>• 0 – Conversational<br>• 1 – Streaming<br>• 2 – Interactive<br>• 3 – Background |
| *pDataRate* | • IP flow data rate min max<br>• See dataRate for more information |
| *pTokenBucket* | • IP flow data rate token bucket<br>• See tokenBucket for more information |
| *pLatency* | • IP flow latency<br>• Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link |
| *pJitter* | • 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 |
| *pPktErrRate* | • IP flow packet error rate<br>• See pktErrRate for more information |
| *pMinPolicedPkt-Sz* | • 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 |
| *pMaxAllowed-PktSz* | • 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 |

| | |
|---|---|
| *p3GPPRes-ResidualBER* | • 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 |
| *p3GPPTraHdlPri* | • 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 |
| *p3GPPImCn* | • 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 |
| *p3GPPSigInd* | • IP flow 3GPP signaling indication<br>• 0x00 – FALSE<br>• 0x01 – TRUE<br>• This parameter applies only to 3GPP networks |
| *pLteQci* | • 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 |

### 8.1017.2 Field Documentation

#### 8.1017.2.1 BYTE swiQosFlow::index

#### 8.1017.2.2 BYTE∗ swiQosFlow::p3GPP2Pri

#### 8.1017.2.3 BYTE* swiQosFlow::p3GPPImCn

#### 8.1017.2.4 WORD* swiQosFlow::p3GPPResResidualBER

#### 8.1017.2.5 BYTE* swiQosFlow::p3GPPSigInd

#### 8.1017.2.6 BYTE* swiQosFlow::p3GPPTraHdlPri

#### 8.1017.2.7 dataRate* swiQosFlow::pDataRate

#### 8.1017.2.8 ULONG* swiQosFlow::pJitter

#### 8.1017.2.9 ULONG* swiQosFlow::pLatency

#### 8.1017.2.10 BYTE* swiQosFlow::pLteQci

#### 8.1017.2.11 ULONG* swiQosFlow::pMaxAllowedPktSz

#### 8.1017.2.12 ULONG* swiQosFlow::pMinPolicedPktSz

#### 8.1017.2.13 pktErrRate* swiQosFlow::pPktErrRate

#### 8.1017.2.14 WORD* swiQosFlow::pProfileId3GPP2

#### 8.1017.2.15 tokenBucket* swiQosFlow::pTokenBucket

#### 8.1017.2.16 BYTE* swiQosFlow::pTrafficClass

## 8.1018 swiQosGranted Struct Reference

### Data Fields

- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow

### 8.1018.1 Detailed Description

This structure contains the QoS granted flow

**Parameters**

| | |
|---|---|
| *pTxFlow* | See swiQosFlow for more information |
| *pRxFlow* | See swiQosFlow for more information |

### 8.1018.2 Field Documentation

#### 8.1018.2.1 swiQosFlow* swiQosGranted::pRxFlow

#### 8.1018.2.2 swiQosFlow* swiQosGranted::pTxFlow

## 8.1019 swiQosIds Struct Reference

### Data Fields

- BYTE sz
- ULONG * plds

#### 8.1019.1 Detailed Description

This structure contains the QoS Response parameters.

**Parameters**

| | |
|---|---|
| *sz* | Number of QoS identifiers |
| *plds* | Identifier for the QoS flow requested; number of QoS identifiers present will be equal to number of QoS specs requested in the QoS Request message |

#### 8.1019.2 Field Documentation

##### 8.1019.2.1 ULONG* swiQosIds::plds

##### 8.1019.2.2 BYTE swiQosIds::sz

## 8.1020 swiQosModifyReq Struct Reference

### Data Fields

- ULONG id
- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow
- swiQosFilter * pTxFilter
- swiQosFilter * pRxFilter

#### 8.1020.1 Detailed Description

This structure contains the QoS Request parameters.

**Parameters**

| | |
|---|---|
| *id* | Identifier for the QoS flow/instance that has been negotiated and that needs to be modified The QoS_identifier is used to reference the actual flow/filter specifications that are in effect as a result of the negotiation triggered by QMI_QOS_REQUEST_QOS_REQ |
| *pTxFlow* | See swiQosFlow for more information |
| *pRxFlow* | See swiQosFlow for more information |
| *pTxFilter* | See swiQosFilter for more information |
| *pRxFilter* | See swiQosFilter for more information |

#### 8.1020.2 Field Documentation

##### 8.1020.2.1 ULONG swiQosModifyReq::id

##### 8.1020.2.2 swiQosFilter* swiQosModifyReq::pRxFilter

##### 8.1020.2.3 swiQosFlow* swiQosModifyReq::pRxFlow

#### 8.1020.2.4 swiQosFilter* swiQosModifyReq::pTxFilter

#### 8.1020.2.5 swiQosFlow* swiQosModifyReq::pTxFlow

## 8.1021 swiQosReq Struct Reference

### Data Fields

- BYTE index
- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow
- swiQosFilter * pTxFilter
- swiQosFilter * pRxFilter

### 8.1021.1 Detailed Description

This structure contains the QoS Request parameters.

**Parameters**

| | |
|---|---|
| *index* | • An integer that uniquely identifies each QoS spec included in the QoS request message |
| *pTxFlow* | • See swiQosFlow for more information |
| *pRxFlow* | • See swiQosFlow for more information |
| *pTxFilter* | • See swiQosFilter for more information |
| *pRxFilter* | • See swiQosFilter for more information |

### 8.1021.2 Field Documentation

#### 8.1021.2.1 BYTE swiQosReq::index

#### 8.1021.2.2 swiQosFilter* swiQosReq::pRxFilter

#### 8.1021.2.3 swiQosFlow* swiQosReq::pRxFlow

#### 8.1021.2.4 swiQosFilter* swiQosReq::pTxFilter

#### 8.1021.2.5 swiQosFlow* swiQosReq::pTxFlow

## 8.1022 swiRMTrasnferStaticsReq Struct Reference

### Data Fields

- BYTE bResetStatistics

- ULONG ulMask

### 8.1022.1 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 |

### 8.1022.2 Field Documentation

#### 8.1022.2.1 BYTE swiRMTrasnferStaticsReq::bResetStatistics

#### 8.1022.2.2 ULONG swiRMTrasnferStaticsReq::ulMask

## 8.1023 sysInfoCommon Struct Reference

### Data Fields

- BYTE srvDomainValid
- BYTE srvDomain
- BYTE srvCapabilityValid
- BYTE srvCapability
- BYTE roamStatusValid
- BYTE roamStatus
- BYTE isSysForbiddenValid
- BYTE isSysForbidden

### 8.1023.1 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 |

### 8.1023.2 Field Documentation

#### 8.1023.2.1 BYTE sysInfoCommon::isSysForbidden

#### 8.1023.2.2 BYTE sysInfoCommon::isSysForbiddenValid

#### 8.1023.2.3 BYTE sysInfoCommon::roamStatus

#### 8.1023.2.4 BYTE sysInfoCommon::roamStatusValid

#### 8.1023.2.5 BYTE sysInfoCommon::srvCapability

#### 8.1023.2.6 BYTE sysInfoCommon::srvCapabilityValid

#### 8.1023.2.7 BYTE sysInfoCommon::srvDomain

#### 8.1023.2.8 BYTE sysInfoCommon::srvDomainValid

## 8.1024 sysSelectPrefParamsExt Struct Reference

### Data Fields

- sysSelectPrefParams * pSysSelectPrefParams
- ULONGLONG * pCiotLteOpModePref
- ULONGLONG * pLteM1BandPref
- ULONGLONG * pLteNb1BandPref
- ULONG * pCiotLteOpMode
- struct ciotAcqOrderPref * pCiotAcqOrderPref
- struct nr5gBandPref * pNr5gBandPref

### 8.1024.1 Detailed Description

Contain the system selection preferences.

**Parameters**

| | |
|---|---|
| *pSysSelectPref-Params[IN]* | • See sysSelectPrefParams 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 ciotAcqOrderPref for more information |
| *pNr5gBandPref[-IN]* | • NR5G Band Preference<br>• see nr5gBandPref for more information |

### 8.1024.2 Field Documentation

#### 8.1024.2.1 struct ciotAcqOrderPref* sysSelectPrefParamsExt::pCiotAcqOrderPref

#### 8.1024.2.2 ULONG* sysSelectPrefParamsExt::pCiotLteOpMode

#### 8.1024.2.3 ULONGLONG* sysSelectPrefParamsExt::pCiotLteOpModePref

#### 8.1024.2.4 ULONGLONG* sysSelectPrefParamsExt::pLteM1BandPref

#### 8.1024.2.5 ULONGLONG* sysSelectPrefParamsExt::pLteNb1BandPref

#### 8.1024.2.6 struct nr5gBandPref* sysSelectPrefParamsExt::pNr5gBandPref

#### 8.1024.2.7 sysSelectPrefParams* sysSelectPrefParamsExt::pSysSelectPrefParams

## 8.1025 t_gpsTime Struct Reference

### Data Fields

- USHORT gpsWeek
- ULONG gpsTimeOfWeekMs

### 8.1025.1 Field Documentation

#### 8.1025.1.1 ULONG t_gpsTime::gpsTimeOfWeekMs

#### 8.1025.1.2 USHORT t_gpsTime::gpsWeek

## 8.1026 t_sensor Struct Reference

### Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

### 8.1026.1 Field Documentation

#### 8.1026.1.1 ULONG t_sensor::aidingIndicatorMask

#### 8.1026.1.2 ULONG t_sensor::usageMask

## 8.1027 t_Sv Struct Reference

### Data Fields

- BYTE len
- USHORT entries [255]

### 8.1027.1 Field Documentation

#### 8.1027.1.1 USHORT t_Sv::entries[255]

#### 8.1027.1.2 BYTE t_Sv::len

## 8.1028 TDSCDMAECIOThresh Struct Reference

### Data Fields

- BYTE TDSCDMAECIOThreshListLen
- ULONG * pTDSCDMAECIOThreshList

### 8.1028.1 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. |

### 8.1028.2 Field Documentation

#### 8.1028.2.1 ULONG* TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

#### 8.1028.2.2 BYTE TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

## 8.1029 TDSCDMARSCPThresh Struct Reference

### Data Fields

- BYTE TDSCDMARSCPThreshListLen
- WORD * pTDSCDMARSCPThreshList

### 8.1029.1 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). |

### 8.1029.2 Field Documentation

#### 8.1029.2.1 WORD* TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

#### 8.1029.2.2 BYTE TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

## 8.1030 TDSCDMARSSIThresh Struct Reference

### Data Fields

- BYTE TDSCDMARSSIThreshListLen
- ULONG * pTDSCDMARSSIThreshList

### 8.1030.1 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. |

### 8.1030.2 Field Documentation

#### 8.1030.2.1 ULONG* TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

#### 8.1030.2.2 BYTE TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

## 8.1031 TDSCDMASigInfoExt Struct Reference

### Data Fields

- FLOAT rssi
- FLOAT rscp
- FLOAT ecio
- FLOAT sinr

### 8.1031.1 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 |

### 8.1031.2 Field Documentation

#### 8.1031.2.1 FLOAT TDSCDMASigInfoExt::ecio

#### 8.1031.2.2 FLOAT TDSCDMASigInfoExt::rscp

#### 8.1031.2.3 FLOAT TDSCDMASigInfoExt::rssi

#### 8.1031.2.4 FLOAT TDSCDMASigInfoExt::sinr

## 8.1032 tdscdmaSigInfoExt Struct Reference

### Data Fields

- float rssi
- float rscp
- float ecio
- float sinr

### 8.1032.1 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 |

### 8.1032.2 Field Documentation

#### 8.1032.2.1 float tdscdmaSigInfoExt::ecio

#### 8.1032.2.2 float tdscdmaSigInfoExt::rscp

#### 8.1032.2.3 float tdscdmaSigInfoExt::rssi

#### 8.1032.2.4 float tdscdmaSigInfoExt::sinr

## 8.1033 TDSCDMASINRCONFThresh Struct Reference

### Data Fields

- BYTE TDSCDMASINRCONFThreshListLen
- FLOAT * pTDSCDMASINRCONFThreshList

### 8.1033.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

**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 |

### 8.1033.2 Field Documentation

#### 8.1033.2.1 FLOAT* TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

#### 8.1033.2.2 BYTE TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

## 8.1034 TDSCDMASINRThresh Struct Reference

### Data Fields

- BYTE TDSCDMASINRThreshListLen
- ULONG * pTDSCDMASINRThreshList

### 8.1034.1 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 |

### 8.1034.2 Field Documentation

#### 8.1034.2.1 ULONG* TDSCDMASINRThresh::pTDSCDMASINRThreshList

#### 8.1034.2.2 BYTE TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.1035 tempData_t Struct Reference

### Data Fields

- uint32_t timeSource
- uint32_t timeOfFirstSample
- uint8_t temperatureDataLen
- uint16_t timeOffset [64]
- uint32_t temperature [64]

### 8.1035.1 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 |

### 8.1035.2 Field Documentation

#### 8.1035.2.1 uint32_t tempData_t::temperature[64]

#### 8.1035.2.2 uint8_t tempData_t::temperatureDataLen

#### 8.1035.2.3 uint32_t tempData_t::timeOfFirstSample

#### 8.1035.2.4 uint16_t tempData_t::timeOffset[64]

#### 8.1035.2.5 uint32_t tempData_t::timeSource

## 8.1036 tempratureData Struct Reference

### Data Fields

- ULONG timeSource
- ULONG timeOfFirstSample
- BYTE temperatureDataLen
- WORD timeOffset [64]
- ULONG temperature [64]

### 8.1036.1 Detailed Description

This structure specifies information regarding the Temperature Data.

**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 |

### 8.1036.2 Field Documentation

#### 8.1036.2.1 ULONG tempratureData::temperature[64]

#### 8.1036.2.2 BYTE tempratureData::temperatureDataLen

#### 8.1036.2.3 ULONG tempratureData::timeOfFirstSample

#### 8.1036.2.4 WORD tempratureData::timeOffset[64]

#### 8.1036.2.5 ULONG tempratureData::timeSource

## 8.1037 TFTIDParams Struct Reference

### Data Fields

- BYTE filterId
- BYTE eValid
- BYTE ipVersion
- WORD * pSourceIP
- BYTE sourceIPMask
- BYTE nextHeader
- WORD destPortRangeStart
- WORD destPortRangeEnd
- WORD srcPortRangeStart

- WORD srcPortRangeEnd
- ULONG IPSECSPI
- WORD tosMask
- ULONG flowLabel

### 8.1037.1 Detailed Description

structure contains traffic flow template parameters

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

**Parameters**

| | |
|---|---|
| *filterId* | • Filter identifier |
| *eValid* | • Evaluation precedence index |
| *pVersion* | • IP version number<br>– 4 - IPv4<br>– 6 - IPv6 |
| *sourceIP* | • Source IP address<br>– IPv4 - Fill the first 4 bytes<br>– IPv6 - Fill all the 16 bytes |
| *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 |

### 8.1037.2 Field Documentation

#### 8.1037.2.1 WORD TFTIDParams::destPortRangeEnd

#### 8.1037.2.2 WORD TFTIDParams::destPortRangeStart

#### 8.1037.2.3 BYTE TFTIDParams::eValid

#### 8.1037.2.4 BYTE TFTIDParams::filterId

#### 8.1037.2.5 ULONG TFTIDParams::flowLabel

#### 8.1037.2.6 ULONG TFTIDParams::IPSECSPI

#### 8.1037.2.7 BYTE TFTIDParams::ipVersion

#### 8.1037.2.8 BYTE TFTIDParams::nextHeader

#### 8.1037.2.9 WORD* TFTIDParams::pSourceIP

#### 8.1037.2.10 BYTE TFTIDParams::sourceIPMask

#### 8.1037.2.11 WORD TFTIDParams::srcPortRangeEnd

#### 8.1037.2.12 WORD TFTIDParams::srcPortRangeStart

#### 8.1037.2.13 WORD TFTIDParams::tosMask

## 8.1038 timeInfo Struct Reference

### Data Fields

- WORD year
- BYTE month
- BYTE day
- BYTE hour
- BYTE minute
- BYTE second
- BYTE dayOfWeek
- INT8 timeZone
- BYTE dayLtSavingAdj
- BYTE radioInterface
- BYTE TlvPresent

### 8.1038.1 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. |

### 8.1038.2 Field Documentation

#### 8.1038.2.1 BYTE timeInfo::day

#### 8.1038.2.2 BYTE timeInfo::dayLtSavingAdj

#### 8.1038.2.3 BYTE timeInfo::dayOfWeek

#### 8.1038.2.4 BYTE timeInfo::hour

#### 8.1038.2.5 BYTE timeInfo::minute

#### 8.1038.2.6 BYTE timeInfo::month

#### 8.1038.2.7 BYTE timeInfo::radioInterface

#### 8.1038.2.8 BYTE timeInfo::second

#### 8.1038.2.9 INT8 timeInfo::timeZone

#### 8.1038.2.10 BYTE timeInfo::TlvPresent

#### 8.1038.2.11 WORD timeInfo::year

## 8.1039 TmdDeRegNotMitigationLvlReq Struct Reference

### Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

### 8.1039.1 Detailed Description

This structure contains mitigation devices Level deregister request parameters

**Parameters**

| | |
|---|---|
| *mitigationDevID-Len* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevID* | • Mitigation device ID |

### 8.1039.2 Field Documentation

#### 8.1039.2.1 CHAR TmdDeRegNotMitigationLvlReq::mitigationDevID[255]

#### 8.1039.2.2 BYTE TmdDeRegNotMitigationLvlReq::mitigationDevIDLen

## 8.1040 TmdGetMitigationDevListResp Struct Reference

### Data Fields

- BYTE * pMitigationDevListLen
- mitigationDevList * pMitigationDevList

#### 8.1040.1 Detailed Description

This structure contains mitigation devices list from the remote endpoint

**Parameters**

| | |
|---|---|
| *pMitigationDevListLen* | • Mitigation Device List Length (Optional)<br>• Number of sets of the following elements<br>• pMitigationDevList |
| *pMitigationDevList* | • Mitigation Device List (Optional)<br>• See mitigationDevList for more information. |

#### 8.1040.2 Field Documentation

##### 8.1040.2.1 mitigationDevList* TmdGetMitigationDevListResp::pMitigationDevList

##### 8.1040.2.2 BYTE* TmdGetMitigationDevListResp::pMitigationDevListLen

## 8.1041 TmdGetMitigationLvlReq Struct Reference

### Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

#### 8.1041.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *mitigationDevIDLen* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevID* | • Mitigation device ID |

#### 8.1041.2 Field Documentation

##### 8.1041.2.1 CHAR TmdGetMitigationLvlReq::mitigationDevID[255]

#### 8.1041.2.2 BYTE TmdGetMitigationLvlReq::mitigationDevIDLen

## 8.1042 TmdGetMitigationLvlResp Struct Reference

### Data Fields

- BYTE * pCurrentmitigationLvl
- BYTE * pReqMitigationLvl

### 8.1042.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *p-Currentmitigation-Lvl* | • Current thermal mitigation level (Optional) |
| *pReqMitigation-Lvl* | • 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. |

### 8.1042.2 Field Documentation

#### 8.1042.2.1 BYTE* TmdGetMitigationLvlResp::pCurrentmitigationLvl

#### 8.1042.2.2 BYTE* TmdGetMitigationLvlResp::pReqMitigationLvl

## 8.1043 TmdMitigationLvlIndReq Struct Reference

### Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

### 8.1043.1 Detailed Description

This structure contains mitigation Level Indication request parameters

**Parameters**

| | |
|---|---|
| *mitigationDevID-Len* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevID* | • Mitigation device ID |

### 8.1043.2 Field Documentation

#### 8.1043.2.1 CHAR TmdMitigationLvlIndReq::mitigationDevID[255]

#### 8.1043.2.2 BYTE TmdMitigationLvlIndReq::mitigationDevIDLen

## 8.1044 TmdRegNotMitigationLvlReq Struct Reference

### Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

### 8.1044.1 Detailed Description

This structure contains mitigation devices Level register request parameters

**Parameters**

| | |
|---|---|
| *mitigationDevID-Len* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevID* | • Mitigation device ID |

### 8.1044.2 Field Documentation

#### 8.1044.2.1 CHAR TmdRegNotMitigationLvlReq::mitigationDevID[255]

#### 8.1044.2.2 BYTE TmdRegNotMitigationLvlReq::mitigationDevIDLen

## 8.1045 tokenBucket Struct Reference

### Data Fields

- ULONG peakRate
- ULONG tokenRate
- ULONG bucketSz

### 8.1045.1 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 |

### 8.1045.2 Field Documentation

#### 8.1045.2.1 ULONG tokenBucket::bucketSz

#### 8.1045.2.2 ULONG tokenBucket::peakRate

#### 8.1045.2.3 ULONG tokenBucket::tokenRate

## 8.1046 Tos Struct Reference

### Data Fields

- BYTE val
- BYTE mask

### 8.1046.1 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. |

### 8.1046.2 Field Documentation

#### 8.1046.2.1 BYTE Tos::mask

#### 8.1046.2.2 BYTE Tos::val

## 8.1047 transferRouteMessageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSTransferRouteMTMessageInfo TransferRouteMTMessageInfo

### 8.1047.1 Detailed Description

This structure contains Transfer route MT message TLV information.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |

| | |
|---|---|
| *TransferRouteMTMessageInfo* | • Transfer Route MT Message<br>• See sMSTransferRouteMTMessageInfo for more information |

### 8.1047.2 Field Documentation

#### 8.1047.2.1 uint8_t transferRouteMessageTlv::TlvPresent

#### 8.1047.2.2 sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo

## 8.1048 TransferStatInd Struct Reference

### Data Fields

- BYTE StatsPeriod
- ULONG StatsMask

### 8.1048.1 Detailed Description

This structure contains Transfer Statistcics Indicator

**Parameters**

| | |
|---|---|
| *StatsPeriod* | • Period between transfer statistics reports<br>– 0 - Do not report<br>– Other - Period between reports (seconds) |
| *StatsMask* | • Requested statistic bit mask. Each bit set causes the corresponding optional TLV to be sent in the indication. All unlisted bits are reserviced for future use and must be set to zero.<br>– 0x00000040 - Tx bytes OK<br>– 0x00000080 - Rx bytes OK |

### 8.1048.2 Field Documentation

#### 8.1048.2.1 ULONG TransferStatInd::StatsMask

#### 8.1048.2.2 BYTE TransferStatInd::StatsPeriod

## 8.1049 TransferStatsDataType Struct Reference

### Data Fields

- BYTE interval

### 8.1049.1 Field Documentation

#### 8.1049.1.1 BYTE TransferStatsDataType::interval

## 8.1050 TrStatInd Struct Reference

### Data Fields

- BYTE statsPeriod
- ULONG statsMask

### 8.1050.1 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. |

### 8.1050.2 Field Documentation

#### 8.1050.2.1 ULONG TrStatInd::statsMask

#### 8.1050.2.2 BYTE TrStatInd::statsPeriod

## 8.1051 trueIMSI Struct Reference

### Data Fields

- BYTE mccT [3]
- WORD imsiT1112
- BYTE imsiTS1 [7]
- BYTE imsiTS2 [3]
- BYTE imsiTaddrNum

### 8.1051.1 Detailed Description

This structure contains the parameters for True IMSI Information

**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 |

### 8.1051.2 Field Documentation

#### 8.1051.2.1 WORD trueIMSI::imsiT1112

#### 8.1051.2.2 BYTE trueIMSI::imsiTaddrNum

#### 8.1051.2.3 BYTE trueIMSI::imsiTS1[7]

#### 8.1051.2.4 BYTE trueIMSI::imsiTS2[3]

#### 8.1051.2.5 BYTE trueIMSI::mccT[3]

## 8.1052 TXAGCList Struct Reference

### Data Fields

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

### 8.1052.1 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 |

### 8.1052.2 Field Documentation

#### 8.1052.2.1 WORD* TXAGCList::pTXAIG

#### 8.1052.2.2 WORD* TXAGCList::pTXComprSlope

#### 8.1052.2.3 WORD* TXAGCList::pTXComprThres

#### 8.1052.2.4 WORD* TXAGCList::pTXExpSlope

#### 8.1052.2.5 WORD* TXAGCList::pTXExpThres

#### 8.1052.2.6 WORD* TXAGCList::pTXStaticGain

## 8.1053 txInfo Struct Reference

### Data Fields

- BYTE isInTraffic
- INT32 txPower

### 8.1053.1 Detailed Description

This structure contains the Tx Information.

**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. |

### 8.1053.2 Field Documentation

#### 8.1053.2.1 BYTE txInfo::isInTraffic

#### 8.1053.2.2 INT32 txInfo::txPower

## 8.1054 TXPCMIIRFltr Struct Reference

### Data Fields

- WORD * pFlag
- WORD * pStageCnt
- BYTE * pStage0Val
- BYTE * pStage1Val
- BYTE * pStage2Val
- BYTE * pStage3Val
- BYTE * pStage4Val

### 8.1054.1 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 |

### 8.1054.2 Field Documentation

#### 8.1054.2.1 WORD* TXPCMIIRFItr::pFlag

#### 8.1054.2.2 BYTE* TXPCMIIRFItr::pStage0Val

#### 8.1054.2.3 BYTE* TXPCMIIRFItr::pStage1Val

#### 8.1054.2.4 BYTE* TXPCMIIRFItr::pStage2Val

#### 8.1054.2.5 BYTE* TXPCMIIRFItr::pStage3Val

#### 8.1054.2.6 BYTE* TXPCMIIRFItr::pStage4Val

#### 8.1054.2.7 WORD* TXPCMIIRFItr::pStageCnt

## 8.1055 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

#### 8.1055.1 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_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. |

### 8.1055.2 Field Documentation

#### 8.1055.2.1 uint8_t uim_appStatus::aidLength

#### 8.1055.2.2 uint8_t uim_appStatus::aidVal[255]

#### 8.1055.2.3 uint8_t uim_appStatus::appState

#### 8.1055.2.4 uint8_t uim_appStatus::appType

#### 8.1055.2.5 uint8_t uim_appStatus::persoFeature

#### 8.1055.2.6 uint8_t uim_appStatus::persoRetries

#### 8.1055.2.7 uint8_t uim_appStatus::persoState

#### 8.1055.2.8 uint8_t uim_appStatus::persoUnblockRetries

#### 8.1055.2.9 uint8_t uim_appStatus::pin1Retries

#### 8.1055.2.10 uint8_t uim_appStatus::pin1State

#### 8.1055.2.11 uint8_t uim_appStatus::pin2Retries

#### 8.1055.2.12 uint8_t uim_appStatus::pin2State

#### 8.1055.2.13 uint8_t uim_appStatus::puk1Retries

#### 8.1055.2.14 uint8_t uim_appStatus::puk2Retries

#### 8.1055.2.15 uint8_t uim_appStatus::univPin

## 8.1056 uim_authenticateResult Struct Reference

### Data Fields

- uint16_t contentLen

- uint8_t content [1024]

### 8.1056.1 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. |

### 8.1056.2 Field Documentation

#### 8.1056.2.1 uint8_t uim_authenticateResult::content[1024]

#### 8.1056.2.2 uint16_t uim_authenticateResult::contentLen

## 8.1057 uim_authenticationData Struct Reference

### Data Fields

- uint8_t context
- uint16_t dataLen
- uint8_t data [1024]

### 8.1057.1 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-_DESCRIPTIO-N_LENGTH]* | • Authenticate Data. |

### 8.1057.2 Field Documentation

#### 8.1057.2.1 uint8_t uim_authenticationData::context

#### 8.1057.2.2 uint8_t uim_authenticationData::data[1024]

#### 8.1057.2.3 uint16_t uim_authenticationData::dataLen

## 8.1058 uim_cardResult Struct Reference

### Data Fields

- uint8_t sw1
- uint8_t sw2

### 8.1058.1 Detailed Description

This structure contains the information about the card result.

**Parameters**

| | |
|---|---|
| *sw1* | • SW1 received from the card. |
| *sw2* | • SW2 received from the card. |

### 8.1058.2 Field Documentation

#### 8.1058.2.1 uint8_t uim_cardResult::sw1

#### 8.1058.2.2 uint8_t uim_cardResult::sw2

## 8.1059 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]

### 8.1059.1 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. |

### 8.1059.2 Field Documentation

#### 8.1059.2.1 uint16_t uim_cardStatus::index1xPri

#### 8.1059.2.2 uint16_t uim_cardStatus::index1xSec

#### 8.1059.2.3 uint16_t uim_cardStatus::indexGwPri

#### 8.1059.2.4 uint16_t uim_cardStatus::indexGwSec

#### 8.1059.2.5 uint8_t uim_cardStatus::numSlot

#### 8.1059.2.6 uim_slotInfo uim_cardStatus::SlotInfo[5]

## 8.1060 uim_changeUIMPIN Struct Reference

### Data Fields

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

### 8.1060.1 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. |

### 8.1060.2 Field Documentation

#### 8.1060.2.1 uint8_t uim_changeUIMPIN::oldPINLen

#### 8.1060.2.2 uint8_t uim_changeUIMPIN::oldPINVal[255]

#### 8.1060.2.3 uint8_t uim_changeUIMPIN::pinID

#### 8.1060.2.4 uint8_t uim_changeUIMPIN::pinLen

#### 8.1060.2.5 uint8_t uim_changeUIMPIN::pinVal[255]

## 8.1061 uim_depersonalizationInformation Struct Reference

### Data Fields

- uint8_t feature
- uint8_t operation
- uint8_t ckLen
- uint8_t ckVal [255]

### 8.1061.1 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. |

### 8.1061.2 Field Documentation

#### 8.1061.2.1 uint8_t uim_depersonalizationInformation::ckLen

#### 8.1061.2.2 uint8_t uim_depersonalizationInformation::ckVal[255]

#### 8.1061.2.3 uint8_t uim_depersonalizationInformation::feature

#### 8.1061.2.4 uint8_t uim_depersonalizationInformation::operation

## 8.1062 uim_encryptedPIN1 Struct Reference

### Data Fields

- uint8_t pin1Len
- uint8_t pin1Val [255]

### 8.1062.1 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.

### 8.1062.2 Field Documentation

#### 8.1062.2.1 uint8_t uim_encryptedPIN1::pin1Len

#### 8.1062.2.2 uint8_t uim_encryptedPIN1::pin1Val[255]

## 8.1063 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]

### 8.1063.1 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. |

### 8.1063.2 Field Documentation

#### 8.1063.2.1 uint16_t uim_fileAttributes::fileID

#### 8.1063.2.2 uint16_t uim_fileAttributes::fileSize

#### 8.1063.2.3 uint8_t uim_fileAttributes::fileType

#### 8.1063.2.4 uint16_t uim_fileAttributes::rawLen

#### 8.1063.2.5 uint8_t uim_fileAttributes::rawValue[255]

#### 8.1063.2.6 uint16_t uim_fileAttributes::recordCount

#### 8.1063.2.7 uint16_t uim_fileAttributes::recordSize

#### 8.1063.2.8 uint8_t uim_fileAttributes::secActivate

#### 8.1063.2.9 uint16_t uim_fileAttributes::secActivateMask

#### 8.1063.2.10 uint8_t uim_fileAttributes::secDeactivate

#### 8.1063.2.11 uint16_t uim_fileAttributes::secDeactivateMask

#### 8.1063.2.12 uint8_t uim_fileAttributes::secIncrease

#### 8.1063.2.13 uint16_t uim_fileAttributes::secIncreaseMask

#### 8.1063.2.14 uint8_t uim_fileAttributes::secRead

#### 8.1063.2.15 uint16_t uim_fileAttributes::secReadMask

#### 8.1063.2.16 uint8_t uim_fileAttributes::secWrite

#### 8.1063.2.17 uint16_t uim_fileAttributes::secWriteMask

## 8.1064 uim_fileInfo Struct Reference

### Data Fields

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

### 8.1064.1 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. |

### 8.1064.2 Field Documentation

#### 8.1064.2.1 uint16_t uim_fileInfo::fileID

#### 8.1064.2.2 uint16_t uim_fileInfo::path[255]

#### 8.1064.2.3 uint8_t uim_fileInfo::pathLen

## 8.1065 uim_hotSwapStatus Struct Reference

### Data Fields

- uint8_t hotSwapLength
- uint8_t hotSwap [255]

### 8.1065.1 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 |

### 8.1065.2 Field Documentation

#### 8.1065.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

#### 8.1065.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

## 8.1066 uim_personalizationStatus Struct Reference

### Data Fields

- uint8_t numFeatures
- uint8_t feature [12]
- uint8_t verifyLeft [12]
- uint8_t unblockLeft [12]

#### 8.1066.1 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. |

#### 8.1066.2 Field Documentation

##### 8.1066.2.1 uint8_t uim_personalizationStatus::feature[12]

##### 8.1066.2.2 uint8_t uim_personalizationStatus::numFeatures

##### 8.1066.2.3 uint8_t uim_personalizationStatus::unblockLeft[12]

##### 8.1066.2.4 uint8_t uim_personalizationStatus::verifyLeft[12]

## 8.1067 uim_readResult Struct Reference

### Data Fields

- uint16_t contentLen
- uint8_t content [255]

### 8.1067.1 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. |

### 8.1067.2 Field Documentation

#### 8.1067.2.1 uint8_t uim_readResult::content[255]

#### 8.1067.2.2 uint16_t uim_readResult::contentLen

## 8.1068 uim_readTransparentInfo Struct Reference

### Data Fields

- uint16_t offset
- uint16_t length

### 8.1068.1 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. |

### 8.1068.2 Field Documentation

#### 8.1068.2.1 uint16_t uim_readTransparentInfo::length

#### 8.1068.2.2 uint16_t uim_readTransparentInfo::offset

## 8.1069 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]

### 8.1069.1 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 |

### 8.1069.2 Field Documentation

#### 8.1069.2.1 uint8_t uim_refreshevent::aid[255]

#### 8.1069.2.2 uint8_t uim_refreshevent::aidLength

#### 8.1069.2.3 uim_fileInfo uim_refreshevent::arrfileInfo[255]

#### 8.1069.2.4 uint8_t uim_refreshevent::mode

#### 8.1069.2.5 uint16_t uim_refreshevent::numOfFiles

#### 8.1069.2.6 uint8_t uim_refreshevent::sessionType

#### 8.1069.2.7 uint8_t uim_refreshevent::stage

## 8.1070 uim_registerRefresh Struct Reference

### Data Fields

- uint8_t registerFlag
- uint8_t voteForInit
- uint16_t numFiles
- uim_fileInfo arrfileInfo [255]

### 8.1070.1 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 |

### 8.1070.2 Field Documentation

#### 8.1070.2.1 uim_fileInfo uim_registerRefresh::arrfileInfo[255]

#### 8.1070.2.2 uint16_t uim_registerRefresh::numFiles

#### 8.1070.2.3 uint8_t uim_registerRefresh::registerFlag

#### 8.1070.2.4 uint8_t uim_registerRefresh::voteForInit

## 8.1071 uim_remainingRetries Struct Reference

### Data Fields

- uint8_t verifyLeft
- uint8_t unblockLeft

### 8.1071.1 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.

### 8.1071.2 Field Documentation

#### 8.1071.2.1 uint8_t uim_remainingRetries::unblockLeft

#### 8.1071.2.2 uint8_t uim_remainingRetries::verifyLeft

## 8.1072 uim_sessionInformation Struct Reference

### Data Fields

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

### 8.1072.1 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. |

### 8.1072.2 Field Documentation

#### 8.1072.2.1 uint8_t uim_sessionInformation::aid[255]

#### 8.1072.2.2 uint8_t uim_sessionInformation::aidLength

#### 8.1072.2.3 uint8_t uim_sessionInformation::sessionType

## 8.1073 uim_setPINProtection Struct Reference

### Data Fields

- uint8_t pinID
- uint8_t pinOperation

- uint8_t pinLength
- uint8_t pinValue [255]

### 8.1073.1 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. |

### 8.1073.2 Field Documentation

#### 8.1073.2.1 uint8_t uim_setPINProtection::pinID

#### 8.1073.2.2 uint8_t uim_setPINProtection::pinLength

#### 8.1073.2.3 uint8_t uim_setPINProtection::pinOperation

#### 8.1073.2.4 uint8_t uim_setPINProtection::pinValue[255]

## 8.1074 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]

### 8.1074.1 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. |

### 8.1074.2 Field Documentation

#### 8.1074.2.1 uim_appStatus uim_slotInfo::AppStatus[10]

#### 8.1074.2.2 uint8_t uim_slotInfo::cardState

#### 8.1074.2.3 uint8_t uim_slotInfo::errorState

#### 8.1074.2.4 uint8_t uim_slotInfo::numApp

#### 8.1074.2.5 uint8_t uim_slotInfo::upinRetries

#### 8.1074.2.6 uint8_t uim_slotInfo::upinState

#### 8.1074.2.7 uint8_t uim_slotInfo::upukRetries

## 8.1075 uim_UIMSessionInformation Struct Reference

### Data Fields

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

### 8.1075.1 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. |

### 8.1075.2 Field Documentation

#### 8.1075.2.1 uint8_t uim_UIMSessionInformation::aid[255]

#### 8.1075.2.2 uint8_t uim_UIMSessionInformation::aidLength

#### 8.1075.2.3 uint8_t uim_UIMSessionInformation::sessionType

## 8.1076 uim_unblockUIMPIN Struct Reference

### Data Fields

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

### 8.1076.1 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. |

### 8.1076.2 Field Documentation

#### 8.1076.2.1 uint8_t uim_unblockUIMPIN::newPINLen

#### 8.1076.2.2 uint8_t uim_unblockUIMPIN::newPINVal[255]

#### 8.1076.2.3 uint8_t uim_unblockUIMPIN::pinID

#### 8.1076.2.4 uint8_t uim_unblockUIMPIN::pukLen

#### 8.1076.2.5 uint8_t uim_unblockUIMPIN::pukVal[255]

## 8.1077 uim_verifyUIMPIN Struct Reference

### Data Fields

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

### 8.1077.1 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. |

### 8.1077.2 Field Documentation

#### 8.1077.2.1 uint8_t uim_verifyUIMPIN::pinID

#### 8.1077.2.2 uint8_t uim_verifyUIMPIN::pinLen

#### 8.1077.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

## 8.1078 UIMAuthenticateReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- authenticationData authData
- ULONG * pIndicationToken

### 8.1078.1 Detailed Description

This structure contains information of the request parameters associated with a Authenticate API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See UIMSessionInformation for more information. |
| *authData* | • See 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.

### 8.1078.2 Field Documentation

#### 8.1078.2.1 authenticationData UIMAuthenticateReq::authData

#### 8.1078.2.2 ULONG* UIMAuthenticateReq::pIndicationToken

#### 8.1078.2.3 UIMSessionInformation UIMAuthenticateReq::sessionInfo

## 8.1079 UIMAuthenticateResp Struct Reference

### Data Fields

- cardResult * pCardResult
- authenticateResult * pAuthenticateResult
- ULONG * pIndicationToken

### 8.1079.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pCard-Result(optional)* | • See cardResult for more information. |
| *pAuthenticate-Result(optional)* | • See authenticateResult 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 returned.

### 8.1079.2 Field Documentation

#### 8.1079.2.1 authenticateResult∗ UIMAuthenticateResp::pAuthenticateResult

#### 8.1079.2.2 cardResult∗ UIMAuthenticateResp::pCardResult

#### 8.1079.2.3 ULONG∗ UIMAuthenticateResp::pIndicationToken

## 8.1080 UIMChangePinReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- changeUIMPIN changePIN
- BYTE ∗ pKeyReferenceID
- ULONG ∗ pIndicationToken

### 8.1080.1 Detailed Description

This structure contains information of the request parameters associated with a Change PIN API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See UIMSessionInformation for more information. |
| *changePIN* | • See 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. |

**Note**

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

### 8.1080.2 Field Documentation

#### 8.1080.2.1 changeUIMPIN UIMChangePinReq::changePIN

#### 8.1080.2.2 ULONG∗ UIMChangePinReq::pIndicationToken

#### 8.1080.2.3 BYTE∗ UIMChangePinReq::pKeyReferenceID

#### 8.1080.2.4 UIMSessionInformation UIMChangePinReq::sessionInfo

## 8.1081 UIMDepersonalizationReq Struct Reference

### Data Fields

- depersonalizationInformation depersonilisationInfo

### 8.1081.1 Detailed Description

This structure contains information of the request parameters associated with a Depersonalization API.

**Parameters**

| | |
|---|---|
| *depersonilisation-Info* | • See depersonalizationInformation for more information. |

### 8.1081.2 Field Documentation

#### 8.1081.2.1 depersonalizationInformation UIMDepersonalizationReq::depersonilisationInfo

## 8.1082 UIMDepersonalizationResp Struct Reference

### Data Fields

- remainingRetries * pRemainingRetries

### 8.1082.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pRemaining-Retries(optional)* | • See remainingRetries for more information. |

**Note**

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

### 8.1082.2 Field Documentation

#### 8.1082.2.1 remainingRetries* UIMDepersonalizationResp::pRemainingRetries

## 8.1083 UIMEventRegisterReqResp Struct Reference

### Data Fields

- ULONG eventMask

### 8.1083.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *eventMask(-Mandatory)* | • Bitmask of the events that were successfully enabled. This result can be different from the mask used in the request when notifications are not supported. Additional bits are reserved for future use.<br>– Bit 0 - Card status<br>– Bit 1 - SAP connection<br>– Bit 4 - Physical Slot Status |

### 8.1083.2 Field Documentation

#### 8.1083.2.1 ULONG UIMEventRegisterReqResp::eventMask

## 8.1084 UIMGetCardStatusResp Struct Reference

### Data Fields

- cardStatus * pCardStatus
- hotSwapStatus * pHotSwapStatus

### 8.1084.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pCard-Status(optional)* | • See cardStatus for more information. |
| *pHotSwap-Status(optional)* | • See hotSwapStatus for more information. |

**Note**

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

### 8.1084.2 Field Documentation

#### 8.1084.2.1 cardStatus* UIMGetCardStatusResp::pCardStatus

#### 8.1084.2.2 hotSwapStatus* UIMGetCardStatusResp::pHotSwapStatus

## 8.1085 UIMGetConfigurationReq Struct Reference

### Data Fields

- ULONG * pConfigurationMask

### 8.1085.1 Detailed Description

This structure contains information of the request parameters associated with to gets the modem configuration for the UIM module API.

**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.

### 8.1085.2 Field Documentation

#### 8.1085.2.1 ULONG* UIMGetConfigurationReq::pConfigurationMask

## 8.1086 UIMGetConfigurationResp Struct Reference

### Data Fields

- BYTE * pAutoSelection
- personalizationStatus * pPersonalizationStatus
- BYTE * pHaltSubscription

### 8.1086.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pAuto-Selection(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 |
| *p-Personalization-Status(optional)* | • See personalizationStatus for more information. |
| *pHalt-Subscription(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 |

### 8.1086.2 Field Documentation

#### 8.1086.2.1 BYTE* UIMGetConfigurationResp::pAutoSelection

#### 8.1086.2.2 BYTE* UIMGetConfigurationResp::pHaltSubscription

#### 8.1086.2.3 personalizationStatus* UIMGetConfigurationResp::pPersonalizationStatus

## 8.1087 UIMGetFileAttributesReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- ULONG * pIndicationToken

### 8.1087.1 Detailed Description

This structure contains information of the request parameters associated with a Get File Attributes API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See UIMSessionInformation for more information. |
| *fileIndex* | • See 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.

### 8.1087.2 Field Documentation

#### 8.1087.2.1 fileInfo UIMGetFileAttributesReq::fileIndex

#### 8.1087.2.2 ULONG* UIMGetFileAttributesReq::pIndicationToken

#### 8.1087.2.3 UIMSessionInformation UIMGetFileAttributesReq::sessionInfo

## 8.1088 UIMGetFileAttributesResp Struct Reference

### Data Fields

- cardResult * pCardResult
- fileAttributes * pFileAttributes
- ULONG * pIndicationToken

### 8.1088.1 Detailed Description

This structure contains information of the response parameters associated with a Get File Attributes API.

**Parameters**

| | |
|---|---|
| *pCard-Result(optional)* | • See cardResult for more information. |
| *pFile-Attributes(optional)* | • See fileAttributes 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 returned.

### 8.1088.2 Field Documentation

#### 8.1088.2.1 cardResult∗ UIMGetFileAttributesResp::pCardResult

#### 8.1088.2.2 fileAttributes∗ UIMGetFileAttributesResp::pFileAttributes

#### 8.1088.2.3 ULONG∗ UIMGetFileAttributesResp::pIndicationToken

## 8.1089 UIMGetSlotsStatusResp Struct Reference

### Data Fields

- BYTE ∗ pNumberOfPhySlot
- UIMSlotsStatus ∗ pUimSlotsStatus

### 8.1089.1 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. |
| *pUimSlotsStatus* | • Slots Status See UIMSlotsStatus for more information.. |

### 8.1089.2 Field Documentation

#### 8.1089.2.1 BYTE∗ UIMGetSlotsStatusResp::pNumberOfPhySlot

#### 8.1089.2.2 UIMSlotsStatus* UIMGetSlotsStatusResp::pUimSlotsStatus

## 8.1090 UIMPinResp Struct Reference

### Data Fields

- remainingRetries * pRemainingRetries
- encryptedPIN1 * pEncryptedPIN1
- ULONG * pIndicationToken

### 8.1090.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pRemaining-Retries(optional)* | • See remainingRetries for more information. |
| *pEncryptedPI-N1(optional)* | • See encryptedPIN1 for more information. |
| *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 |

**Note**

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

### 8.1090.2 Field Documentation

#### 8.1090.2.1 encryptedPIN1* UIMPinResp::pEncryptedPIN1

#### 8.1090.2.2 ULONG* UIMPinResp::pIndicationToken

#### 8.1090.2.3 remainingRetries* UIMPinResp::pRemainingRetries

## 8.1091 UIMPowerDownReq Struct Reference

### Data Fields

- BYTE slot

### 8.1091.1 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 |

### 8.1091.2 Field Documentation

#### 8.1091.2.1 BYTE UIMPowerDownReq::slot

## 8.1092 UIMPowerUpReq Struct Reference

### Data Fields

- BYTE slot
- BYTE * pIgnoreHotSwapSwitch

### 8.1092.1 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 |

### 8.1092.2 Field Documentation

#### 8.1092.2.1 BYTE* UIMPowerUpReq::pIgnoreHotSwapSwitch

#### 8.1092.2.2 BYTE UIMPowerUpReq::slot

## 8.1093 UIMReadTransparentReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- readTransparentInfo readTransparent
- ULONG * pIndicationToken
- BYTE * pEncryptData

### 8.1093.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See UIMSessionInformation for more information. |
| *fileIndex* | • See fileInfo for more information. |
| *readTransparent* | • See 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. |
| *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.

### 8.1093.2 Field Documentation

#### 8.1093.2.1 fileInfo UIMReadTransparentReq::fileIndex

#### 8.1093.2.2 BYTE* UIMReadTransparentReq::pEncryptData

#### 8.1093.2.3 ULONG* UIMReadTransparentReq::pIndicationToken

#### 8.1093.2.4 readTransparentInfo UIMReadTransparentReq::readTransparent

#### 8.1093.2.5 UIMSessionInformation UIMReadTransparentReq::sessionInfo

## 8.1094 UIMReadTransparentResp Struct Reference

### Data Fields

- cardResult * pCardResult
- readResult * pReadResult
- ULONG * pIndicationToken
- BYTE * pEncryptedData

### 8.1094.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pCardResult* | • See cardResult for more information. |
| *pReadResult* | • See readResult 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. |
| *pEncrypted-Data(optional)* | • Encrypted Data.<br>• Indicates whether the data from the card passed in read_result is encrypted. |

**Note**

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

### 8.1094.2 Field Documentation

#### 8.1094.2.1 cardResult* UIMReadTransparentResp::pCardResult

#### 8.1094.2.2 BYTE* UIMReadTransparentResp::pEncryptedData

#### 8.1094.2.3 ULONG* UIMReadTransparentResp::pIndicationToken

#### 8.1094.2.4 readResult* UIMReadTransparentResp::pReadResult

## 8.1095 UIMRefreshCompleteReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- BYTE refreshComplete

### 8.1095.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *sessionInfo(-Mandatory)* | • See UIMSessionInformation 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 |

### 8.1095.2 Field Documentation

#### 8.1095.2.1 BYTE UIMRefreshCompleteReq::refreshComplete

#### 8.1095.2.2 UIMSessionInformation UIMRefreshCompleteReq::sessionInfo

## 8.1096 UIMRefreshEvent Struct Reference

### Data Fields

- BYTE stage
- BYTE mode
- BYTE sessionType
- BYTE aidLength
- BYTE aid [255]
- WORD numOfFiles
- fileInfo arrfileInfo [255]

### 8.1096.1 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 |

### 8.1096.2 Field Documentation

#### 8.1096.2.1 BYTE UIMRefreshEvent::aid[255]

#### 8.1096.2.2 BYTE UIMRefreshEvent::aidLength

#### 8.1096.2.3 fileInfo UIMRefreshEvent::arrfileInfo[255]

#### 8.1096.2.4 BYTE UIMRefreshEvent::mode

#### 8.1096.2.5 WORD UIMRefreshEvent::numOfFiles

#### 8.1096.2.6 BYTE UIMRefreshEvent::sessionType

#### 8.1096.2.7 BYTE UIMRefreshEvent::stage

## 8.1097 UIMRefreshGetLastEventReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo

### 8.1097.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *sessionInfo(-Mandatory)* | • See UIMSessionInformation for more information. |

### 8.1097.2 Field Documentation

#### 8.1097.2.1 UIMSessionInformation UIMRefreshGetLastEventReq::sessionInfo

## 8.1098 UIMRefreshGetLastEventResp Struct Reference

### Data Fields

- UIMRefreshEvent * pRefreshEvent

### 8.1098.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMRefreshGetLastEvent.

**Parameters**

| | |
|---|---|
| *refreshEvent(-Optional)* | • See UIMRefreshEvent for more information. |

### 8.1098.2 Field Documentation

#### 8.1098.2.1 UIMRefreshEvent* UIMRefreshGetLastEventResp::pRefreshEvent

## 8.1099 UIMRefreshOKReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- BYTE OKtoRefresh

### 8.1099.1 Detailed Description

This structure contains Parameters of the Session Information

**Parameters**

| | |
|---|---|
| *sessionInfo* | • Session Information<br>• See UIMSessionInformation for more information |

| | |
|---|---|
| *OKtoRefresh* | • Indicates whether a refresh is OK. Valid values:<br>– 0 - Not OK to refresh<br>– 1 - OK to refresh |

### 8.1099.2 Field Documentation

#### 8.1099.2.1 BYTE UIMRefreshOKReq::OKtoRefresh

#### 8.1099.2.2 UIMSessionInformation UIMRefreshOKReq::sessionInfo

## 8.1100 UIMRefreshRegisterReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- registerRefresh regRefresh

### 8.1100.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Register.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • Session Information params<br>• See UIMSessionInformation for more information |
| *regRefresh* | • Register Refresh parameters<br>• See registerRefresh for more information |

### 8.1100.2 Field Documentation

#### 8.1100.2.1 registerRefresh UIMRefreshRegisterReq::regRefresh

#### 8.1100.2.2 UIMSessionInformation UIMRefreshRegisterReq::sessionInfo

## 8.1101 UIMSessionInformation Struct Reference

### Data Fields

- BYTE sessionType
- BYTE aidLength
- BYTE aid [255]

### 8.1101.1 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. |

### 8.1101.2 Field Documentation

#### 8.1101.2.1 BYTE UIMSessionInformation::aid[255]

#### 8.1101.2.2 BYTE UIMSessionInformation::aidLength

#### 8.1101.2.3 BYTE UIMSessionInformation::sessionType

## 8.1102 UIMSetPinProtectionReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- setPINProtection pinProtection
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

### 8.1102.1 Detailed Description

This structure contains information of the request parameters associated with a set pin protection API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See UIMSessionInformation for more information. |

| | |
|---|---|
| *pinProtection* | • See setPINProtection 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. |

**Note**

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

### 8.1102.2 Field Documentation

#### 8.1102.2.1 ULONG* UIMSetPinProtectionReq::pIndicationToken

#### 8.1102.2.2 setPINProtection UIMSetPinProtectionReq::pinProtection

#### 8.1102.2.3 BYTE* UIMSetPinProtectionReq::pKeyReferenceID

#### 8.1102.2.4 UIMSessionInformation UIMSetPinProtectionReq::sessionInfo

## 8.1103 UIMSlotsStatus Struct Reference

### Data Fields

- UIMSlotStatus uimSlotStatus [255]

### 8.1103.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *uimSlotStatus[-MAX_SLOTS_S-TATUS]* | • Contain all slots status. |

### 8.1103.2 Field Documentation

#### 8.1103.2.1 UIMSlotStatus UIMSlotsStatus::uimSlotStatus[255]

## 8.1104 UIMSlotStatus Struct Reference

### Data Fields

- ULONG uPhyCardStatus
- ULONG uPhySlotStatus
- BYTE bLogicalSlot
- BYTE bICCIDLength
- BYTE bICCID [255]

### 8.1104.1 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. |

### 8.1104.2 Field Documentation

#### 8.1104.2.1 BYTE UIMSlotStatus::bICCID[255]

#### 8.1104.2.2 BYTE UIMSlotStatus::bICCIDLength

#### 8.1104.2.3 BYTE UIMSlotStatus::bLogicalSlot

#### 8.1104.2.4 ULONG UIMSlotStatus::uPhyCardStatus

#### 8.1104.2.5 ULONG UIMSlotStatus::uPhySlotStatus

## 8.1105 UIMSlotStatusChangeInfo Struct Reference

### Data Fields

- UIMSlotsStatus slotsstatusChange
- BYTE bNumberOfPhySlots

#### 8.1105.1 Detailed Description

Structure consist of cardstatus params

**Parameters**

| | |
|---|---|
| *slotstatus-Change* | • See UIMSlotStatus for more information |
| *bNumberOfPhy-Slots* | • Number of Physical Slot(s) |

#### 8.1105.2 Field Documentation

##### 8.1105.2.1 BYTE UIMSlotStatusChangeInfo::bNumberOfPhySlots

##### 8.1105.2.2 UIMSlotsStatus UIMSlotStatusChangeInfo::slotsstatusChange

## 8.1106 UIMStatusChangeInfo Struct Reference

### Data Fields

- cardStatus statusChange

#### 8.1106.1 Detailed Description

Structure consist of cardstatus params

**Parameters**

| | |
|---|---|
| *statusChange* | • See cardStatus for more information |

#### 8.1106.2 Field Documentation

##### 8.1106.2.1 cardStatus UIMStatusChangeInfo::statusChange

## 8.1107 UIMSwitchSlotReq Struct Reference

### Data Fields

- BYTE bLogicalSlot
- ULONG ulPhysicalSlot

### 8.1107.1 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 |
| *bPhysicalSlot* | • 1 - Slot 1<br>• 2 - Slot 2<br>• 3 - Slot 3<br>• 4 - Slot 4<br>• 5 - Slot 5 |

### 8.1107.2 Field Documentation

#### 8.1107.2.1 BYTE UIMSwitchSlotReq::bLogicalSlot

#### 8.1107.2.2 ULONG UIMSwitchSlotReq::ulPhysicalSlot

## 8.1108 UIMUnblockPinReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- unblockUIMPIN unblockPIN
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

### 8.1108.1 Detailed Description

This structure contains information of the request parameters associated with a Unblock PIN API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See UIMSessionInformation for more information. |
| *unblockPIN* | • See unblockUIMPIN 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. |

**Note**

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

### 8.1108.2 Field Documentation

#### 8.1108.2.1 ULONG* UIMUnblockPinReq::pIndicationToken

#### 8.1108.2.2 BYTE* UIMUnblockPinReq::pKeyReferenceID

#### 8.1108.2.3 UIMSessionInformation UIMUnblockPinReq::sessionInfo

#### 8.1108.2.4 unblockUIMPIN UIMUnblockPinReq::unblockPIN

## 8.1109 UIMVerifyPinReq Struct Reference

### Data Fields

- UIMSessionInformation sessionInfo
- verifyUIMPIN verifyPIN
- encryptedPIN1 * pEncryptedPIN1
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

### 8.1109.1 Detailed Description

This structure contains information of the request parameters associated with a verify PIN API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See UIMSessionInformation for more information. |
| *verifyPIN* | • See verifyUIMPIN for more information. |
| *pEncryptedPIN1(optional)* | • See encryptedPIN1 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. |

**Note**

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

### 8.1109.2 Field Documentation

#### 8.1109.2.1 encryptedPIN1∗ UIMVerifyPinReq::pEncryptedPIN1

#### 8.1109.2.2 ULONG∗ UIMVerifyPinReq::pIndicationToken

#### 8.1109.2.3 BYTE∗ UIMVerifyPinReq::pKeyReferenceID

#### 8.1109.2.4 UIMSessionInformation UIMVerifyPinReq::sessionInfo

#### 8.1109.2.5 verifyUIMPIN UIMVerifyPinReq::verifyPIN

## 8.1110 UMTSInfo Struct Reference

### Data Fields

- WORD cellID
- BYTE plmn [3]
- WORD lac
- WORD uarfcn
- WORD psc
- SHORT rscp
- SHORT ecio
- BYTE umtsInst
- UMTSinstInfo UMTSInstInfo [255]
- BYTE geranInst
- geranInstInfo GeranInstInfo [255]

### 8.1110.1 Detailed Description

This structure contains information about the UMTS Network.

**Parameters**

| | |
|---|---|
| *cellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present. |

| | |
|---|---|
| *plmn[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[MAX_DESCRIPTION_LENGTH]* | • See UMTSinstInfo for more information. |
| *geranInst* | • Provides the number of set of GERAN info instances.<br>• If 0(zero), then no information follows it. |
| *GeranInstInfo[MAX_DESCRIPTION_LENGTH]* | • See geranInstInfo for more information. |

## 8.1110.2 Field Documentation

### 8.1110.2.1 WORD UMTSInfo::cellID

### 8.1110.2.2 SHORT UMTSInfo::ecio

### 8.1110.2.3 BYTE UMTSInfo::geranInst

### 8.1110.2.4 geranInstInfo UMTSInfo::GeranInstInfo[255]

### 8.1110.2.5 WORD UMTSInfo::lac

#### 8.1110.2.6 BYTE UMTSInfo::plmn[3]

#### 8.1110.2.7 WORD UMTSInfo::psc

#### 8.1110.2.8 SHORT UMTSInfo::rscp

#### 8.1110.2.9 WORD UMTSInfo::uarfcn

#### 8.1110.2.10 BYTE UMTSInfo::umtsInst

#### 8.1110.2.11 UMTSinstInfo UMTSInfo::UMTSInstInfo[255]

## 8.1111 UMTSinstInfo Struct Reference

### Data Fields

- WORD umtsUarfcn
- WORD umtsPsc
- SHORT umtsRscp
- SHORT umtsEcio

### 8.1111.1 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). |

### 8.1111.2 Field Documentation

#### 8.1111.2.1 SHORT UMTSinstInfo::umtsEcio

#### 8.1111.2.2 WORD UMTSinstInfo::umtsPsc

#### 8.1111.2.3 SHORT UMTSinstInfo::umtsRscp

#### 8.1111.2.4 WORD UMTSinstInfo::umtsUarfcn

## 8.1112 umtsLTENbrCell Struct Reference

### Data Fields

- WORD earfcn
- WORD pci
- ULONG rsrp
- ULONG rsrq
- SHORT srxlev
- BYTE cellIsTDD

### 8.1112.1 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. |

### 8.1112.2 Field Documentation

#### 8.1112.2.1 BYTE umtsLTENbrCell::cellIsTDD

#### 8.1112.2.2 WORD umtsLTENbrCell::earfcn

#### 8.1112.2.3 WORD umtsLTENbrCell::pci

#### 8.1112.2.4 ULONG umtsLTENbrCell::rsrp

#### 8.1112.2.5 ULONG umtsLTENbrCell::rsrq

#### 8.1112.2.6 SHORT umtsLTENbrCell::srxlev

## 8.1113 UMTSMinQoS Struct Reference

### Data Fields

- BYTE trafficClass
- ULONG maxUplinkBitrate
- ULONG maxDownlinkBitrate
- ULONG grntUplinkBitrate
- ULONG grntDownlinkBitrate
- BYTE qosDeliveryOrder
- ULONG maxSDUSize
- BYTE sduErrorRatio
- BYTE resBerRatio
- BYTE deliveryErrSDU
- ULONG transferDelay
- ULONG trafficPriority

### 8.1113.1 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. |

### 8.1113.2 Field Documentation

#### 8.1113.2.1 BYTE UMTSMinQoS::deliveryErrSDU

#### 8.1113.2.2 ULONG UMTSMinQoS::grntDownlinkBitrate

#### 8.1113.2.3 ULONG UMTSMinQoS::grntUplinkBitrate

#### 8.1113.2.4 ULONG UMTSMinQoS::maxDownlinkBitrate

#### 8.1113.2.5 ULONG UMTSMinQoS::maxSDUSize

#### 8.1113.2.6 ULONG UMTSMinQoS::maxUplinkBitrate

#### 8.1113.2.7 BYTE UMTSMinQoS::qosDeliveryOrder

#### 8.1113.2.8 BYTE UMTSMinQoS::resBerRatio

#### 8.1113.2.9 BYTE UMTSMinQoS::sduErrorRatio

#### 8.1113.2.10 BYTE UMTSMinQoS::trafficClass

#### 8.1113.2.11 ULONG UMTSMinQoS::trafficPriority

#### 8.1113.2.12 ULONG UMTSMinQoS::transferDelay

## 8.1114 UMTSQoS Struct Reference

### Data Fields

- BYTE trafficClass
- ULONG maxUplinkBitrate
- ULONG maxDownlinkBitrate
- ULONG grntUplinkBitrate
- ULONG grntDownlinkBitrate
- BYTE qosDeliveryOrder
- ULONG maxSDUSize
- BYTE sduErrorRatio
- BYTE resBerRatio
- BYTE deliveryErrSDU
- ULONG transferDelay
- ULONG trafficPriority

### 8.1114.1 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 |

| | |
|---|---|
| *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. |

### 8.1114.2 Field Documentation

#### 8.1114.2.1 BYTE UMTSQoS::deliveryErrSDU

#### 8.1114.2.2 ULONG UMTSQoS::grntDownlinkBitrate

#### 8.1114.2.3 ULONG UMTSQoS::grntUplinkBitrate

#### 8.1114.2.4 ULONG UMTSQoS::maxDownlinkBitrate

#### 8.1114.2.5 ULONG UMTSQoS::maxSDUSize

#### 8.1114.2.6 ULONG UMTSQoS::maxUplinkBitrate

#### 8.1114.2.7 BYTE UMTSQoS::qosDeliveryOrder

#### 8.1114.2.8 BYTE UMTSQoS::resBerRatio

#### 8.1114.2.9 BYTE UMTSQoS::sduErrorRatio

#### 8.1114.2.10 BYTE UMTSQoS::trafficClass

#### 8.1114.2.11 ULONG UMTSQoS::trafficPriority

#### 8.1114.2.12 ULONG UMTSQoS::transferDelay

## 8.1115 UMTSReqQoSSigInd Struct Reference

### Data Fields

- struct UMTSQoS UMTSReqQoS
- BYTE SigInd

### 8.1115.1 Detailed Description

structure contains UMTS requested QoS with Signaling Indication flag

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

**Parameters**

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

### 8.1115.2 Field Documentation

#### 8.1115.2.1 BYTE UMTSReqQoSSigInd::SigInd

#### 8.1115.2.2 struct UMTSQoS UMTSReqQoSSigInd::UMTSReqQoS

## 8.1116 unblockUIMPIN Struct Reference

### Data Fields

- BYTE pinID
- BYTE pukLen
- BYTE pukVal [255]
- BYTE newPINLen
- BYTE newPINVal [255]

### 8.1116.1 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[MAX_PUK_LENGTH]* | • PIN Unlock Key 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. |

### 8.1116.2 Field Documentation

#### 8.1116.2.1 BYTE unblockUIMPIN::newPINLen

#### 8.1116.2.2 BYTE unblockUIMPIN::newPINVal[255]

#### 8.1116.2.3 BYTE unblockUIMPIN::pinID

#### 8.1116.2.4 BYTE unblockUIMPIN::pukLen

#### 8.1116.2.5 BYTE unblockUIMPIN::pukVal[255]

## 8.1117 UniversalTime Struct Reference

### Data Fields

- WORD year
- BYTE month
- BYTE day
- BYTE hour
- BYTE minute
- BYTE second
- BYTE dayOfWeek

### 8.1117.1 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. |

### 8.1117.2 Field Documentation

#### 8.1117.2.1 BYTE UniversalTime::day

#### 8.1117.2.2 BYTE UniversalTime::dayOfWeek

#### 8.1117.2.3 BYTE UniversalTime::hour

#### 8.1117.2.4 BYTE UniversalTime::minute

#### 8.1117.2.5 BYTE UniversalTime::month

#### 8.1117.2.6 BYTE UniversalTime::second

#### 8.1117.2.7 WORD UniversalTime::year

## 8.1118 unpack_dms_GetActivationState_t Struct Reference

### Data Fields

- uint8_t state

### 8.1118.1 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 0xff - Invalid data. |

### 8.1118.2 Field Documentation

#### 8.1118.2.1 uint8_t unpack_dms_GetActivationState_t::state

## 8.1119 unpack_dms_GetBandCapability_t Struct Reference

### Data Fields

- uint32_t BandCapability
- uint16_t Tlvresult

### 8.1119.1 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 |
| *Tlvresult* | • unpack Tlv Result.<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1119.2 Field Documentation

#### 8.1119.2.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

#### 8.1119.2.2 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

## 8.1120 unpack_dms_GetCrashAction_t Struct Reference

### Data Fields

- uint8_t DevCrashState
- uint16_t Tlvresult

### 8.1120.1 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>– 0xff - Invaild data. |
| *Tlvresult* | • unpack Tlv result |

### 8.1120.2 Field Documentation

#### 8.1120.2.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

#### 8.1120.2.2 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

## 8.1121 unpack_dms_GetCustFeature_t Struct Reference

### Data Fields

- uint32_t GpsEnable
- uint8_t DisableIMSI
- uint16_t IPFamSupport
- uint8_t RMAutoConnect
- uint8_t GPSSel
- uint8_t SMSSupport
- uint8_t IsVoiceEnabled
- uint8_t DHCPRelayEnabled
- uint8_t GPSLPM
- uint16_t Tlvresult

### 8.1121.1 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>– 0xff - Invaild data. |
| *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>– 0xff - Invaild data. |
| *IPFamSupport* | • optional 2 byte BitMask<br>• bitmask representing the IP families supported<br>• values:<br>– 0x01 - IPv4<br>– 0x02 - IPv6<br>– 0x04 - IPv4v6<br>– 0xff - Invaild data. |
| *RMAutoConnect* | • optional 1 byte parameter<br>• QMI Mode RM Net Auto Connect Support<br>• values:<br>– 0x00 - Not Supported<br>– 0x01 - Supported<br>– 0xff - Invaild data. |
| *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>– 0xff - Invaild data. |

| | |
|---|---|
| *SMSSupport* | • optional 1 byte parameter<br>• SMS support<br>• values:<br>– 0x00 - Not supported<br>– 0x01 - supported<br>– 0xff - Invaild data.<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>– 0xff - Invaild data. |
| *DHCPRelay-Enabled* | • optional 1 byte parameter<br>• DHCP Relay support<br>• values:<br>– 0x00 - Disable DHCP relay<br>– 0x01 - Enable DHCP relay<br>– 0xff - Invaild data. |
| *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>– 0xff - Invaild data. |
| *Tlvresult* | • unpack Tlv Result. |

### 8.1121.2 Field Documentation

#### 8.1121.2.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

#### 8.1121.2.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

#### 8.1121.2.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

#### 8.1121.2.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

#### 8.1121.2.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

#### 8.1121.2.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

#### 8.1121.2.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

#### 8.1121.2.8 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

#### 8.1121.2.9 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

#### 8.1121.2.10 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

## 8.1122 unpack_dms_GetCustFeaturesV2_t Struct Reference

### Data Fields

- DMSgetCustomFeatureV2 GetCustomFeatureV2
- uint16_t Tlvresult

### 8.1122.1 Detailed Description

This structure contains customization settings set to modem unpack

**Parameters**

| | |
|---|---|
| *GetCustom-FeatureV2* | • See DMSgetCustomFeatureV2 for more information |
| *Tlvresult* | • Unpack Result |

### 8.1122.2 Field Documentation

#### 8.1122.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

#### 8.1122.2.2 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

## 8.1123 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

### 8.1123.1 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 |
| *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 |
| *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 |
| *SimCapability* | Device SIM capability<br>• 0 - SIM not supported<br>• 1 - SIM supported |
| *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 |
| *RadioIfaces* | • Radio interface array. This is a structure of array containing the elements below. uint8_t RadioIfaces<br>– See qaGobiApiTableRadioInterfaces.h for Radio Interfaces |
| *Tlvresult* | • Unpack Result<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1123.2 Field Documentation

#### 8.1123.2.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

#### 8.1123.2.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

#### 8.1123.2.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

#### 8.1123.2.4 uint8_t unpack_dms_GetDeviceCap_t::RadioIfaces[64]

#### 8.1123.2.5 uint32_t unpack_dms_GetDeviceCap_t::RadioIfacesSize

#### 8.1123.2.6 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

#### 8.1123.2.7 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

## 8.1124 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]

## 8.1124.1 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 |
| *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 |
| *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 |
| *simCapability* | • Device SIM capability<br>– 0 - SIM not supported<br>– 1 - SIM supported<br>– 0xff - Invalid Device capablitityes Data. |
| *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 |
| *RadioIfaces[OUT]* | • Radio interface array. This is a structure of array containing the elements below.<br>– See qaGobiApiTableRadioInterfaces.h for Radio Interfaces |

## 8.1124.2 Field Documentation

### 8.1124.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability

### 8.1124.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate

#### 8.1124.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate

#### 8.1124.2.4 uint8_t unpack_dms_GetDeviceCapabilities_t::RadioIfaces[255]

#### 8.1124.2.5 uint32_t unpack_dms_GetDeviceCapabilities_t::radioIfacesSize

#### 8.1124.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

## 8.1125 unpack_dms_GetDeviceHardwareRev_t Struct Reference

### Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult

### 8.1125.1 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 |
| *String* | • NULL terminated Hardware Revision string<br>• Maximum Length is 255 Bytes |
| *Tlvresult* | • Unpack Result<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1125.2 Field Documentation

#### 8.1125.2.1 char unpack_dms_GetDeviceHardwareRev_t::String[255]

#### 8.1125.2.2 uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize

#### 8.1125.2.3 uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult

## 8.1126 unpack_dms_GetDeviceMfr_t Struct Reference

### Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult

### 8.1126.1 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 |
| *String* | • NULL terminated Device Manufacture string<br>• Maximum Length is 255 Bytes |
| *Tlvresult* | • unpack Tlv result<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1126.2 Field Documentation

#### 8.1126.2.1 char unpack_dms_GetDeviceMfr_t::String[255]

#### 8.1126.2.2 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

#### 8.1126.2.3 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

## 8.1127 unpack_dms_GetDeviceSerialNumbers_t Struct Reference

### Data Fields

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

### 8.1127.1 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 |
| *ESNString* | • NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed<br>• Maximum Length is 255 Bytes |

| | |
|---|---|
| *imeiSize* | • The maximum number of characters (including NULL terminator) that the IMEI string array can contain |
| *IMEIString* | • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed<br>• Maximum Length is 255 Bytes |
| *meidSize* | • The maximum number of characters (including NULL terminator) that the MEID string array can contain |
| *MEIDString* | • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed<br>• Maximum Length is 255 Bytes |
| *imeiSvnSize* | • The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain. |
| *ImeiSvnString* | • NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.<br>• Maximum Length is 255 Bytes |
| *Tlvresult* | • unpack Tlv Result. |

### 8.1127.2 Field Documentation

#### 8.1127.2.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

#### 8.1127.2.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

#### 8.1127.2.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

#### 8.1127.2.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

#### 8.1127.2.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

#### 8.1127.2.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

#### 8.1127.2.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

#### 8.1127.2.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

#### 8.1127.2.9 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

## 8.1128 unpack_dms_GetFirmwareInfo_t Struct Reference

### Data Fields

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

#### 8.1128.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareInfo parameters

**Parameters**

| | |
|---|---|
| *modelid_str* | - Mode ID String.<br>• NULL-terminated Mode ID String.<br>• Invalid data : index 0 equal 0x7F |
| *bootversion_str* | - Boot Version.<br>• NULL-terminated Boot Version String.<br>• Invalid data : index 0 equal 0x7F |
| *appversion_str* | - Application Version String.<br>• NULL-terminated Application Version String.<br>• Invalid data : index 0 equal 0x7F |
| *sku_str* | - SKU String.<br>• NULL-terminated SKU String.<br>• Invalid data : index 0 equal 0x7F |
| *packageid_str* | - Package ID String.<br>• NULL-terminated Package ID String.<br>• deprecated on EM/MC74xx(9x30) devices<br>• Invalid data : index 0 equal 0x7F |
| *carrier_str* | - Carrier String.<br>• NULL-terminated Carrier String.<br>• Invalid data : index 0 equal 0x7F |
| *priversion_str* | - PRI Version String.<br>• NULL-terminated PRI Version String.<br>• Invalid data : index 0 equal 0x7F |
| *cur_carr_name* | - Current Carrier Name String.<br>• NULL-terminated Current Carrier Name String.<br>• Invalid data : index 0 equal 0x7F |

| | |
|---|---|
| *cur_carr_rev* | - Current Carrier Revision String.<br>• NULL-terminated Current Carrier Revision String.<br>• Invalid data : index 0 equal 0x7F |
| *Tlvresult* | - unpack Tlv Result. |

### 8.1128.2 Field Documentation

#### 8.1128.2.1 char unpack_dms_GetFirmwareInfo_t::appversion_str[85]

#### 8.1128.2.2 char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]

#### 8.1128.2.3 char unpack_dms_GetFirmwareInfo_t::carrier_str[20]

#### 8.1128.2.4 char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]

#### 8.1128.2.5 char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]

#### 8.1128.2.6 char unpack_dms_GetFirmwareInfo_t::modelid_str[20]

#### 8.1128.2.7 char unpack_dms_GetFirmwareInfo_t::packageid_str[85]

#### 8.1128.2.8 char unpack_dms_GetFirmwareInfo_t::priversion_str[16]

#### 8.1128.2.9 char unpack_dms_GetFirmwareInfo_t::sku_str[15]

#### 8.1128.2.10 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

## 8.1129 unpack_dms_GetFirmwareRevision_t Struct Reference

### Data Fields

- uint8_t amssSize
- char AMSSString [255]
- char PRIString [255]
- uint16_t Tlvresult

### 8.1129.1 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. |
| *AMSSString* | • NULL-terminated AMSS revision string.<br>• Maximum Length is 255 Bytes<br>• Invalid data : index 0 equal 0x7F |

| | |
|---|---|
| *PRIString[OUT]* | • NULL-terminated PRI revision string.<br>• Maximum Length is 255 Bytes<br>• Invalid data : index 0 equal 0x7F |
| *Tlvresult* | • unpack Tlv Result. |

### 8.1129.2 Field Documentation

#### 8.1129.2.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

#### 8.1129.2.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

#### 8.1129.2.3 char unpack_dms_GetFirmwareRevision_t::PRIString[255]

#### 8.1129.2.4 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

## 8.1130 unpack_dms_GetFirmwareRevisions_t Struct Reference

### Data Fields

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

### 8.1130.1 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 |
| *AMSSString* | • NULL-terminated AMSS revision string<br>• Maximum Length is 255 Bytes<br>• Invalid data : index 0 equal 0x7F and amssSize is equal 0. |
| *bootSize* | • Maximum number of characters (including NULL terminator) that the boot string array can contain |

| | |
|---|---|
| *BootString* | • NULL-terminated boot code revision string<br>• Maximum Length is 255 Bytes<br>• Invalid data : index 0 equal 0x7F and bootSize is equal 0. |
| *priSize* | • Maximum number of characters (including NULL terminator) that the PRI string array can contain |
| *PRIString* | • NULL-terminated PRI revision string<br>• Maximum Length is 255 Bytes<br>• Invalid data : index 0 equal 0x7F and priSize is equal 0. |
| *Tlvresult* | • unpack Tlv Result |

### 8.1130.2 Field Documentation

#### 8.1130.2.1 uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize

#### 8.1130.2.2 char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]

#### 8.1130.2.3 uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize

#### 8.1130.2.4 char unpack_dms_GetFirmwareRevisions_t::BootString[255]

#### 8.1130.2.5 uint8_t unpack_dms_GetFirmwareRevisions_t::priSize

#### 8.1130.2.6 char unpack_dms_GetFirmwareRevisions_t::PRIString[255]

#### 8.1130.2.7 uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult

## 8.1131 unpack_dms_GetFSN_t Struct Reference

### Data Fields

- char String [255]
- uint16_t Tlvresult

### 8.1131.1 Detailed Description

This structure used to store Factory Sequence Number parameter

**Parameters**

| | |
|---|---|
| *String* | • Facorty Sequence Number<br>• Maximum Length is 255 Bytes |
| *Tlvresult* | • Unpack Result 8 - eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1131.2 Field Documentation

#### 8.1131.2.1 char unpack_dms_GetFSN_t::String[255]

#### 8.1131.2.2 uint16_t unpack_dms_GetFSN_t::Tlvresult

## 8.1132 unpack_dms_GetHardwareRevision_t Struct Reference

### Data Fields

- char hwVer [255]

### 8.1132.1 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>• Invalid data : index 0 equal 0x7F. |

### 8.1132.2 Field Documentation

#### 8.1132.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

## 8.1133 unpack_dms_GetIMSI_t Struct Reference

### Data Fields

- char imsi [255]
- uint16_t Tlvresult

### 8.1133.1 Detailed Description

This structure used to store unpack_dms_GetIMSI parameters

**Parameters**

| | |
|---|---|
| *imsi* | - IMSI no.<br>• NULL-terminated String. |
| *Tlvresult* | - unpack Tlv Result. |

### 8.1133.2 Field Documentation

#### 8.1133.2.1 char unpack_dms_GetIMSI_t::imsi[255]

#### 8.1133.2.2 uint16_t unpack_dms_GetIMSI_t::Tlvresult

## 8.1134 unpack_dms_GetManufacturer_t Struct Reference

### Data Fields

- char manufacturer [255]
- uint16_t Tlvresult

### 8.1134.1 Detailed Description

This structure is used to store device manufacturer information.

**Parameters**

| | |
|---|---|
| *manufacturer[OUT]* | • NULL terminated string |
| *Tlvresult* | • Unpack Result<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1134.2 Field Documentation

#### 8.1134.2.1 char unpack_dms_GetManufacturer_t::manufacturer[255]

#### 8.1134.2.2 uint16_t unpack_dms_GetManufacturer_t::Tlvresult

## 8.1135 unpack_dms_GetModelID_t Struct Reference

### Data Fields

- char modelid [255]
- uint16_t Tlvresult

### 8.1135.1 Detailed Description

This structure used to store unpack_dms_GetModelID parameters

**Parameters**

| | |
|---|---|
| *modelid* | - Device model id.<br>• NULL-terminated String. |
| *Tlvresult* | - unpack Tlv Result. |

### 8.1135.2 Field Documentation

#### 8.1135.2.1 char unpack_dms_GetModelID_t::modelid[255]

#### 8.1135.2.2 uint16_t unpack_dms_GetModelID_t::Tlvresult

## 8.1136 unpack_dms_GetNetworkTime_t Struct Reference

### Data Fields

- uint16_t source
- uint64_t timestamp
- uint16_t Tlvresult

### 8.1136.1 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 |
| *timestamp* | • Count of 1.25 ms that have elapsed from the start of GPS time (Jan 6, 1980) |
| *Tlvresult* | • Unpack Result<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

**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.

### 8.1136.2 Field Documentation

#### 8.1136.2.1 uint16_t unpack_dms_GetNetworkTime_t::source

#### 8.1136.2.2 uint64_t unpack_dms_GetNetworkTime_t::timestamp

#### 8.1136.2.3 uint16_t unpack_dms_GetNetworkTime_t::Tlvresult

## 8.1137 unpack_dms_GetOfflineReason_t Struct Reference

### Data Fields

- uint32_t * pReasonMask
- uint32_t * pbPlatform
- uint16_t Tlvresult

### 8.1137.1 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>– NULL pointer - Invalid data. |
| *pbPlatform[OUT]* | • Optional parameter<br>• Is the device offline due to a platform restriction?<br>– 0 - No<br>– 1 - Yes<br>– NULL pointer - Invalid data. |
| *Tlvresult* | • Unpack Result |

### 8.1137.2 Field Documentation

#### 8.1137.2.1 uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform

#### 8.1137.2.2 uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask

#### 8.1137.2.3 uint16_t unpack_dms_GetOfflineReason_t::Tlvresult

## 8.1138 unpack_dms_GetPower_t Struct Reference

### Data Fields

- uint32_t OperationMode
- uint32_t OfflineReason
- uint32_t HardwareControlledMode
- uint16_t Tlvresult

### 8.1138.1 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>• 0xff - Invalid data. |

**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>• 0xffff - Invalid data. |
| *Hardware-ControlledMode* | - hardware restricted mode.<br>• 0x00 - FALSE.<br>• 0x01 - TRUE.<br>• 0xFF - Invalid data. |
| *Tlvresult* | - unpack Tlv Result.<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

## 8.1138.2 Field Documentation

### 8.1138.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

### 8.1138.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

### 8.1138.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

### 8.1138.2.4 uint16_t unpack_dms_GetPower_t::Tlvresult

## 8.1139 unpack_dms_GetPRLVersion_t Struct Reference

### Data Fields

- uint8_t u8PRLPreference
- uint16_t u16PRLVersion
- uint16_t Tlvresult

### 8.1139.1 Detailed Description

This structure used to store unpack_dms_GetPRLVersion parameters

**Parameters**

| | |
|---|---|
| *u8PRL-Preference* | • PRL Preference<br>– 0 - Unset<br>– 1 - Set<br>– 0xff invalid data. |
| *u16PRLVersion* | • PRL version of device.<br>• 0xffff invalid data. |
| *Tlvresult* | • unpack Tlv Result |

### 8.1139.2 Field Documentation

#### 8.1139.2.1 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

#### 8.1139.2.2 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

#### 8.1139.2.3 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

## 8.1140 unpack_dms_GetSerialNumbers_t Struct Reference

### Data Fields

- char esn [255]
- char imei_no [255]
- char meid [255]
- char imeisv_svn [255]

### 8.1140.1 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>• Invalid data : index 0 equal 0x7F. |
| *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>• Invalid data : index 0 equal 0x7F. |
| *meid* | - Mobile Equipment Identifier of the device.<br>• NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed.<br>• Invalid data : index 0 equal 0x7F. |
| *imeisv_svn* | - NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.imei software version revision.<br>• Invalid data : index 0 equal 0x7F. |

### 8.1140.2 Field Documentation

#### 8.1140.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

#### 8.1140.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

#### 8.1140.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

#### 8.1140.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

## 8.1141 unpack_dms_GetUSBComp_t Struct Reference

### Data Fields

- uint8_t USBComp [255]
- uint8_t NumSupUSBComps
- uint8_t SupUSBComps
- uint16_t Tlvresult

### 8.1141.1 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 |
| *NumSupUSB-Comps* | • Number of supported USB compositions in the parameter to follow<br>• Range - 0-255<br>• Invalid data : USBComp index 0 equal 0x7F and NumSupUSBComps is equal 0. |

| | |
|---|---|
| *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>– 255 - invalid data. |
| *Tlvresult* | • unpack Tlv Result. |

### 8.1141.2 Field Documentation

#### 8.1141.2.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

#### 8.1141.2.2 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

#### 8.1141.2.3 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

#### 8.1141.2.4 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

## 8.1142 unpack_dms_GetVoiceNumber_t Struct Reference

### Data Fields

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

### 8.1142.1 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. |
| *VoiceNumber* | • Voice number string: MDN or MS ISDN<br>• Maximum Length is 255 Bytes<br>• Invalid data : index 0 equal 0x7F and voiceNumberSize is equal 0. |
| *minSize* | • Maximum number of characters (including NULL terminator) that the MIN array can contain. |
| *MIN* | • Optional Parameter<br>• MIN string: Empty string returned when MIN is not supported/ programmed.<br>• Maximum Length is 255 Bytes<br>• Invalid data : index 0 equal 0x7F and minSize is equal 0. |
| *Tlvresult* | • Unpack Result |

### 8.1142.2 Field Documentation

#### 8.1142.2.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

#### 8.1142.2.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

#### 8.1142.2.3 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

#### 8.1142.2.4 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

#### 8.1142.2.5 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

## 8.1143 unpack_dms_PSMCfgChange_ind_t Struct Reference

### Data Fields

- dms_PSMEnableStateIndTlv EnableState
- dms_PSMActiveTimerIndTlv ActiveTimer
- dms_PSMPeriodicUpdateTimerIndTlv PeriodicUpdateTimer
- uint16_t Tlvresult

### 8.1143.1 Detailed Description

DMS Event Report indication structure

**Parameters**

| | |
|---|---|
| *EnableState* | • See dms_PSMEnableStateIndTlv |
| *ActiveTimer* | • See dms_PSMActiveTimerIndTlv |
| *PeriodicUpdate-Timer* | • See dms_PSMPeriodicUpdateTimerIndTlv |
| *Tlvresult* | • Unpack Result |

### 8.1143.2 Field Documentation

#### 8.1143.2.1 dms_PSMActiveTimerIndTlv unpack_dms_PSMCfgChange_ind_t::ActiveTimer

#### 8.1143.2.2 dms_PSMEnableStateIndTlv unpack_dms_PSMCfgChange_ind_t::EnableState

#### 8.1143.2.3 dms_PSMPeriodicUpdateTimerIndTlv unpack_dms_PSMCfgChange_ind_t::PeriodicUpdateTimer

#### 8.1143.2.4 uint16_t unpack_dms_PSMCfgChange_ind_t::Tlvresult

## 8.1144 unpack_dms_ResetToFactoryDefaults_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1144.1 Detailed Description

This structure contains reset to factory default unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1144.2 Field Documentation

#### 8.1144.2.1 uint16_t unpack_dms_ResetToFactoryDefaults_t::Tlvresult

## 8.1145 unpack_dms_SetActivationStatusCallback_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1145.1 Detailed Description

This structure is used to store Set Service Activation Status callback parameter unpack.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1145.2 Field Documentation

#### 8.1145.2.1 uint16_t unpack_dms_SetActivationStatusCallback_t::Tlvresult

## 8.1146 unpack_dms_SetCrashAction_t Struct Reference

### Data Fields

- uint8_t notused

### 8.1146.1 Detailed Description

This structure is used to store unpack_dms_SetCrashAction parameters

**Parameters**

| | |
|---|---|
| *notused* | • unpack Tlv result |

### 8.1146.2 Field Documentation

#### 8.1146.2.1 uint8_t unpack_dms_SetCrashAction_t::notused

## 8.1147 unpack_dms_SetCustFeature_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1147.1 Detailed Description

This structure is used to store unpack_dms_SetCustFeature parameters

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |

### 8.1147.2 Field Documentation

#### 8.1147.2.1 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

## 8.1148 unpack_dms_SetCustFeaturesV2_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1148.1 Detailed Description

This structure contains customization settings set to modem unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1148.2 Field Documentation

#### 8.1148.2.1 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

## 8.1149 unpack_dms_SetEventReport_ind_t Struct Reference

### Data Fields

- dms_ActivationStatusTlv ActivationStatusTlv
- dms_OperatingModeTlv OperatingModeTlv
- uint16_t Tlvresult

### 8.1149.1 Detailed Description

DMS Event Report indication structure

**Parameters**

| | |
|---|---|
| *ActivationStatus-Tlv* | • See dms_ActivationStatusTlv |
| *OperatingMode-Tlv* | • See dms_OperatingModeTlv |
| *Tlvresult* | • Unpack Result |

### 8.1149.2 Field Documentation

#### 8.1149.2.1 dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv

#### 8.1149.2.2 dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv

#### 8.1149.2.3 uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult

## 8.1150 unpack_dms_SetEventReport_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1150.1 Detailed Description

This structure is used to store unpack_dms_SetEventReport parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |

### 8.1150.2 Field Documentation

#### 8.1150.2.1 uint16_t unpack_dms_SetEventReport_t::Tlvresult

## 8.1151 unpack_dms_SetFirmwarePreference_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1151.1 Detailed Description

This structure is used to store unpack_dms_SetFirmwarePreference parameters

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |

### 8.1151.2 Field Documentation

#### 8.1151.2.1 uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult

## 8.1152 unpack_dms_SetIndicationRegister_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1152.1 Detailed Description

This structure is used to store unpack_dms_SetIndicationRegister parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |

### 8.1152.2 Field Documentation

#### 8.1152.2.1 uint16_t unpack_dms_SetIndicationRegister_t::Tlvresult

## 8.1153 unpack_dms_SetPower_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1153.1 Detailed Description

Structure to store unpack the operating mode of the device.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |

### 8.1153.2 Field Documentation

#### 8.1153.2.1 uint16_t unpack_dms_SetPower_t::Tlvresult

## 8.1154 unpack_dms_SetUSBComp_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1154.1 Detailed Description

This structure is used to store unpack USB composition information

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |

### 8.1154.2 Field Documentation

#### 8.1154.2.1 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

## 8.1155 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

#### 8.1155.1 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 |
| *has_LpmFlag* | LPM Flag Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |
| *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. |
| *has_Wdisable* | W-Disable Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |

| | |
|---|---|
| *Wdisable[-Optional]* | • W_DISABLE<br>• Values<br>– 0 - Switch set to ON position<br>– 1 - Switch set to OFF position |
| *has_PowerOff-Mode* | Power off mode Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |
| *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 |
| *has_Persistent-Lpm* | Persistent LPM Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |
| *PersistentLpm[-Optional]* | • LPM Persistence<br>• Values<br>– 0 - Non-persistent LPM<br>– 1 - Persistent LPM |

### 8.1155.2 Field Documentation

#### 8.1155.2.1 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_LpmFlag

#### 8.1155.2.2 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_PersistentLpm

#### 8.1155.2.3 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_PowerOffMode

#### 8.1155.2.4 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_Wdisable

#### 8.1155.2.5 uint32_t unpack_dms_SLQSDmsSwiGetPCInfo_t::LpmFlag

#### 8.1155.2.6 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::opMode

#### 8.1155.2.7 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::PersistentLpm

#### 8.1155.2.8 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::PowerOffMode

#### 8.1155.2.9 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::Wdisable

## 8.1156 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult

#### 8.1156.1 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 |
| *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 |
| *Tlvresult[OUT]* | • Unpack Result 8 - eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

#### 8.1156.2 Field Documentation

##### 8.1156.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

##### 8.1156.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

##### 8.1156.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

## 8.1157 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult

### 8.1157.1 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>– 255 - invalid data. |
| *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<br>– 255 - invalid data. |
| *Tlvresult* | • Unpack Result<br>• eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1157.2 Field Documentation

#### 8.1157.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

#### 8.1157.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

#### 8.1157.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

## 8.1158 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1158.1 Detailed Description

This structure contains set registration state for different indication unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1158.2 Field Documentation

#### 8.1158.2.1 uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::Tlvresult

## 8.1159 unpack_dms_SLQSGetBandCapability_t Struct Reference

### Data Fields

- uint64_t bandCapability
- int is_LteBandCapability_Available
- uint64_t LteBandCapability
- int is_TdsBandCapability_Available
- uint64_t TdsBandCapability

### 8.1159.1 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>• 0xffffffffffffffff - invalid data |
| *is_LteBand-Capability_-Available[OUT]* | LTE Band Capabilities Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |

| | |
|---|---|
| *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>• Bits 29 through 31 - Reserved<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 |
| *is_TdsBand-Capability_-Available[OUT]* | TDS Band Capabilities Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |

| | |
|---|---|
| *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 |

### 8.1159.2 Field Documentation

#### 8.1159.2.1 uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability

#### 8.1159.2.2 int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available

#### 8.1159.2.3 int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available

#### 8.1159.2.4 uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability

#### 8.1159.2.5 uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability

## 8.1160 unpack_dms_SLQSGetERIFile_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- eriDataparams eriFile

### 8.1160.1 Detailed Description

This structure contains Get ERI file unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result<br>• eQCWWAN_ERR_NULL_TLV - Invalid TLV data. |
| *eriFile* | • Pointer to structure ERIFileparams<br>• See eriDataparams for more information |

### 8.1160.2 Field Documentation

#### 8.1160.2.1 eriDataparams unpack_dms_SLQSGetERIFile_t::eriFile

#### 8.1160.2.2 uint16_t unpack_dms_SLQSGetERIFile_t::Tlvresult

## 8.1161 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

### 8.1161.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

**Parameters**

| | |
|---|---|
| *pPsmEnable-State* | [Optional]<br>• See dms_PSMEnableStateTlv for more information |
| *pDuration-Threshold* | [Optional]<br>• See dms_PSMDurationThresholdTlv for more information |
| *pDurationDueTo-OOS* | [Optional]<br>• See dms_PSMDurationDueToOOSTlv for more information |
| *pRandomization-Window* | [Optional]<br>• See dms_PSMRandomizationWindowTlv for more information |
| *pActiveTimer* | [Optional]<br>• See dms_PSMActiveTimerTlv for more information |
| *pPeriodic-UpdateTimer* | [Optional]<br>• See dms_PSMPeriodicUpdateTimerTlv for more information |
| *pEarlyWakeup-Time* | [Optional]<br>• See dms_PSMEarlyWakeupTimeTlv for more information |

### 8.1161.2 Field Documentation

#### 8.1161.2.1 dms_PSMActiveTimerTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pActiveTimer

#### 8.1161.2.2 dms_PSMDurationDueToOOSTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pDurationDueToOOS

#### 8.1161.2.3 dms_PSMDurationThresholdTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pDurationThreshold

#### 8.1161.2.4 dms_PSMEarlyWakeupTimeTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pEarlyWakeupTime

#### 8.1161.2.5 dms_PSMPeriodicUpdateTimerTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pPeriodicUpdateTimer

#### 8.1161.2.6 dms_PSMEnableStateTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pPsmEnableState

#### 8.1161.2.7 dms_PSMRandomizationWindowTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pRandomization-Window

## 8.1162 unpack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1162.1 Detailed Description

This structure is used to store unpack_dms_SLQSSetPowerSaveModeConfig parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |

### 8.1162.2 Field Documentation

#### 8.1162.2.1 uint16_t unpack_dms_SLQSSetPowerSaveModeConfig_t::Tlvresult

## 8.1163 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1163.1 Detailed Description

This structure contains Clear Dying GASP unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1163.2 Field Documentation

#### 8.1163.2.1 uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult

## 8.1164 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- crashInfoParams crashInfoParam

### 8.1164.1 Detailed Description

This structure contains SWI get crash information unpack information

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *crashInfoParam[-OUT]* | • See crashInfoParams |

### 8.1164.2 Field Documentation

#### 8.1164.2.1 crashInfoParams unpack_dms_SLQSSwiGetCrashInfo_t::crashInfoParam

#### 8.1164.2.2 uint16_t unpack_dms_SLQSSwiGetCrashInfo_t::Tlvresult

## 8.1165 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

### Data Fields

- packgetDyingGaspCfg * pGetDyingGaspCfg
- uint16_t Tlvresult

### 8.1165.1 Detailed Description

This structure contains Get Dying GASP Config unpack

**Parameters**

| | |
|---|---|
| *pGetDyingGasp-Cfg* | • See packgetDyingGaspCfg<br>• NULL pDestSMSNum for invalid data.<br>• NULL pDestSMSContent for invalid data. |
| *Tlvresult* | • Unpack Result |

### 8.1165.2 Field Documentation

#### 8.1165.2.1 packgetDyingGaspCfg* unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg

#### 8.1165.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult

## 8.1166 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference

### Data Fields

- packgetDyingGaspStatistics * pGetDyingGaspStatistics
- uint16_t Tlvresult

### 8.1166.1 Detailed Description

This structure contains Get Dying GASP Statistics unpack.

**Parameters**

| | |
|---|---|
| *pGetDyingGasp-Statistics* | • See packgetDyingGaspStatistics<br>• NULL pTimeStamp for invalid Timestamp data.<br>• NULL pSMSAttemptedFlag for invalid SMS attempted data. |
| *Tlvresult* | • Unpack Result |

### 8.1166.2 Field Documentation

#### 8.1166.2.1 packgetDyingGaspStatistics∗ unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics

#### 8.1166.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult

## 8.1167 unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference

### Data Fields

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

### 8.1167.1 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 |
| *pCurrImgInfo[O-UT]* | • Currently Active Image List |
| *priver[OUT]* | • PRI version of the currently running firmware<br>• Invalid Data - index 0 equal 0x7F. |
| *pkgver[OUT]* | • Package version of the currently running firmware<br>• Invalid Data - index 0 equal 0x7F. |

| | |
|---|---|
| *fwvers[OUT]* | • firmware version of the currently running firmware<br>• Invalid Data - index 0 equal 0x7F. |
| *carrier[OUT]* | • Carrier string of the currently running firmware<br>• Invalid Data - index 0 equal 0x7F. |

### 8.1167.2 Field Documentation

#### 8.1167.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

#### 8.1167.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

#### 8.1167.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

#### 8.1167.2.4 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

#### 8.1167.2.5 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

#### 8.1167.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

## 8.1168 unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference

### Data Fields

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

### 8.1168.1 Detailed Description

This structure is used to store Firmware Update Status

**Parameters**

| | |
|---|---|
| *ResCode* | • FW Update Result Code<br>• Values:<br>– 0x00000001 - Successful<br>– 0x000000FF - Invalid Data.<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 $\sim$ 0FF - Insignia defined error code<br>* 100 $\sim$ 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 |
| *imgType* | • Optional parameter<br>• Firmware image type that failed the update<br>• 0xff - Invalid Data. |
| *refData* | • Optional parameter<br>• Failed image reference data<br>• This is normally the offset of the image that caused the failure<br>• 0xff - Invalid Data. |
| *refString* | • Optional parameter<br>• Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.<br>• Invalid Data - index 0 equal 0xFF. |
| *logString* | • Optional parameter<br>• Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.<br>• Invalid Data - index 0 equal 0xFF. |
| *TlvResult* | • Unpack Result |

## 8.1168.2 Field Documentation

### 8.1168.2.1 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType

### 8.1168.2.2 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]

#### 8.1168.2.3 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData

#### 8.1168.2.4 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]

#### 8.1168.2.5 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode

#### 8.1168.2.6 uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult

## 8.1169 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- char manString [255]
- char modelString [255]
- char swVerString [255]
- char plasmaIDString [255]

### 8.1169.1 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>• Invalid Data - index 0 equal 0x7F. |
| *modelString[OUT]* | • Host Device Model String<br>• Null terminated ASCII String<br>• Invalid Data - index 0 equal 0x7F. |
| *swVerString[OUT]* | • Host Device Software Version String<br>• Null terminated ASCII string<br>• Invalid Data - index 0 equal 0x7F. |
| *plasmaIDString[OUT]* | • Host Device Plasma ID String<br>• Null terminated alphanumeric ASCII String.<br>• Invalid Data - index 0 equal 0x7F. |

### 8.1169.2 Field Documentation

#### 8.1169.2.1 char unpack_dms_SLQSSwiGetHostDevInfo_t::manString[255]

#### 8.1169.2.2 char unpack_dms_SLQSSwiGetHostDevInfo_t::modelString[255]

#### 8.1169.2.3 char unpack_dms_SLQSSwiGetHostDevInfo_t::plasmaIDString[255]

#### 8.1169.2.4 char unpack_dms_SLQSSwiGetHostDevInfo_t::swVerString[255]

#### 8.1169.2.5 uint16_t unpack_dms_SLQSSwiGetHostDevInfo_t::TlvResult

## 8.1170 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference

### Data Fields

- uint16_t TlvResult
- char nameString [255]
- char versionString [255]

### 8.1170.1 Detailed Description

This structure contains SWI get host os info unpack information

**Parameters**

| | |
|---|---|
| *TlvResult* | • Unpack Result |
| *nameString[OUT]* | • Host operating system name<br>• Invalid Data - index 0 equal 0xFF. |
| *VersionString[OUT]* | • Host operating system version<br>• Invalid Data - index 0 equal 0xFF. |

### 8.1170.2 Field Documentation

#### 8.1170.2.1 char unpack_dms_SLQSSwiGetOSInfo_t::nameString[255]

#### 8.1170.2.2 uint16_t unpack_dms_SLQSSwiGetOSInfo_t::TlvResult

#### 8.1170.2.3 char unpack_dms_SLQSSwiGetOSInfo_t::versionString[255]

## 8.1171 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference

### Data Fields

- uint16_t TlvResult
- char meidString [8]

### 8.1171.1 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>• Invalid Data - index 0 equal 0x7F. |

### 8.1171.2 Field Documentation

#### 8.1171.2.1 char unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]

#### 8.1171.2.2 uint16_t unpack_dms_SLQSSwiGetSerialNoExt_t::Tlvresult

## 8.1172 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1172.1 Detailed Description

This structure contains set Dying GASP Config unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1172.2 Field Documentation

#### 8.1172.2.1 uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

## 8.1173 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1173.1 Detailed Description

This structure contains SWI set host dev info unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1173.2 Field Documentation

#### 8.1173.2.1 uint16_t unpack_dms_SLQSSwiSetHostDevInfo_t::Tlvresult

## 8.1174 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1174.1 Detailed Description

This structure contains SWI set host OS info unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.1174.2 Field Documentation

#### 8.1174.2.1 uint16_t unpack_dms_SLQSSwiSetOSInfo_t::Tlvresult

## 8.1175 unpack_dms_SLQSUIMGetState_t Struct Reference

### Data Fields

- uint8_t state
- uint16_t Tlvresult

### 8.1175.1 Detailed Description

This structure contains UIM get state unpack information

**Parameters**

| | |
|---|---|
| *state[OUT]* | • UIM state |
| *Tlvresult* | • Unpack Result |

### 8.1175.2 Field Documentation

#### 8.1175.2.1 uint8_t unpack_dms_SLQSUIMGetState_t::state

#### 8.1175.2.2 uint16_t unpack_dms_SLQSUIMGetState_t::Tlvresult

## 8.1176 unpack_dms_SwiEventReportCallBack_ind_t Struct Reference

### Data Fields

- dms_TemperatureTlv TempTlv
- dms_VoltageTlv VoltTlv

#### 8.1176.1 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 |

**Note**

: None

#### 8.1176.2 Field Documentation

##### 8.1176.2.1 dms_TemperatureTlv unpack_dms_SwiEventReportCallBack_ind_t::TempTlv

##### 8.1176.2.2 dms_VoltageTlv unpack_dms_SwiEventReportCallBack_ind_t::VoltTlv

## 8.1177 unpack_dms_SwiSetEventReport_t Struct Reference

### Data Fields

- uint16_t Tlvresult

#### 8.1177.1 Detailed Description

This structure is used to store unpack_dms_SwiSetEventReport parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |

#### 8.1177.2 Field Documentation

##### 8.1177.2.1 uint16_t unpack_dms_SwiSetEventReport_t::Tlvresult

## 8.1178 unpack_dms_UIMGetControlKeyStatus_t Struct Reference

### Data Fields

- uint8_t facilityState
- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

### 8.1178.1 Detailed Description

This structure contains PIN retries status

**Parameters**

| | |
|---|---|
| *facilityState[OUT]* | • Control key status<br>– 0 - Deactivated<br>– 1 - Activated<br>– 2 - Blocked<br>– 0xFF - Invalid Data. |
| *verifyRetriesLeft[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of retries left, after which PIN will be blocked.<br>• 0xFF - Invalid Data. |
| *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>• 0xFF - Invalid Data. |
| *Tlvresult* | • Unpack Result |

### 8.1178.2 Field Documentation

#### 8.1178.2.1 uint8_t unpack_dms_UIMGetControlKeyStatus_t::facilityState

#### 8.1178.2.2 uint16_t unpack_dms_UIMGetControlKeyStatus_t::Tlvresult

#### 8.1178.2.3 uint8_t unpack_dms_UIMGetControlKeyStatus_t::unblockRetriesLeft

#### 8.1178.2.4 uint8_t unpack_dms_UIMGetControlKeyStatus_t::verifyRetriesLeft

## 8.1179 unpack_dms_UIMGetICCID_t Struct Reference

### Data Fields

- uint8_t stringSize
- uint8_t String [255]
- uint16_t Tlvresult

### 8.1179.1 Detailed Description

This structure is used to store unpack_dms_UIMGetICCID parameters.

**Parameters**

| | |
|---|---|
| *stringSize* | • Size of String. |
| *String* | • ICCID String.<br>• Maximum Length is 255 Bytes |
| *Tlvresult* | • Unack result. 8 - eQCWWAN_ERR_NULL_TLV - Invalid Tlv Data. |

### 8.1179.2 Field Documentation

#### 8.1179.2.1 uint8_t unpack_dms_UIMGetICCID_t::String[255]

#### 8.1179.2.2 uint8_t unpack_dms_UIMGetICCID_t::stringSize

#### 8.1179.2.3 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

## 8.1180 unpack_dms_UIMGetPINStatus_t Struct Reference

### Data Fields

- uint8_t p1Status
- uint8_t p1VerifyRetriesLeft
- uint8_t p1UnblockRetriesLeft
- uint8_t p2Status
- uint8_t p2VerifyRetriesLeft
- uint8_t p2UnblockRetriesLeft
- uint16_t Tlvresult

### 8.1180.1 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>– 0xFF - Invalid data. |

| | |
|---|---|
| *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>• 0xFF - Invalid data. |
| *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>• 0xFF - Invalid data. |
| *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>– 0xFF - Invalid data. |
| *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>• 0xFF - Invalid data. |
| *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>• 0xFF - Invalid data. |
| *Tlvresult* | • Unpack Result |

### 8.1180.2 Field Documentation

#### 8.1180.2.1 uint8_t unpack_dms_UIMGetPINStatus_t::p1Status

#### 8.1180.2.2 uint8_t unpack_dms_UIMGetPINStatus_t::p1UnblockRetriesLeft

#### 8.1180.2.3 uint8_t unpack_dms_UIMGetPINStatus_t::p1VerifyRetriesLeft

#### 8.1180.2.4 uint8_t unpack_dms_UIMGetPINStatus_t::p2Status

#### 8.1180.2.5 uint8_t unpack_dms_UIMGetPINStatus_t::p2UnblockRetriesLeft

#### 8.1180.2.6 uint8_t unpack_dms_UIMGetPINStatus_t::p2VerifyRetriesLeft

#### 8.1180.2.7 uint16_t unpack_dms_UIMGetPINStatus_t::Tlvresult

## 8.1181 unpack_dms_UIMSetControlKeyProtection_t Struct Reference

### Data Fields

- uint8_t verifyRetriesLeft
- uint16_t Tlvresult

### 8.1181.1 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 |
| *Tlvresult* | • Unpack Result |

### 8.1181.2 Field Documentation

#### 8.1181.2.1 uint16_t unpack_dms_UIMSetControlKeyProtection_t::Tlvresult

#### 8.1181.2.2 uint8_t unpack_dms_UIMSetControlKeyProtection_t::verifyRetriesLeft

## 8.1182 unpack_dms_UIMSetPINProtection_t Struct Reference

### Data Fields

- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

### 8.1182.1 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>• 0xFF - Invalid data. |
| *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>• 0xFF - Invalid data. |

| | |
|---|---|
| *TlvresultOUT* | • Unpack Result |

### 8.1182.2 Field Documentation

#### 8.1182.2.1 uint16_t unpack_dms_UIMSetPINProtection_t::Tlvresult

#### 8.1182.2.2 uint8_t unpack_dms_UIMSetPINProtection_t::unblockRetriesLeft

#### 8.1182.2.3 uint8_t unpack_dms_UIMSetPINProtection_t::verifyRetriesLeft

## 8.1183 unpack_dms_UIMUnblockControlKey_t Struct Reference

### Data Fields

- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

### 8.1183.1 Detailed Description

This structure contains UIM Set control key protection unpack information

**Parameters**

| | |
|---|---|
| *unblockRetriesLeftOUT* | • Optional parameter<br>• number of unblock retries left after which the control key is permanently blocked<br>• 0xff - Invalid data. |
| *TlvresultOUT* | • Unpack Result |

### 8.1183.2 Field Documentation

#### 8.1183.2.1 uint16_t unpack_dms_UIMUnblockControlKey_t::Tlvresult

#### 8.1183.2.2 uint8_t unpack_dms_UIMUnblockControlKey_t::unblockRetriesLeft

## 8.1184 unpack_fms_GetImagesPreference_t Struct Reference

### Data Fields

- uint32_t ImageListSize
- FMSPrefImageList * pImageList
- uint16_t Tlvresult

### 8.1184.1 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 |
| *Tlvresult* | • Unpack result |

### 8.1184.2 Field Documentation

#### 8.1184.2.1 uint32_t unpack_fms_GetImagesPreference_t::ImageListSize

#### 8.1184.2.2 FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList

#### 8.1184.2.3 uint16_t unpack_fms_GetImagesPreference_t::Tlvresult

## 8.1185 unpack_fms_GetStoredImages_t Struct Reference

### Data Fields

- uint32_t imagelistSize
- FMSImageList imageList
- uint16_t Tlvresult

### 8.1185.1 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 |
| *Tlvresult* | • Unpack result |

### 8.1185.2 Field Documentation

#### 8.1185.2.1 FMSImageList unpack_fms_GetStoredImages_t::imageList

#### 8.1185.2.2 uint32_t unpack_fms_GetStoredImages_t::imagelistSize

#### 8.1185.2.3 uint16_t unpack_fms_GetStoredImages_t::Tlvresult

## 8.1186 unpack_fms_SetImagesPreference_t Struct Reference

### Data Fields

- uint32_t ImageTypesSize
- uint8_t ImageTypes [255]
- uint16_t Tlvresult

### 8.1186.1 Detailed Description

This structure contains the Set Images Preference unpack

**Parameters**

| | |
|---|---|
| *ImageTypesSize* | • Image Type Size |
| *ImageTypes* | • Image Type |
| *Tlvresult* | • Unpack result |

### 8.1186.2 Field Documentation

#### 8.1186.2.1 uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]

#### 8.1186.2.2 uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize

#### 8.1186.2.3 uint16_t unpack_fms_SetImagesPreference_t::Tlvresult

## 8.1187 unpack_loc_BestAvailPos_Ind_t Struct Reference

### Data Fields

- uint32_t status
- uint32_t * pXid
- uint64_t * pLatitude
- uint64_t * pLongitude
- uint32_t * pHorUncCircular
- uint32_t * pAltitudeWrtEllipsoid
- uint32_t * pVertUnc
- uint64_t * pTimestampUtc
- uint32_t * pTimeUnc
- uint32_t * pHorUncEllipseSemiMinor
- uint32_t * pHorUncEllipseSemiMajor
- uint32_t * pHorUncEllipseOrientAzimuth
- uint8_t * pHorCirConf
- uint8_t * pHorEllpConf
- uint32_t * pHorReliability
- uint32_t * pSpeedHorizontal
- uint32_t * pSpeedUnc

- uint32_t * pAltitudeWrtMeanSeaLevel
- uint8_t * pVertConfidence
- uint32_t * pVertReliability
- uint32_t * pSpeedVertical
- uint32_t * pSpeedVerticalUnc
- uint32_t * pHeading
- uint32_t * pHeadingUnc
- uint32_t * pMagneticDeviation
- uint32_t * pTechnologyMask
- loc_precisionDilution * pPrecisionDilution
- loc_gpsTime * pGpsTime
- uint32_t * pTimeSrc
- loc_sensorDataUsage * pSensorDataUsage
- loc_svUsedforFix * pSvUsedforFix
- uint16_t Tlvresult

### 8.1187.1 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 |
| *xid* | Transaction ID that was specified in the Get Best Available Position request. |
| *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 |

| | |
|---|---|
| *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 |
| *pHorUncCircular* | • Horizontal position uncertainty.<br>• Units - Meters |
| *pAltitudeWrt-Ellipsoid* | • Altitude With Respect to WGS84 Ellipsoid.<br>• Units - Meters<br>• Range -500 to 15883 |
| *pVertUnc* | • Vertical uncertainty.<br>• Units - Meters |
| *pTimestampUtc* | • UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970 |
| *pTimeUnc* | • Time uncertainty.<br>• Units - Milliseconds |
| *pHorUncEllipse-SemiMinor* | • Semi-minor axis of horizontal elliptical uncertainty.<br>• Units - Meters |
| *pHorUncEllipse-SemiMajor* | • Semi-major axis of horizontal elliptical uncertainty.<br>• Units: Meters |
| *pHorUncEllipse-OrientAzimuth* | • Elliptical horizontal uncertainty azimuth of orientation.<br>• Units - Decimal degrees<br>• Range - 0 to 180 |
| *pHorCirConf* | • Horizontal circular uncertainty confidence<br>• Units: Precent<br>• Range: 0 to 99 |
| *pHorEllpConf* | • Horizontal elliptical uncertainty confidence<br>• Units: Precent<br>• Range: 0 to 99 |

| | |
|---|---|
| *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 |
| *pSpeed-Horizontal* | • Horizontal speed.<br>• Units - Meters/second |
| *pSpeedUnc* | • 3-D Speed uncertainty.<br>• Units - Meters/second. |
| *pAltitudeWrt-MeanSeaLevel* | • Altitude With Respect to Sea Level.<br>• Units - Meters |
| *pVertConfidence* | • Vertical uncertainty confidence.<br>• Units - Percentage<br>• Range 0 to 99 |
| *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 |
| *pSpeedVertical* | • Vertical speed.<br>• Units - Meters/second |
| *pSpeedVertical-Unc* | • Veritical speed<br>• Units: Meters/second |
| *pHeading* | • Heading.<br>• Units - Degree<br>• Range 0 to 359.999 |
| *pHeadingUnc* | • Heading uncertainty.<br>• Units - Degree<br>• Range 0 to 359.999 |

| | |
|---|---|
| *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. |
| *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 |
| *-pPrecision-Dilution* | • See loc_precisionDilution for more information |
| *-pGpsTime* | • See loc_gpsTime for more information |
| *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. |
| *-pSensorData-Usage* | • See loc_sensorDataUsage for more information |
| *-pSvUsedforFix* | • See loc_svUsedforFix for more information |

### 8.1187.2 Field Documentation

#### 8.1187.2.1 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid

#### 8.1187.2.2 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel

#### 8.1187.2.3 loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime

#### 8.1187.2.4 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading

#### 8.1187.2.5 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc

#### 8.1187.2.6 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf

#### 8.1187.2.7 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf

#### 8.1187.2.8 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability

#### 8.1187.2.9 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular

#### 8.1187.2.10 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth

#### 8.1187.2.11 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor

#### 8.1187.2.12 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor

#### 8.1187.2.13 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude

#### 8.1187.2.14 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude

#### 8.1187.2.15 uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation

#### 8.1187.2.16 loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution

#### 8.1187.2.17 loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage

#### 8.1187.2.18 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal

#### 8.1187.2.19 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc

#### 8.1187.2.20 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical

#### 8.1187.2.21 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc

#### 8.1187.2.22 loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix

#### 8.1187.2.23 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask

#### 8.1187.2.24 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc

#### 8.1187.2.25 uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc

#### 8.1187.2.26 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc

#### 8.1187.2.27 uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence

#### 8.1187.2.28 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability

#### 8.1187.2.29 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc

#### 8.1187.2.30 uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid

#### 8.1187.2.31 uint32_t unpack_loc_BestAvailPos_Ind_t::status

#### 8.1187.2.32 uint16_t unpack_loc_BestAvailPos_Ind_t::Tlvresult

## 8.1188 unpack_loc_CradleMountCallback_Ind_t Struct Reference

### Data Fields

- uint32_t cradleMountConfigStatus

### 8.1188.1 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 |

### 8.1188.2 Field Documentation

#### 8.1188.2.1 uint32_t unpack_loc_CradleMountCallback_Ind_t::cradleMountConfigStatus

## 8.1189 unpack_loc_Delete_Assist_Data_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1189.1 Detailed Description

This structure contains LOC delete assist data unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result. |

### 8.1189.2 Field Documentation

#### 8.1189.2.1 uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult

## 8.1190 unpack_loc_DeleteAssistData_Ind_t Struct Reference

### Data Fields

- uint32_t status
- uint16_t Tlvresult

### 8.1190.1 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>– 0xffffffff - Invalid data. |
| *Tlvresult* | • unpack result |

### 8.1190.2 Field Documentation

#### 8.1190.2.1 uint32_t unpack_loc_DeleteAssistData_Ind_t::status

#### 8.1190.2.2 uint16_t unpack_loc_DeleteAssistData_Ind_t::Tlvresult

## 8.1191 unpack_loc_EngineState_Ind_t Struct Reference

### Data Fields

- uint32_t engineState
- uint16_t Tlvresult

### 8.1191.1 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>• 0xffffffff - Invalid data. |
| *Tlvresult* | • unpack result |

### 8.1191.2 Field Documentation

#### 8.1191.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

#### 8.1191.2.2 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

## 8.1192 unpack_loc_EventNMEA_Ind_t Struct Reference

### Data Fields

- char NMEAData [201]
- uint16_t Tlvresult

### 8.1192.1 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 |
| *Tlvresult* | • unpack result |

### 8.1192.2 Field Documentation

#### 8.1192.2.1 char unpack_loc_EventNMEA_Ind_t::NMEAData[201]

#### 8.1192.2.2 uint16_t unpack_loc_EventNMEA_Ind_t::TlvResult

## 8.1193 unpack_loc_EventRegister_t Struct Reference

### Data Fields

- uint16_t TlvResult

### 8.1193.1 Detailed Description

This structure contains Event Register unpack

**Parameters**

| | |
|---|---|
| *TlvResult* | • Unpack result. |

### 8.1193.2 Field Documentation

#### 8.1193.2.1 uint16_t unpack_loc_EventRegister_t::TlvResult

## 8.1194 unpack_loc_EventTimeSyncCallback_Ind_t Struct Reference

### Data Fields

- uint32_t timeSyncRefCounter

### 8.1194.1 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. |

### 8.1194.2 Field Documentation

#### 8.1194.2.1 uint32_t unpack_loc_EventTimeSyncCallback_Ind_t::timeSyncRefCounter

## 8.1195 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

### 8.1195.1 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>– 0xffffffff - Invalid data. |
| *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>– 0xffffffff - Invalid data. |
| *pIPv4AddrInfo* | • See loc_IPv4Info for more information. |
| *pIPv6AddrInfo* | • See loc_IPv6Info for more information. |
| *pURL* | • See loc_urlAddr for more information. |

### 8.1195.2 Field Documentation

#### 8.1195.2.1 loc_IPv4Info* unpack_loc_GetServer_Ind_t::pIPv4AddrInfo

#### 8.1195.2.2 loc_IPv6Info* unpack_loc_GetServer_Ind_t::pIPv6AddrInfo

#### 8.1195.2.3 loc_urlAddr* unpack_loc_GetServer_Ind_t::pURL

#### 8.1195.2.4 uint32_t unpack_loc_GetServer_Ind_t::serverStatus

#### 8.1195.2.5 uint32_t unpack_loc_GetServer_Ind_t::serverType

#### 8.1195.2.6 uint16_t unpack_loc_GetServer_Ind_t::TlvResult

## 8.1196 unpack_loc_GnssSvInfo_Ind_t Struct Reference

### Data Fields

- uint8_t altitudeAssumed
- loc_satelliteInfo * pSatelliteInfo
- uint16_t TlvResult

### 8.1196.1 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>– 0xff - Invalid data. |
| *pSatelliteInfo* | • See loc_satelliteInfo for more information.<br>• NULL pointer - Invalid data. |
| *TlvResult* | • unpack result |

### 8.1196.2 Field Documentation

#### 8.1196.2.1 uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed

#### 8.1196.2.2 loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo

#### 8.1196.2.3 uint16_t unpack_loc_GnssSvInfo_Ind_t::TlvResult

## 8.1197 unpack_loc_InjectPositionCallback_Ind_t Struct Reference

### Data Fields

- uint32_t status

### 8.1197.1 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 |

**Note**

None

### 8.1197.2 Field Documentation

#### 8.1197.2.1 uint32_t unpack_loc_InjectPositionCallback_Ind_t::status

## 8.1198 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

### 8.1198.1 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 |
| *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 |
| *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 |
| *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 |
| *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 |
| *pGyroTempSamplesAccepted[OPTIONAL]* | • 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 |

### 8.1198.2 Field Documentation

#### 8.1198.2.1 uint32_t unpack_loc_InjectSensorDataCallback_Ind_t::injectSensorDataStatus

#### 8.1198.2.2 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pAccelSamplesAccepted

#### 8.1198.2.3 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pAccelTempSamplesAccepted

#### 8.1198.2.4 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pGyroSamplesAccepted

#### 8.1198.2.5 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pGyroTempSamplesAccepted

#### 8.1198.2.6 uint32_t* unpack_loc_InjectSensorDataCallback_Ind_t::pOpaqueIdentifier

## 8.1199 unpack_loc_InjectTimeSyncDataCallback_Ind_t Struct Reference

### Data Fields

- uint32_t injectTimeSyncStatus

### 8.1199.1 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 |

### 8.1199.2 Field Documentation

#### 8.1199.2.1 uint32_t unpack_loc_InjectTimeSyncDataCallback_Ind_t::injectTimeSyncStatus

## 8.1200 unpack_loc_InjectUTCTimeCallback_Ind_t Struct Reference

### Data Fields

- uint32_t status

### 8.1200.1 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 |

**Note**

None

### 8.1200.2 Field Documentation

#### 8.1200.2.1 uint32_t unpack_loc_InjectUTCTimeCallback_Ind_t::status

## 8.1201 unpack_loc_PositionRpt_Ind_t Struct Reference

### Data Fields

- uint32_t sessionStatus
- uint8_t sessionId
- uint64_t * pLatitude
- uint64_t * pLongitude
- uint32_t * pHorUncCircular
- uint32_t * pHorUncEllipseSemiMinor
- uint32_t * pHorUncEllipseSemiMajor
- uint32_t * pHorUncEllipseOrientAzimuth
- uint8_t * pHorConfidence
- uint32_t * pHorReliability
- uint32_t * pSpeedHorizontal
- uint32_t * pSpeedUnc
- uint32_t * pAltitudeWrtEllipsoid
- uint32_t * pAltitudeWrtMeanSeaLevel
- uint32_t * pVertUnc
- uint8_t * pVertConfidence
- uint32_t * pVertReliability
- uint32_t * pSpeedVertical
- uint32_t * pHeading
- uint32_t * pHeadingUnc
- uint32_t * pMagneticDeviation
- uint32_t * pTechnologyMask
- loc_precisionDilution * pPrecisionDilution

- uint64_t * pTimestampUtc
- uint8_t * pLeapSeconds
- loc_gpsTime * pGpsTime
- uint32_t * pTimeUnc
- uint32_t * pTimeSrc
- loc_sensorDataUsage * pSensorDataUsage
- uint32_t * pFixId
- loc_svUsedforFix * pSvUsedforFix
- uint8_t * pAltitudeAssumed
- uint16_t Tlvresult

### 8.1201.1 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>– 0xffffffff - Invalid data. |
| *sessionId* | • ID of the session that was specified in the Start request<br>• Range - 0 to 255 |
| *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>• NULL pointer - Invalid data. |
| *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>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pHorUncCircular* | • Horizontal position uncertainty.<br>• Units - Meters<br>• NULL pointer - Invalid data. |
| *pHorUncEllipse-SemiMinor* | • Semi-minor axis of horizontal elliptical uncertainty.<br>• Units - Meters<br>• NULL pointer - Invalid data. |
| *pHorUncEllipse-SemiMajor* | • Semi-major axis of horizontal elliptical uncertainty.<br>• Units: Meters<br>• NULL pointer - Invalid data. |
| *pHorUncEllipse-OrientAzimuth* | • Elliptical horizontal uncertainty azimuth of orientation.<br>• Units - Decimal degrees<br>• Range - 0 to 180<br>• NULL pointer - Invalid data. |
| *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>• NULL pointer - Invalid data. |
| *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>– NULL pointer - Invalid data. |
| *pSpeed-Horizontal* | • Horizontal speed.<br>• Units - Meters/second<br>• NULL pointer - Invalid data. |
| *pSpeedUnc* | • 3-D Speed uncertainty.<br>• Units - Meters/second.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pAltitudeWrt-Ellipsoid* | • Altitude With Respect to WGS84 Ellipsoid.<br>• Units - Meters<br>• Range -500 to 15883<br>• NULL pointer - Invalid data. |
| *pAltitudeWrt-MeanSeaLevel* | • Altitude With Respect to Sea Level.<br>• Units - Meters<br>• NULL pointer - Invalid data. |
| *pVertUnc* | • Vertical uncertainty.<br>• Units - Meters<br>• NULL pointer - Invalid data. |
| *pVertConfidence* | • Vertical uncertainty confidence.<br>• Units - Percentage<br>• Range 0 to 99<br>• NULL pointer - Invalid data. |
| *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>– NULL pointer - Invalid data. |
| *pSpeedVertical* | • Vertical speed.<br>• Units - Meters/second<br>• NULL pointer - Invalid data. |
| *pHeading* | • Heading.<br>• Units - Degree<br>• Range 0 to 359.999<br>• NULL pointer - Invalid data. |
| *pHeadingUnc* | • Heading uncertainty.<br>• Units - Degree<br>• Range 0 to 359.999<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *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>• NULL pointer - Invalid data. |
| *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>– NULL pointer - Invalid data. |
| *-pPrecision-Dilution* | • See loc_precisionDilution for more information<br>• NULL pointer - Invalid data. |
| *pTimestampUtc* | • UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970<br>• NULL pointer - Invalid data. |
| *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>• NULL pointer - Invalid data. |
| *-pGpsTime* | • See loc_gpsTime for more information<br>• NULL pointer - Invalid data. |
| *pTimeUnc* | • Time uncertainty.<br>• Units - Milliseconds<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *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>– NULL pointer - Invalid data. |
| *-pSensorData-Usage* | • See loc_sensorDataUsage for more information<br>– NULL pointer - Invalid data. |
| *pFixId* | • Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session.<br>• NULL pointer - Invalid data. |
| *-pSvUsedforFix* | • See loc_svUsedforFix for more information<br>– NULL pointer - Invalid data. |
| *pAltitude-Assumed* | • Indicates whether altitude is assumed or calculated. |

- Value
  - 0x00 - Altitude is calculated
  - 0x01 - Altitude is assumed
  - NULL pointer - Invalid data.

## 8.1201.2 Field Documentation

### 8.1201.2.1 uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed

### 8.1201.2.2 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid

8.1201.2.3 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel

8.1201.2.4 uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId

8.1201.2.5 loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime

8.1201.2.6 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading

8.1201.2.7 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc

8.1201.2.8 uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence

8.1201.2.9 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability

8.1201.2.10 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular

8.1201.2.11 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth

8.1201.2.12 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor

8.1201.2.13 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor

8.1201.2.14 uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude

8.1201.2.15 uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds

8.1201.2.16 uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude

8.1201.2.17 uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation

8.1201.2.18 loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution

8.1201.2.19 loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage

8.1201.2.20 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal

8.1201.2.21 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc

8.1201.2.22 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical

8.1201.2.23 loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix

8.1201.2.24 uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask

8.1201.2.25 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc

8.1201.2.26 uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc

8.1201.2.27 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc

8.1201.2.28 uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence

8.1201.2.29 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability

8.1201.2.30 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc

#### 8.1201.2.31 uint8_t unpack_loc_PositionRpt_Ind_t::sessionId

#### 8.1201.2.32 uint32_t unpack_loc_PositionRpt_Ind_t::sessionStatus

#### 8.1201.2.33 uint16_t unpack_loc_PositionRpt_Ind_t::Tlvresult

## 8.1202 unpack_loc_SensorStreamingCallback_Ind_t Struct Reference

### Data Fields

- loc_accelAcceptReady * pAccelAcceptReady
- loc_gyroAcceptReady * pGyroAcceptReady
- loc_accelTempAcceptReady * pAccelTempAcceptReady
- loc_gyroTempAcceptReady * pGyroTempAcceptReady

### 8.1202.1 Detailed Description

This structure contains LOC Event Sensor Streaming Ready Status

**Parameters**

| | |
|---|---|
| *-pAccelAcceptReady[OPTIONAL]* | • See loc_accelAcceptReady for more information<br>• NULL when TLV is absent |
| *-pGyroAcceptReady[OPTIONAL]* | • See loc_gyroAcceptReady for more information<br>• NULL when TLV is absent |
| *-pAccelTempAcceptReady[OPTIONAL]* | • See loc_accelTempAcceptReady for more information<br>• NULL when TLV is absent |
| *-pGyroTempAcceptReady[OPTIONAL]* | • See loc_gyroTempAcceptReady for more information<br>• NULL when TLV is absent |

### 8.1202.2 Field Documentation

#### 8.1202.2.1 loc_accelAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pAccelAcceptReady

#### 8.1202.2.2 loc_accelTempAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pAccelTempAcceptReady

#### 8.1202.2.3 loc_gyroAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pGyroAcceptReady

#### 8.1202.2.4 loc_gyroTempAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pGyroTempAcceptReady

## 8.1203 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

### Data Fields

- uint32_t status

- uint16_t Tlvresult

### 8.1203.1 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 - 0xffffffff - Invalid data. |
| *Tlvresult* | • unpack result |

### 8.1203.2 Field Documentation

#### 8.1203.2.1 uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status

#### 8.1203.2.2 uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult

## 8.1204 unpack_loc_SetExtPowerState_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1204.1 Detailed Description

This structure contains Set Ext Power State unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result. |

### 8.1204.2 Field Documentation

#### 8.1204.2.1 uint16_t unpack_loc_SetExtPowerState_t::Tlvresult

## 8.1205 unpack_loc_SetOperationMode_Ind_t Struct Reference

### Data Fields

- uint32_t status
- uint16_t Tlvresult

### 8.1205.1 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>– 0xffffffff - Invalid data. |
| *Tlvresult* | • unpack result |

### 8.1205.2 Field Documentation

#### 8.1205.2.1 uint32_t unpack_loc_SetOperationMode_Ind_t::status

#### 8.1205.2.2 uint16_t unpack_loc_SetOperationMode_Ind_t::Tlvresult

## 8.1206 unpack_loc_SetOperationMode_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1206.1 Detailed Description

This structure contains Set Operation Mode unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result. |

### 8.1206.2 Field Documentation

#### 8.1206.2.1 uint16_t unpack_loc_SetOperationMode_t::Tlvresult

## 8.1207 unpack_loc_SetServer_Ind_t Struct Reference

### Data Fields

- uint32_t serverStatus
- uint16_t Tlvresult

### 8.1207.1 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. |

### 8.1207.2 Field Documentation

#### 8.1207.2.1 uint32_t unpack_loc_SetServer_Ind_t::serverStatus

#### 8.1207.2.2 uint16_t unpack_loc_SetServer_Ind_t::Tlvresult

## 8.1208 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1208.1 Detailed Description

This structure contains Set Operation Mode unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result. |

### 8.1208.2 Field Documentation

#### 8.1208.2.1 uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

## 8.1209 unpack_loc_Start_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1209.1 Detailed Description

This structure contains Start LOC unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result. |

### 8.1209.2 Field Documentation

#### 8.1209.2.1 uint16_t unpack_loc_Start_t::Tlvresult

## 8.1210 unpack_loc_Stop_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1210.1 Detailed Description

This structure contains Stop LOC unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result. |

### 8.1210.2 Field Documentation

#### 8.1210.2.1 uint16_t unpack_loc_Stop_t::Tlvresult

## 8.1211 unpack_nas_GetCDMANetworkParameters_t Struct Reference

### Data Fields

- uint8_t SCI
- uint8_t SCM
- uint8_t RegHomeSID
- uint8_t RegForeignSID
- uint8_t RegForeignNID
- uint8_t ForceRev0
- uint8_t CustomSCP
- uint32_t Protocol
- uint32_t Broadcast
- uint32_t Application
- uint32_t Roaming

### 8.1211.1 Detailed Description

Gets the current CDMA network parameters

**Parameters**

| | |
|---|---|
| *SCI* | • Slot cycle index<br>– 0xFF-Unknown |
| *SCM* | • Station class mark<br>– 0xFF-Unknown |
| *RegHomeSID* | • Register on home SID<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *RegForeignSID* | • Register on foreign SID<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |

| | |
|---|---|
| *RegForeignNID* | • Register on foreign NID<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *ForceRev0* | • Force CDMA 1x-EV-DO Rev. 0 mode<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *CustomSCP* | • Use a custom config for CDMA 1x-EV-DO SCP<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *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>– 0xFFFFFFFF - Unknown |
| *Broadcast* | • Broadcast mask for custom SCP config<br>– 0x00000001 - Generic broadcast enabled<br>– 0xFFFFFFFF - Unknown |
| *Application* | • Application mask for custom SCP config<br>– 0x00000001 - SN Multiflow Packet Application<br>– 0x00000002 - SN Enhanced Multiflow Packet Application<br>– 0xFFFFFFFF - Unknown |
| *Roaming* | • Roaming preference<br>– 0 - Automatic<br>– 1 - Home Only<br>– 2 - Affiliated Roaming Only<br>– 3 - Home and Affiliated Roaming<br>– 0xFFFFFFFF - Unknown |

### 8.1211.2 Field Documentation

#### 8.1211.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

#### 8.1211.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

#### 8.1211.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

#### 8.1211.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

#### 8.1211.2.5 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

#### 8.1211.2.6 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

#### 8.1211.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

#### 8.1211.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

#### 8.1211.2.9 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

#### 8.1211.2.10 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

#### 8.1211.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

## 8.1212 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

### 8.1212.1 Detailed Description

Structure for HomeNetwork3GPP2 unpack.

**Parameters**

| | |
|---|---|
| *pMCC* | • Mobile country code (UMTS only).<br>• NULL for invalid data |
| *pMNC* | • Mobile network code (UMTS only).<br>• NULL for invalid data |

| | |
|---|---|
| *nameSize* | • Maximum number of characters (including NULL terminator) that network name array can contain (UMTS only). |
| *pName* | • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).<br>• NULL for invalid data |
| *pSID* | • Home network system ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000<br>• NULL for invalid data |
| *pNID* | • Home network ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000<br>• NULL for invalid data |
| *pNw2MCC* | • Mobile country code (3GPP2 only).<br>• Range : 0 to 999<br>• NULL for invalid data |
| *pNw2MNC* | • Mobile network code (3GPP2 only).<br>• Range : 0 to 999<br>• NULL for invalid data |
| *pNw2DescDisp* | • Network Name Display (3GPP2 only). -Valid Value<br>– 0x00 - Do not display<br>– 0x01 - Display<br>– 0xFF - Unknown<br>• NULL for invalid data |
| *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>• NULL for invalid data |
| *pNw2DescLen* | • Network Description Length (3GPP2 only).<br>• NULL for invalid data |

| | |
|---|---|
| *pNw2Name* | • Network Name (3GPP2 only).<br>• NULL for invalid data |

### 8.1212.2 Field Documentation

#### 8.1212.2.1 uint8_t unpack_nas_GetHomeNetwork3GPP2_t::nameSize

#### 8.1212.2.2 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMCC

#### 8.1212.2.3 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMNC

#### 8.1212.2.4 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pName

#### 8.1212.2.5 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNID

#### 8.1212.2.6 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescDisp

#### 8.1212.2.7 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescEnc

#### 8.1212.2.8 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescLen

#### 8.1212.2.9 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MCC

#### 8.1212.2.10 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MNC

#### 8.1212.2.11 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2Name

#### 8.1212.2.12 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pSID

## 8.1213 unpack_nas_GetHomeNetwork_t Struct Reference

### Data Fields

- uint16_t mcc
- uint16_t mnc
- char name [255]
- uint16_t sid
- uint16_t nid

### 8.1213.1 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>• 0xFFFF - Invalid data. |

| | |
|---|---|
| *mnc* | • Mobile network code (UMTS only).<br>• A 16-bit integer representation of MNC. Range: 0 to 999.<br>• 0xFFFF - Invalid data. |
| *name* | • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only). |
| *sid* | • Home network system ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000 |
| *nid* | • Home network ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000 |

### 8.1213.2 Field Documentation

#### 8.1213.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

#### 8.1213.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

#### 8.1213.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

#### 8.1213.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

#### 8.1213.2.5 uint16_t unpack_nas_GetHomeNetwork_t::sid

## 8.1214 unpack_nas_GetNetworkPreference_t Struct Reference

### Data Fields

- uint32_t ActiveTechPref
- uint32_t Duration
- uint32_t PersistentTechPref
- uint16_t Tlvresult

### 8.1214.1 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>– 0xffffffff - invalid data. |
| *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>– 0xffffffff - invalid data. |
| *Persistent-TechnologyPref[-OUT]* | • Bit field representing persistent radio technology preference<br>– Same representation as the pTechnologyPref parameter<br>• 0xffffffff - invalid data. |
| *Tlvresult* | • unpack result |

### 8.1214.2 Field Documentation

#### 8.1214.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

#### 8.1214.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

#### 8.1214.2.3 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

#### 8.1214.2.4 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

## 8.1215 unpack_nas_GetRFInfo_t Struct Reference

### Data Fields

- uint8_t instancesSize
- RFBandInfoElements RFBandInfoElements [255]

### 8.1215.1 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 |

### 8.1215.2 Field Documentation

#### 8.1215.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

#### 8.1215.2.2 RFBandInfoElements unpack_nas_GetRFInfo_t::RFBandInfoElements[255]

## 8.1216 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]

### 8.1216.1 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 |

| | |
|---|---|
| *CSDomain* | • Circuit switch domain status:<br>– 0 - Unknown/Not Applicable<br>– 1 - Attached<br>– 2 - Detached |
| *PSDomain* | • Packet switch domain status<br>– 0 - Unknown/Not Applicable<br>– 1 - Attached<br>– 2 - Detached |
| *RAN* | • Type of radio access network on which mobile is registered:<br>– 0 - Unknown<br>– 1 - cdma2000 network<br>– 2 - UMTS network |
| *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. |
| *RadioIfaces* | • An array of Radio Interface Technology<br>– See qaGobiApiTableRadioInterfaces.h for the Radio Interface Technologies |
| *Roaming* | • Roaming indicator<br>• 0xffffffff - invaild data. |
| *MCC* | • Mobile country code<br>• 0xffff - invalid dta. |
| *MNC* | • Mobile network code<br>• 0xffff - invalid dta. |
| *nameSize* | • Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks |
| *Name* | • Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks |
| *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. |

| | |
|---|---|
| *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 |

### 8.1216.2 Field Documentation

#### 8.1216.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

#### 8.1216.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

#### 8.1216.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

#### 8.1216.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

#### 8.1216.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

#### 8.1216.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

#### 8.1216.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

#### 8.1216.2.8 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

#### 8.1216.2.9 uint8_t unpack_nas_GetServingNetwork_t::RadioIfaces[255]

#### 8.1216.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadioIfacesSize

#### 8.1216.2.11 uint32_t unpack_nas_GetServingNetwork_t::RAN

#### 8.1216.2.12 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

#### 8.1216.2.13 uint32_t unpack_nas_GetServingNetwork_t::Roaming

## 8.1217 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

### Data Fields

- uint8_t DataCapsLen
- uint8_t DataCaps [255]

### 8.1217.1 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. |
| *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 |

### 8.1217.2 Field Documentation

#### 8.1217.2.1 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]

#### 8.1217.2.2 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen

## 8.1218 unpack_nas_GetSignalStrengths_t Struct Reference

### Data Fields

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

### 8.1218.1 Detailed Description

This structure contains unpack get signal strengths parameters.

**Parameters**

| | |
|---|---|
| *len* | number of rssi & radio items following |
| *rssi* | signal strength array |
| *radio* | radio interface array |

### 8.1218.2 Field Documentation

#### 8.1218.2.1 uint32_t unpack_nas_GetSignalStrengths_t::len

#### 8.1218.2.2 uint32_t unpack_nas_GetSignalStrengths_t::radio[8]

#### 8.1218.2.3 signed char unpack_nas_GetSignalStrengths_t::rssi[8]

## 8.1219 unpack_nas_PerformNetworkScan_t Struct Reference

### Data Fields

- uint8_t * p3GppNetworkInstanceSize
- nas_QmiNas3GppNetworkInfo * p3GppNetworkInfoInstances
- uint8_t * pRATInstanceSize
- nas_QmiNas3GppNetworkRAT * pRATINstance
- uint8_t * pPCSInstanceSize
- nas_QmisNasPcsDigit * pPCSInstance
- uint32_t * pScanResult

### 8.1219.1 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. |
| *p3GppNetwork-InfoInstances* | • Network info instance array<br>– See nas_QmiNas3GppNetworkInfo for more information |
| *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. |
| *pRATINstance* | • RAT info instance array<br>– See nas_QmiNas3GppNetworkRAT for more information |
| *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. |
| *pPCSInstance* | • PCS Digit info instance array<br>– See nas_QmisNasPcsDigit for more information |
| *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>• 0xffffffff - invalid data. |

### 8.1219.2 Field Documentation

#### 8.1219.2.1 nas_QmiNas3GppNetworkInfo* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances

#### 8.1219.2.2 uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize

#### 8.1219.2.3 nas_QmisNasPcsDigit* unpack_nas_PerformNetworkScan_t::pPCSInstance

#### 8.1219.2.4 uint8_t* unpack_nas_PerformNetworkScan_t::pPCSInstanceSize

#### 8.1219.2.5 nas_QmiNas3GppNetworkRAT* unpack_nas_PerformNetworkScan_t::pRATInstance

#### 8.1219.2.6 uint8_t* unpack_nas_PerformNetworkScan_t::pRATInstanceSize

#### 8.1219.2.7 uint32_t* unpack_nas_PerformNetworkScan_t::pScanResult

## 8.1220 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference

### Data Fields

- uint8_t dataCapsSize
- uint8_t dataCaps [255]

### 8.1220.1 Detailed Description

This structure contains unpack set data capabilities callback indication parameters.

**Parameters**

| | |
|---|---|
| *dataCapsSize* | Number of Data Capabilities |
| *dataCaps* | Data Capabilities |

### 8.1220.2 Field Documentation

#### 8.1220.2.1 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]

#### 8.1220.2.2 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize

## 8.1221 unpack_nas_SetEventReportInd_t Struct Reference

### Data Fields

- nas_SignalStrengthTlv SSTlv
- nas_RFInfoTlv RFTlv
- nas_RejectReasonTlv RRTlv
- nas_SLQSSignalStrengthsTlv SLQSSSTlv

### 8.1221.1 Detailed Description

This structure contains unpack set event report indication parameters.

**Parameters**

| | |
|---|---|
| *SSTlv* | signal strength tlv |
| *RFTlv* | RF tlv |
| *RRTlv* | RR tlv |
| *SLQSSSTlv* | signal strength complete info tlv |

### 8.1221.2 Field Documentation

#### 8.1221.2.1 nas_RFInfoTlv unpack_nas_SetEventReportInd_t::RFTlv

#### 8.1221.2.2 nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv

#### 8.1221.2.3 nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv

#### 8.1221.2.4 nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv

## 8.1222 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference

### Data Fields

- nas_PhyCaAggScellIndType sPhyCaAggScellIndType
- nas_PhyCaAggScellDlBw sPhyCaAggScellDlBw
- nas_PhyCaAggScellInfo sPhyCaAggScellInfo
- nas_PhyCaAggPcellInfo sPhyCaAggPcellInfo
- nas_PhyCaAggScellIndex sPhyCaAggScellIndex

### 8.1222.1 Detailed Description

Structure for storing the LTE PHY CA indication parameters.

**Parameters**

| | |
|---|---|
| *pPhyCaAgg-ScellIndType* | • See nas_PhyCaAggScellIndType for more information. |
| *sPhyCaAgg-ScellDlBw* | • See nas_PhyCaAggScellDlBw for more information. |
| *sPhyCaAgg-ScellInfo* | • See nas_PhyCaAggScellInfo for more information. |
| *sPhyCaAgg-PcellInfo* | • See nas_PhyCaAggPcellInfo for more information. |
| *sPhyCaAgg-ScellIndex* | • See nas_PhyCaAggScellIndex for more information. |

### 8.1222.2 Field Documentation

#### 8.1222.2.1 nas_PhyCaAggPcellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggPcellInfo

#### 8.1222.2.2 nas_PhyCaAggSceIIDlBw unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIDlBw

#### 8.1222.2.3 nas_PhyCaAggSceIIIndex unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIIndex

#### 8.1222.2.4 nas_PhyCaAggSceIIIndType unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIIndType

#### 8.1222.2.5 nas_PhyCaAggSceIIInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggSceIIInfo

## 8.1223 unpack_nas_SetNetworkPreference_t Struct Reference

### Data Fields

- uint16_t TlvresultTlvresult

### 8.1223.1 Detailed Description

This structure contains unpack set netwokr preference parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack result |

### 8.1223.2 Field Documentation

#### 8.1223.2.1 uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult

## 8.1224 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

### Data Fields

- uint8_t roaming

### 8.1224.1 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 |

### 8.1224.2 Field Documentation

#### 8.1224.2.1 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

## 8.1225 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

### Data Fields

- NASServingSystemInfo SSInfo
- uint16_t Tlvresult

### 8.1225.1 Detailed Description

This structure contains unpack set serving system callback indication parameters.

**Parameters**

| | |
|---|---|
| *SSInfo* | • Serving system parameters information<br>– See NASServingSystemInfo for more details |
| *Tlvresult* | • unpack result |

### 8.1225.2 Field Documentation

#### 8.1225.2.1 NASServingSystemInfo unpack_nas_SetServingSystemCallback_ind_t::SSInfo

#### 8.1225.2.2 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

## 8.1226 unpack_nas_SLQSGetErrorRate_t Struct Reference

### Data Fields

- uint16_t * pCDMAFrameErrRate
- uint16_t * pHDRPackErrRate
- uint8_t * pGSMBER
- uint8_t * pWCDMABER

### 8.1226.1 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>• NULL for invalid data |

| | |
|---|---|
| *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>• NULL for invalid data |
| *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>• NULL for invalid data |
| *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>• NULL for invalid data |

### 8.1226.2 Field Documentation

#### 8.1226.2.1 uint16_t* unpack_nas_SLQSGetErrorRate_t::pCDMAFrameErrRate

#### 8.1226.2.2 uint8_t* unpack_nas_SLQSGetErrorRate_t::pGSMBER

#### 8.1226.2.3 uint16_t* unpack_nas_SLQSGetErrorRate_t::pHDRPackErrRate

#### 8.1226.2.4 uint8_t* unpack_nas_SLQSGetErrorRate_t::pWCDMABER

## 8.1227 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

### Data Fields

- NasGetLTECphyCaInfo LTECphyCaInfo
- uint16_t Tlvresult

### 8.1227.1 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 |

### 8.1227.2 Field Documentation

#### 8.1227.2.1 NasGetLTECphyCaInfo unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCaInfo

#### 8.1227.2.2 uint16_t unpack_nas_SlqsGetLTECphyCAInfo_t::Tlvresult

## 8.1228 unpack_nas_SLQSGetNetworkTime_t Struct Reference

### Data Fields

- nas_timeInfo * p3GPP2TimeInfo
- nas_timeInfo * p3GPPTimeInfo

### 8.1228.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

**Parameters**

| | |
|---|---|
| *p3GPP2Time-Info* | [Optional]<br>• See nas_timeInfo for more information<br>• NULL for invalid data. |
| *p3GPPTimeInfo* | [Optional]<br>• See nas_timeInfo for more information<br>• NULL for invalid data. |

### 8.1228.2 Field Documentation

#### 8.1228.2.1 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPP2TimeInfo

#### 8.1228.2.2 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPPTimeInfo

## 8.1229 unpack_nas_SLQSGetOperatorNameData_t Struct Reference

### Data Fields

- nas_serviceProviderName * pSrvcProviderName
- nas_operatorPLMNList * pOperatorPLMNList
- nas_PLMNNetworkName * pPLMNNetworkName
- nas_operatorNameString * pOperatorNameString
- nas_PLMNNetworkNameData * pNITZInformation

### 8.1229.1 Detailed Description

Structure for Operator Name unpack.

**Parameters**

| | |
|---|---|
| *pSrvcProvider-Name* | • Refer nas_serviceProviderName for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• NULL for invalid data |
| *pOperatorPLM-NList* | • Refer nas_operatorPLMNList for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• NULL for invalid data |
| *pPLMNNetwork-Name* | • Refer nas_PLMNNetworkName for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• NULL for invalid data |
| *pOperatorName-String* | • Refer nas_operatorNameString for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• NULL for invalid data |
| *pNITZ-Information* | • Refer nas_PLMNNetworkNameData for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• NULL for invalid data |

### 8.1229.2 Field Documentation

#### 8.1229.2.1 nas_PLMNNetworkNameData∗ unpack_nas_SLQSGetOperatorNameData_t::pNITZInformation

#### 8.1229.2.2 nas_operatorNameString∗ unpack_nas_SLQSGetOperatorNameData_t::pOperatorNameString

#### 8.1229.2.3 nas_operatorPLMNList∗ unpack_nas_SLQSGetOperatorNameData_t::pOperatorPLMNList

#### 8.1229.2.4 nas_PLMNNetworkName∗ unpack_nas_SLQSGetOperatorNameData_t::pPLMNNetworkName

#### 8.1229.2.5 nas_serviceProviderName∗ unpack_nas_SLQSGetOperatorNameData_t::pSrvcProviderName

## 8.1230 unpack_nas_SLQSGetPLMNName_t Struct Reference

### Data Fields

- uint8_t spnEncoding
- uint8_t spnLength
- char spn [255]
- uint8_t shortNameEn
- uint8_t shortNameCI
- uint8_t shortNameSB
- char shortNameLen
- uint8_t shortName [255]
- uint8_t longNameEn
- uint8_t longNameCI

- uint8_t longNameSB
- uint8_t longNameLen
- char longName [255]

### 8.1230.1 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. |
| *spnLength* | • Length of SPN which follows<br>• Note: This SPN value is ignored if spnLength is zero and spnEncoding is 0xff. |
| *spn* | • Service Provider name string |
| *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. |
| *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. |
| *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. |
| *shortNameLen* | • Length of shortName which follows<br>• Note: This shortName value is ignored if shortNameLen is zero and shortNameEn is 0xff. |

| | |
|---|---|
| *shortName* | • PLMN short name |
| *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. |
| *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. |
| *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. |
| *longNameLen* | • Length of longName which follows<br>• Note: This longName value is ignored if longNameLen is zero and longNameEn is 0xff. |
| *longName* | • PLMN long name |

### 8.1230.2 Field Documentation

#### 8.1230.2.1 char unpack_nas_SLQSGetPLMNName_t::longName[255]

#### 8.1230.2.2 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameCI

#### 8.1230.2.3 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameEn

#### 8.1230.2.4 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameLen

#### 8.1230.2.5 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameSB

#### 8.1230.2.6 uint8_t unpack_nas_SLQSGetPLMNName_t::shortName[255]

8.1230.2.7 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameCI

8.1230.2.8 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameEn

8.1230.2.9 char unpack_nas_SLQSGetPLMNName_t::shortNameLen

8.1230.2.10 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameSB

8.1230.2.11 char unpack_nas_SLQSGetPLMNName_t::spn[255]

8.1230.2.12 uint8_t unpack_nas_SLQSGetPLMNName_t::spnEncoding

8.1230.2.13 uint8_t unpack_nas_SLQSGetPLMNName_t::spnLength

## 8.1231 unpack_nas_SLQSGetServingSystem_t Struct Reference

### Data Fields

- nas_servSystem ServingSystem
- uint8_t RoamIndicatorVal
- nas_dataSrvCapabilities DataSrvCapabilities
- nas_currentPLMN CurrentPLMN
- uint16_t SystemID
- uint16_t NetworkID
- uint16_t BasestationID
- uint32_t BasestationLatitude
- uint32_t BasestationLongitude
- nas_roamIndList RoamingIndicatorList
- uint8_t DefaultRoamInd
- nas_qaQmi3Gpp2TimeZone Gpp2TimeZone
- uint8_t CDMA_P_Rev
- uint8_t GppTimeZone
- uint8_t GppNetworkDSTAdjustment
- uint16_t Lac
- uint32_t CellID
- uint8_t ConcSvcInfo
- uint8_t PRLInd
- uint8_t DTMInd
- nas_detailSvcInfo DetailedSvcInfo
- nas_CDMASysInfoExt CDMASystemInfoExt
- uint8_t HdrPersonality
- uint16_t TrackAreaCode
- nas_callBarStatus CallBarStatus

### 8.1231.1 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 |
| *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 |
| *DataSrv-Capabilities* | • Optional parameter indicating Data services capability<br>• See nas_dataSrvCapabilities for more information |
| *CurrentPLMN* | • Optional parameter indicating Current PLMN<br>• See nas_currentPLMN for more information |
| *SystemID* | • Optional parameter indicating System ID |
| *NetworkID* | • Optional parameter indicating Network ID |
| *BaseStationID* | • Optional parameter indicating Base Station Identification Number |
| *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 |
| *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 |
| *Roaming-IndicatorList* | • Optional parameter indicating Roaming Indicator List<br>• See nas_roamIndList for more information |
| *DefaultRoamInd* | • Optional parameter indicating Default Roaming Indicator<br>• Values:<br>– 0x00 - Roaming<br>– 0x01 - Home |
| *Gpp2TimeZone* | • Optional parameter indicating 3GPP2 Time Zone<br>• See nas_qaQmi3Gpp2TimeZone for more information |

| | |
|---|---|
| *CDMA_P_Rev* | • Optional parameter indicating CDMA P_Rev in use |
| *GppTimeZone* | • Optional parameter indicating Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min. (signed value). |
| *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 |
| *Lac* | • Optional parameter indicating 3GPP Location Area Code |
| *CellID* | • Optional parameter indicating 3GPP Cell ID |
| *ConcSvcInfo* | • Optional parameter indicating 3GPP2 concurrent service Info<br>• Values:<br>– 0x00 - Concurrent service not available<br>– 0x01 - Concurrent service available |
| *PRLInd* | • Optional parameter indicating 3GPP2 PRL Indicator<br>• Values:<br>– 0x00 - System not in PRL<br>– 0x01 - System is in PRL |
| *DTMInd* | • Optional parameter indicating Dual Transfer Mode Indication(GSM Only)<br>• Values:<br>– 0x00 - DTM not supported<br>– 0x01 - DTM supported |
| *DetailedSvcInfo* | • Optional parameter indicating Detailed service information<br>• See nas_detailSvcInfo for more information |
| *CDMASystem-InfoExt* | • Optional parameter indicating CDMA System Info Ext<br>• See nas_CDMASysInfoExt for more information |

| | |
|---|---|
| *HdrPersonality* | • Optional parameter indicating HDR Personality Information<br>• Values:<br>– 0x00 - Unknown<br>– 0x01 - HRPD<br>– 0x02 - eHRPD |
| *TrackAreaCode* | • Optional parameter indicating Tracking area code information for LTE |
| *CallBarStatus* | • Optional parameter indicating Call Barring Status<br>• See nas_callBarStatus for more information |

### 8.1231.2 Field Documentation

#### 8.1231.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID

#### 8.1231.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude

#### 8.1231.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude

#### 8.1231.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus

#### 8.1231.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev

#### 8.1231.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt

#### 8.1231.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID

#### 8.1231.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo

#### 8.1231.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN

#### 8.1231.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities

#### 8.1231.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd

#### 8.1231.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo

#### 8.1231.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd

#### 8.1231.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone

#### 8.1231.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment

#### 8.1231.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone

#### 8.1231.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality

#### 8.1231.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac

8.1231.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID

8.1231.2.20 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd

8.1231.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal

8.1231.2.22 **nas_roamIndList** unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList

8.1231.2.23 **nas_servSystem** unpack_nas_SLQSGetServingSystem_t::ServingSystem

8.1231.2.24 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID

8.1231.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

## 8.1232 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

### 8.1232.1 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 |
| *rxSignal-StrengthListLen* | • Number of elements in Receive Signal Strength List |
| *rxSignal-StrengthList* | • See nas_rxSignalStrengthListElement for more information |
| *ecioListLen* | • Number of elements in ECIO List |
| *ecioList* | • See nas_ecioListElement for more information |
| *Io* | • Received Io in dBm; IO is only applicable for 1xEV-DO |
| *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 |

| | |
|---|---|
| *errorRateListLen* | • Number of elements in Error Rate List |
| *errorRateList* | • See nas_errorRateListElement for more information |
| *rsrqInfo* | • See nas_rsrqInformation for more information |
| *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 |
| *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 |

### 8.1232.2 Field Documentation

#### 8.1232.2.1 nas_ecioListElement unpack_nas_SLQSGetSignalStrength_t::ecioList[18]

#### 8.1232.2.2 uint16_t unpack_nas_SLQSGetSignalStrength_t::ecioListLen

#### 8.1232.2.3 nas_errorRateListElement unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]

#### 8.1232.2.4 uint16_t unpack_nas_SLQSGetSignalStrength_t::errorRateListLen

#### 8.1232.2.5 int32_t unpack_nas_SLQSGetSignalStrength_t::lo

#### 8.1232.2.6 int16_t unpack_nas_SLQSGetSignalStrength_t::ltersrp

#### 8.1232.2.7 int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr

#### 8.1232.2.8 nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo

#### 8.1232.2.9 nas_rxSignalStrengthListElement unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]

#### 8.1232.2.10 uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen

#### 8.1232.2.11 uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask

#### 8.1232.2.12 uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr

## 8.1233 unpack_nas_SLQSGetSysInfo_t Struct Reference

### Data Fields

- nas_SrvStatusInfo * pCDMASrvStatusInfo
- nas_SrvStatusInfo * pHDRSrvStatusInfo
- nas_GSMSrvStatusInfo * pGSMSrvStatusInfo
- nas_GSMSrvStatusInfo * pWCDMASrvStatusInfo
- nas_GSMSrvStatusInfo * pLTESrvStatusInfo

- nas_CDMASysInfo * pCDMASysInfo
- nas_HDRSysInfo * pHDRSysInfo
- nas_GSMSysInfo * pGSMSysInfo
- nas_WCDMASysInfo * pWCDMASysInfo
- nas_LTESysInfo * pLTESysInfo
- nas_AddCDMASysInfo * pAddCDMASysInfo
- uint16_t * pAddHDRSysInfo
- nas_AddSysInfo * pAddGSMSysInfo
- nas_AddSysInfo * pAddWCDMASysInfo
- uint16_t * pAddLTESysInfo
- nas_CallBarringSysInfo * pGSMCallBarringSysInfo
- nas_CallBarringSysInfo * pWCDMACallBarringSysInfo
- uint8_t * pLTEVoiceSupportSysInfo
- uint8_t * pGSMCipherDomainSysInfo
- uint8_t * pWCDMACipherDomainSysInfo

### 8.1233.1 Detailed Description

This structure contains unpack get system information parameters.

**Parameters**

| | |
|---|---|
| *pCDMASrvStatusInfo* | • See nas_SrvStatusInfo for more information.<br>• NULL Pointer for invalid data. |
| *pHDRSrvStatusInfo* | • See nas_SrvStatusInfo for more information.<br>• NULL Pointer for invalid data. |
| *pGSMSrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information.<br>• NULL Pointer for invalid data. |
| *pWCDMASrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information.<br>• NULL Pointer for invalid data. |
| *pLTESrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information.<br>• NULL Pointer for invalid data. |
| *pCDMASysInfo* | • See nas_CDMASysInfo for more information.<br>• NULL Pointer for invalid data. |
| *pHDRSysInfo* | • See nas_HDRSysInfo for more information.<br>• NULL Pointer for invalid data. |
| *pGSMSysInfo* | • See nas_GSMSysInfo for more information.<br>• NULL Pointer for invalid data. |

| | |
|---|---|
| *pWCDMASysInfo* | • See nas_WCDMASysInfo for more information.<br>• NULL Pointer for invalid data. |
| *pLTESysInfo* | • See nas_LTESysInfo for more information.<br>• NULL Pointer for invalid data. |
| *pAddCDMASysInfo* | • See nas_AddCDMASysInfo for more information.<br>• NULL Pointer for invalid data. |
| *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>• NULL Pointer for invalid data. |
| *pAddGSMSysInfo* | • See nas_AddSysInfo for more information.<br>• NULL Pointer for invalid data. |
| *pAddWCDMASysInfo* | • See nas_AddSysInfo for more information.<br>• NULL Pointer for invalid data. |
| *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>• NULL Pointer for invalid data. |
| *pGSMCallBarringSysInfo* | • See nas_CallBarringSysInfo for more information.<br>• NULL Pointer for invalid data. |
| *pWCDMACallBarringSysInfo* | • See nas_CallBarringSysInfo for more information.<br>• NULL Pointer for invalid data. |
| *pLTEVoiceSupportSysInfo* | • Indicates voice support status on LTE.<br>– 0x00 - Voice is not supported<br>– 0x01 - Voice is supported<br>• NULL Pointer for invalid data. |

| | |
|---|---|
| *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>• NULL Pointer for invalid data. |
| *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>• NULL Pointer for invalid data. |

### 8.1233.2 Field Documentation

#### 8.1233.2.1 nas_AddCDMASysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo

#### 8.1233.2.2 nas_AddSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo

#### 8.1233.2.3 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo

#### 8.1233.2.4 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo

#### 8.1233.2.5 nas_AddSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo

#### 8.1233.2.6 nas_SrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo

#### 8.1233.2.7 nas_CDMASysInfo∗ unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo

#### 8.1233.2.8 nas_CallBarringSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo

#### 8.1233.2.9 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo

#### 8.1233.2.10 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMSrvStatusInfo

#### 8.1233.2.11 nas_GSMSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMSysInfo

#### 8.1233.2.12 nas_SrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo

#### 8.1233.2.13 nas_HDRSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo

#### 8.1233.2.14 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo

#### 8.1233.2.15 nas_LTESysInfo∗ unpack_nas_SLQSGetSysInfo_t::pLTESysInfo

#### 8.1233.2.16 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo

#### 8.1233.2.17 nas_CallBarringSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo

#### 8.1233.2.18 uint8_t* unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo

#### 8.1233.2.19 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo

#### 8.1233.2.20 nas_WCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo

## 8.1234 unpack_nas_SLQSGetSysSelectionPref_t Struct Reference

### Data Fields

- uint8_t * pEmerMode
- uint16_t * pModePref
- uint64_t * pBandPref
- uint16_t * pPRLPref
- uint16_t * pRoamPref
- uint64_t * pLTEBandPref
- uint8_t * pNetSelPref
- uint32_t * pSrvDomainPref
- uint32_t * pGWAcqOrderPref

### 8.1234.1 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)<br>• NULL Pointer for invalid data. |
| *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>• NULL Pointer for invalid data. |

| | |
|---|---|
| *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<br>• NULL Pointer for invalid data. |

| | |
|---|---|
| *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<br>• NULL Pointer for invalid data. |
| *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<br>• NULL Pointer for invalid data. |

| | |
|---|---|
| *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<br>• NULL Pointer for invalid data. |
| *pNetSelPref* | • Optional parameter indicating network selection preference<br>• Values:<br>– 0x00 - Automatic network selection<br>– 0x01 - Manual network selection.<br>• NULL Pointer for invalid data. |

| | |
|---|---|
| *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<br>• NULL Pointer for invalid data. |
| *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<br>• NULL Pointer for invalid data. |

### 8.1234.2 Field Documentation

#### 8.1234.2.1 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

#### 8.1234.2.2 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

#### 8.1234.2.3 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

#### 8.1234.2.4 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

#### 8.1234.2.5 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref

#### 8.1234.2.6 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

#### 8.1234.2.7 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

#### 8.1234.2.8 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

#### 8.1234.2.9 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

## 8.1235 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

### 8.1235.1 Detailed Description

Structure for 3GPP2Subscription unpack.

**Parameters**

| | |
|---|---|
| *pNAMNameInfo* | [Optional]<br>• See nas_namName for more information<br>• NULL for invalid data |
| *pDirNum* | [Optional]<br>• See nas_dirNum for more information<br>• NULL for invalid data |
| *pHomeSIDNID* | [Optional]<br>• See nas_homeSIDNID for more information<br>• NULL for invalid data |
| *pMinBasedIMSI* | [Optional]<br>• See nas_minBasedIMSI for more information<br>• NULL for invalid data |
| *pTrueIMSI* | [Optional]<br>• See nas_trueIMSI for more information<br>• NULL for invalid data |
| *pCDMAChannel* | [Optional]<br>• See nas_CDMAChannel for more information<br>• NULL for invalid data |

### 8.1235.2 Field Documentation

#### 8.1235.2.1 nas_CDMAChannel∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pCDMAChannel

#### 8.1235.2.2 nas_dirNum∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pDirNum

#### 8.1235.2.3 nas_homeSIDNID∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pHomeSIDNID

#### 8.1235.2.4 nas_minBasedIMSI∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pMinBasedIMSI

#### 8.1235.2.5 nas_namName∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pNAMNameInfo

#### 8.1235.2.6 nas_trueIMSI∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pTrueIMSI

## 8.1236 unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference

### Data Fields

- nas_GERANInfo ∗ pGERANInfo
- nas_UMTSInfo ∗ pUMTSInfo
- nas_CDMAInfo ∗ pCDMAInfo
- nas_LTEInfoIntrafreq ∗ pLTEInfoIntrafreq
- nas_LTEInfoInterfreq ∗ pLTEInfoInterfreq

- nas_LTEInfoNeighboringGSM * pLTEInfoNeighboringGSM
- nas_LTEInfoNeighboringWCDMA * pLTEInfoNeighboringWCDMA
- uint32_t * pUMTSCellID
- nas_WCDMAInfoLTENeighborCell * pWCDMAInfoLTENeighborCell

### 8.1236.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.

**Parameters**

| | |
|---|---|
| *pGERANInfo* | • See nas_GERANInfo for more information.<br>• NULL for invalid data |
| *pUMTSInfo* | • See nas_UMTSInfo for more information.<br>• NULL for invalid data |
| *pCDMAInfo* | • See nas_CDMAInfo for more information.<br>• NULL for invalid data |
| *pLTEInfo-Intrafreq* | • See nas_LTEInfoIntrafreq for more information.<br>• NULL for invalid data |
| *pLTEInfo-Interfreq* | • See nas_LTEInfoInterfreq for more information.<br>• NULL for invalid data |
| *pLTEInfo-NeighboringGS-M* | • See nas_LTEInfoNeighboringGSM for more information.<br>• NULL for invalid data |
| *pLTEInfo-NeighboringWC-DMA* | • See nas_LTEInfoNeighboringWCDMA for more information.<br>• NULL for invalid data |
| *pUMTSCellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present.<br>• NULL for invalid data |
| *pWCDMAInfoLT-ENeighborCell* | • See nas_WCDMAInfoLTENeighborCell for more information.<br>• NULL for invalid data |

### 8.1236.2 Field Documentation

#### 8.1236.2.1 nas_CDMAInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo

#### 8.1236.2.2 nas_GERANInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo

#### 8.1236.2.3 nas_LTEInfoInterfreq∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq

#### 8.1236.2.4 nas_LTEInfoIntrafreq∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq

#### 8.1236.2.5 nas_LTEInfoNeighboringGSM∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM

#### 8.1236.2.6 nas_LTEInfoNeighboringWCDMA∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA

#### 8.1236.2.7 uint32_t∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID

#### 8.1236.2.8 nas_UMTSInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo

#### 8.1236.2.9 nas_WCDMAInfoLTENeighborCell∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

## 8.1237 unpack_nas_SLQSNASGeteDRXParams_t Struct Reference

### Data Fields

- uint8_t ∗ pCycleLen
- uint8_t ∗ pPagingTimeWindow
- uint8_t ∗ pEdrxEnable

### 8.1237.1 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 |
| *pPagingTime-Window* | [Optional]<br>• eDRX paging time window<br>• Value is per 3GPP TS 24.008 Rel-13 Section10.5.5.32 |
| *pEdrxEnable* | [Optional]<br>• eDRX Enable<br>• Values<br>– 0x00 - Disable<br>– 0x01 - Enable |

### 8.1237.2 Field Documentation

#### 8.1237.2.1 uint8_t∗ unpack_nas_SLQSNASGeteDRXParams_t::pCycleLen

#### 8.1237.2.2 uint8_t∗ unpack_nas_SLQSNASGeteDRXParams_t::pEdrxEnable

#### 8.1237.2.3 uint8_t* unpack_nas_SLQSNASGeteDRXParams_t::pPagingTimeWindow

## 8.1238 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference

### Data Fields

- uint8_t * pColorCode

### 8.1238.1 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>• NULL for invalid data |

### 8.1238.2 Field Documentation

#### 8.1238.2.1 uint8_t* unpack_nas_SLQSNasGetHDRColorCode_t::pColorCode

## 8.1239 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

### 8.1239.1 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 |

### 8.1239.2 Field Documentation

#### 8.1239.2.1 nas_LTEOperationMode* unpack_nas_SLQSNasGetRFInfo_t::pLTEOperationMode

#### 8.1239.2.2 nas_RfBandInfoExtFormat* unpack_nas_SLQSNasGetRFInfo_t::pRfBandInfoExtFormat

#### 8.1239.2.3 nas_RfBandwidthInfo* unpack_nas_SLQSNasGetRFInfo_t::pRfBandwidthInfo

#### 8.1239.2.4 nas_RfDedicatedBandInfo* unpack_nas_SLQSNasGetRFInfo_t::pRfDedicatedBandInfo

#### 8.1239.2.5 nas_RfBandInfoList unpack_nas_SLQSNasGetRFInfo_t::rfbandInfoList

#### 8.1239.2.6 uint16_t unpack_nas_SLQSNasGetRFInfo_t::TlvResult

## 8.1240 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

### Data Fields

- cdmaSSInfo CDMASSInfo
- hdrSSInfo HDRSSInfo
- int8_t GSMSSInfo
- cdmaSSInfo WCDMASSInfo
- lteSSInfo LTESSInfo

### 8.1240.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

**Parameters**

| | |
|---|---|
| *CDMASSInfo* | • See cdmaSSInfo for more information |
| *HDRSSInfo* | • See hdrSSInfo for more information |
| *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. |
| *WCDMASSInfo* | • See cdmaSSInfo for more information |
| *LTESSInfo* | • See lteSSInfo for more information |

### 8.1240.2 Field Documentation

#### 8.1240.2.1 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo

#### 8.1240.2.2 int8_t unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo

#### 8.1240.2.3 hdrSSInfo unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo

#### 8.1240.2.4 lteSSInfo unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo

#### 8.1240.2.5 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo

## 8.1241 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference

### Data Fields

- nas_rxInfo * pRXChain0Info
- nas_rxInfo * pRXChain1Info
- nas_txInfo * pTXInfo

### 8.1241.1 Detailed Description

Structure for TxRx Info unpack

**Parameters**

| | |
|---|---|
| *pRXChain0Info* | [Optional]<br>• See nas_rxInfo for more information.<br>• NULL for invalid data. |
| *pRXChain1Info* | [Optional]<br>• See nas_rxInfo for more information.<br>• NULL for invalid data. |
| *pTXInfo* | [Optional]<br>• See nas_txInfo for more information.<br>• NULL for invalid data. |

### 8.1241.2 Field Documentation

#### 8.1241.2.1 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain0Info

#### 8.1241.2.2 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain1Info

#### 8.1241.2.3 nas_txInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pTXInfo

## 8.1242 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference

### Data Fields

- nas_UniversalTime universalTime
- uint8_t * pTimeZone

- uint8_t * pDayltSavAdj
- uint8_t * pRadioInterface

### 8.1242.1 Detailed Description

Structure for storing the NAS Network Time indication parameters.

**Parameters**

| | |
|---|---|
| *universalTime* | • See nas_UniversalTime for more information. |
| *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>• NULL for invalid data. |
| *pDayltSavAdj* | • Daylight Saving Adjustment.<br>• Daylight saving adjustment in hr.<br>– Possible values: 0, 1, and 2.<br>• NULL for invalid data. |
| *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>• NULL for invalid data. |

### 8.1242.2 Field Documentation

#### 8.1242.2.1 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

#### 8.1242.2.2 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

#### 8.1242.2.3 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

#### 8.1242.2.4 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

## 8.1243 unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference

### Data Fields

- cdmaSSInfo * pCDMASigInfo
- hdrSSInfo * pHDRSigInfo
- int8_t * pGSMSigInfo

- cdmaSSInfo * pWCDMASigInfo
- lteSSInfo * pLTESigInfo
- int8_t * pRscp
- tdscdmaSigInfoExt * pTDSCDMASigInfoExt

### 8.1243.1 Detailed Description

Structure for storing the nasSigInfo indication parameters.

**Parameters**

| | |
|---|---|
| *pCDMASigInfo* | • See cdmaSSInfo for more information.<br>• NULL Pointer for invalid data. |
| *pHDRSigInfo* | • See hdrSSInfo for more information.<br>• NULL Pointer for invalid data. |
| *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>• NULL Pointer for invalid data. |
| *pWCDMASigInfo* | • See cdmaSSInfo for more information.<br>• NULL Pointer for invalid data. |
| *pLTESigInfo* | • See lteSSInfo for more information.<br>• NULL Pointer for invalid data. |
| *pRscp* | • RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. Measurement range: -120 dBm to -25 dBm.<br>• NULL Pointer for invalid data. |
| *pTDSCDMASig-InfoExt* | • See tdscdmaSigInfoExt for more information.<br>• NULL Pointer for invalid data. |

### 8.1243.2 Field Documentation

#### 8.1243.2.1 cdmaSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo

#### 8.1243.2.2 int8_t* unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo

#### 8.1243.2.3 hdrSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo

#### 8.1243.2.4 lteSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo

#### 8.1243.2.5 int8_t* unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp

#### 8.1243.2.6 tdscdmaSigInfoExt∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt

#### 8.1243.2.7 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo

## 8.1244 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference

### Data Fields

- nas_wcdmaUARFCN ∗ pWcdmaUARFCN
- nas_lteEARFCN ∗ pLteEARFCN
- nas_ltePCI ∗ pLtePCI

### 8.1244.1 Detailed Description

Structure for Get Channel Lock unpack.

**Parameters**

| | |
|---|---|
| *pWcdmaUARFCN* | [Optional]<br>• See nas_wcdmaUARFCN for more information<br>• NULL for invalid data |
| *pLteEARFCN* | [Optional]<br>• See nas_lteEARFCN for more information<br>• NULL for invalid data |
| *pLtePCI* | [Optional]<br>• See nas_ltePCI for more information<br>• NULL for invalid data |

### 8.1244.2 Field Documentation

#### 8.1244.2.1 nas_lteEARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLteEARFCN

#### 8.1244.2.2 nas_ltePCI∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLtePCI

#### 8.1244.2.3 nas_wcdmaUARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pWcdmaUARFCN

## 8.1245 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

### Data Fields

- nas_CommInfo commonInfo
- nas_LTEInfo ∗ pLTEInfo

### 8.1245.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

**Parameters**

| | |
|---|---|
| *commonInfo* | (mandatory)<br>• See nas_CommInfo for more information |
| *pLTEInfo* | (optional)<br>• See nas_LTEInfo for more information<br>• NULL for invalid data. |

### 8.1245.2 Field Documentation

#### 8.1245.2.1 nas_CommInfo unpack_nas_SLQSNasSwiModemStatus_t::commonInfo

#### 8.1245.2.2 nas_LTEInfo* unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo

## 8.1246 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference

### Data Fields

- NASQmiCbkNasSwiOTAMessageInd Info
- uint16_t TlvResult

### 8.1246.1 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 |

### 8.1246.2 Field Documentation

#### 8.1246.2.1 NASQmiCbkNasSwiOTAMessageInd unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info

#### 8.1246.2.2 uint16_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::TlvResult

## 8.1247 unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference

### Data Fields

- char t3396_apn [101]
- uint8_t t3396_plmn_id [3]
- uint32_t t3396_val

### 8.1247.1 Detailed Description

Structure for Network Timer indication parameters.

**Parameters**

| | |
|---|---|
| *t3396_apn* | • apn |
| *t3396_plmn_id* | • plmn id |
| *t3396_val* | • timer value |

### 8.1247.2 Field Documentation

#### 8.1247.2.1 char unpack_nas_SLQSNasTimerCallback_ind_t::t3396_apn[101]

#### 8.1247.2.2 uint8_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_plmn_id[3]

#### 8.1247.2.3 uint32_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_val

## 8.1248 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference

### Data Fields

- NASQmiCbkNasSystemSelPrefInd Info
- uint16_t Tlvresult

### 8.1248.1 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 |

### 8.1248.2 Field Documentation

#### 8.1248.2.1 NASQmiCbkNasSystemSelPrefInd unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

#### 8.1248.2.2 uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult

## 8.1249 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference

### Data Fields

- uint16_t * pCurrentPersonality
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtocolSubtypeElement

### 8.1249.1 Detailed Description

Structure for HDRPersonality unpack.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality[Out]* | • Current active personality index.<br>• NULL for invalid data |
| *pPersonalityList-Length[In/Out]* | • Number of Personality Protocol Subtype contains in this response.<br>• maximum input value is 3<br>• NULL for invalid data |
| *pProtocol-Subtype-Element[Out]* | • See nas_protocolSubtypeElement for more information.<br>• NULL for invalid data |

### 8.1249.2 Field Documentation

#### 8.1249.2.1 uint16_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pCurrentPersonality

#### 8.1249.2.2 uint8_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pPersonalityListLength

#### 8.1249.2.3 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRPersonality_t::pProtocolSubtypeElement

## 8.1250 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference

### Data Fields

- uint16_t * pCurrentPrsnlty
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtoSubTypElmnt
- uint64_t * pAppSubType

### 8.1250.1 Detailed Description

Structure for HDRProtSubtype unpack.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality* | • Current active personality index.<br>• NULL for invalid data |

| | |
|---|---|
| *pPersonalityList-Length* | • Number of Personality Protocol Subtype contains in this response.<br>• maximum input value is 4<br>• NULL for invalid data |
| *pProtocol-SubtypeElement* | • See nas_protocolSubtypeElement for more information.<br>• NULL for invalid data |
| *pAppSubType* | • Stream application subtype<br>• Application subtype for each stream<br>• NULL for invalid data |

### 8.1250.2 Field Documentation

#### 8.1250.2.1 uint64_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pAppSubType

#### 8.1250.2.2 uint16_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pCurrentPrsnlty

#### 8.1250.2.3 uint8_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pPersonalityListLength

#### 8.1250.2.4 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pProtoSubTypElmnt

## 8.1251 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference

### Data Fields

- nas_DRCParams * pDRCParams
- uint8_t * pUATI
- nas_PilotSetData * pPilotSetData

### 8.1251.1 Detailed Description

Structure for HRPD Stats unpack.

**Parameters**

| | |
|---|---|
| *pDRCParams* | • See nas_DRCParams for more information.<br>• NULL for invalid data |
| *pUATI* | • A 128-bit address that includes the access terminal identifier and subnet ID<br>• Size must be 16 bytes<br>• NULL for invalid data |
| *pPilotSetData* | • See nas_PilotSetData for more information.<br>• NULL for invalid data |

### 8.1251.2 Field Documentation

#### 8.1251.2.1 nas_DRCParams∗ unpack_nas_SLQSSwiGetHRPDStats_t::pDRCParams

#### 8.1251.2.2 nas_PilotSetData∗ unpack_nas_SLQSSwiGetHRPDStats_t::pPilotSetData

#### 8.1251.2.3 uint8_t∗ unpack_nas_SLQSSwiGetHRPDStats_t::pUATI

## 8.1252 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

### Data Fields

- uint8_t ValidityCW0
- uint8_t CQIValueCW0
- uint8_t ValidityCW1
- uint8_t CQIValueCW1

### 8.1252.1 Detailed Description

This structure contains paramaters unpack fetch CQI parameters for LTE data session.

**Parameters**

| | |
|---|---|
| *ValidityCW0[OUT]* | • Values<br>– 0- Invalid.<br>– 1- Valid. |
| *CQIValueCW0[OUT]* | • Values<br>– Range 0∼15 |
| *ValidityCW1[OUT]* | • Values<br>– 0- Invalid.<br>– 1- Valid. |
| *CQIValueCW1[OUT]* | • Values<br>– Range 0∼15 |

### 8.1252.2 Field Documentation

#### 8.1252.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

#### 8.1252.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

#### 8.1252.2.3 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

#### 8.1252.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

## 8.1253 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference

### Data Fields

- nas_SccRxInfo * pSccRxInfo

### 8.1253.1 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>• NULL for invalid data |

### 8.1253.2 Field Documentation

#### 8.1253.2.1 nas_SccRxInfo* unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo

## 8.1254 unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t Struct Reference

### Data Fields

- nas_HDRPersonality_Ind_Data * pHDRPersInd

### 8.1254.1 Detailed Description

Structure for HDR Personality indication unpack.

**Parameters**

| | |
|---|---|
| *pHDRPersInd* | • See nas_HDRPersonality_Ind_Data for more information. |

### 8.1254.2 Field Documentation

#### 8.1254.2.1 nas_HDRPersonality_Ind_Data* unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t::pHDRPersInd

## 8.1255 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference

### Data Fields

- uint8_t * pObjectVer
- nas_NetworkStat1x * pNetworkStat1x
- nas_NetworkStatEVDO * pNetworkStatEVDO
- nas_DeviceConfigDetail * pDeviceConfigDetail
- nas_DataStatusDetail * pDataStatusDetail

### 8.1255.1 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>• NULL for invalid data |
| *pNetworkStat1x* | • See nas_NetworkStat1x for more information<br>• NULL for invalid data |
| *pNetworkStatEVDO* | • See nas_NetworkStatEVDO for more information.<br>• NULL for invalid data |
| *pDeviceConfigDetail* | • See nas_DeviceConfigDetail for more information.<br>• NULL for invalid data |
| *pDataStatusDetail* | • See nas_DataStatusDetail for more information.<br>• NULL for invalid data |

### 8.1255.2 Field Documentation

#### 8.1255.2.1 nas_DataStatusDetail∗ unpack_nas_SLQSSwiNetworkDebug_t::pDataStatusDetail

#### 8.1255.2.2 nas_DeviceConfigDetail∗ unpack_nas_SLQSSwiNetworkDebug_t::pDeviceConfigDetail

#### 8.1255.2.3 nas_NetworkStat1x∗ unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStat1x

#### 8.1255.2.4 nas_NetworkStatEVDO∗ unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStatEVDO

#### 8.1255.2.5 uint8_t∗ unpack_nas_SLQSSwiNetworkDebug_t::pObjectVer

## 8.1256 unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t Struct Reference

### Data Fields

- nas_RankIndicatorTlv rankIndicatorTlv

### 8.1256.1 Detailed Description

Structure for Rank Indicator indication unpack.

**Parameters**

| | |
|---|---|
| *rankIndicatorTlv* | • See nas_RankIndicatorTlv for more information. |

### 8.1256.2 Field Documentation

#### 8.1256.2.1 nas_RankIndicatorTlv unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t::rankIndicatorTlv

## 8.1257 unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference

### Data Fields

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

### 8.1257.1 Detailed Description

This structure contains unpack system information callback indication parameters.

**Parameters**

| | |
|---|---|
| *pCDMASrvStatusInfo* | • See nas_SrvStatusInfo for more information. |
| *pHDRSrvStatusInfo* | • See nas_SrvStatusInfo for more information. |
| *pGSMSrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information. |
| *pWCDMASrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information. |

| | |
|---|---|
| *pLTESrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information. |
| *pCDMASysInfo* | • See nas_CDMASysInfo for more information. |
| *pHDRSysInfo* | • See nas_HDRSysInfo for more information. |
| *pGSMSysInfo* | • See nas_GSMSysInfo for more information. |
| *pWCDMASysInfo* | • See nas_WCDMASysInfo for more information. |
| *pLTESysInfo* | • See nas_LTESysInfo for more information. |
| *pAddCDMASysInfo* | • See nas_AddCDMASysInfo for more information. |
| *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. |
| *pAddGSMSysInfo* | • See nas_AddSysInfo for more information. |
| *pAddWCDMASysInfo* | • See nas_AddSysInfo for more information. |
| *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. |
| *pGSMCallBarringSysInfo* | • See nas_CallBarringSysInfo for more information. |
| *pWCDMACallBarringSysInfo* | • See nas_CallBarringSysInfo for more information. |
| *pLTEVoiceSupportSysInfo* | • Indicates voice support status on LTE.<br>– 0x00 - Voice is not supported<br>– 0x01 - Voice is supported |

| | |
|---|---|
| *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 |
| *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 |
| *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 |

### 8.1257.2 Field Documentation

#### 8.1257.2.1 nas_AddCDMASysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo

#### 8.1257.2.2 nas_AddSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSMSysInfo

#### 8.1257.2.3 uint16_t* unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDRSysInfo

#### 8.1257.2.4 uint16_t* unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTESysInfo

#### 8.1257.2.5 nas_AddSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo

#### 8.1257.2.6 nas_SrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo

#### 8.1257.2.7 nas_CDMASysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo

#### 8.1257.2.8 nas_CallBarringSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo

#### 8.1257.2.9 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo

#### 8.1257.2.10 nas_GSMSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo

#### 8.1257.2.11 nas_GSMSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo

#### 8.1257.2.12 nas_SrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSrvStatusInfo

#### 8.1257.2.13 nas_HDRSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSysInfo

#### 8.1257.2.14 nas_GSMSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo

#### 8.1257.2.15 nas_LTESysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo

#### 8.1257.2.16 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo

#### 8.1257.2.17 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange

#### 8.1257.2.18 nas_CallBarringSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo

#### 8.1257.2.19 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo

#### 8.1257.2.20 nas_GSMSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo

#### 8.1257.2.21 nas_WCDMASysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo

## 8.1258 unpack_omaDmConfigTlv_t Struct Reference

### Data Fields

- uint8_t state
- uint8_t userInputReq
- uint16_t userInputTimeout
- uint16_t alertmsglength
- uint8_t alertmsg [256]

### 8.1258.1 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) |

### 8.1258.2 Field Documentation

#### 8.1258.2.1 uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]

#### 8.1258.2.2 uint16_t unpack_omaDmConfigTlv_t::alertmsglength

#### 8.1258.2.3 uint8_t unpack_omaDmConfigTlv_t::state

#### 8.1258.2.4 uint8_t unpack_omaDmConfigTlv_t::userInputReq

#### 8.1258.2.5 uint16_t unpack_omaDmConfigTlv_t::userInputTimeout

## 8.1259 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

### 8.1259.1 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 |

## 8.1259.2 Field Documentation

### 8.1259.2.1 uint8_t unpack_omaDmFotaTlv_t::description[256]

### 8.1259.2.2 uint16_t unpack_omaDmFotaTlv_t::descriptionlength

### 8.1259.2.3 uint32_t unpack_omaDmFotaTlv_t::fwdloadsize

### 8.1259.2.4 uint32_t unpack_omaDmFotaTlv_t::fwloadComplete

### 8.1259.2.5 uint16_t unpack_omaDmFotaTlv_t::namelength

### 8.1259.2.6 uint8_t unpack_omaDmFotaTlv_t::package_name[256]

### 8.1259.2.7 uint8_t unpack_omaDmFotaTlv_t::sessionType

### 8.1259.2.8 uint8_t unpack_omaDmFotaTlv_t::severity

### 8.1259.2.9 uint8_t unpack_omaDmFotaTlv_t::state

### 8.1259.2.10 uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus

#### 8.1259.2.11 uint8_t unpack_omaDmFotaTlv_t::userInputReq

#### 8.1259.2.12 uint16_t unpack_omaDmFotaTlv_t::userInputTimeout

#### 8.1259.2.13 uint8_t unpack_omaDmFotaTlv_t::version[256]

#### 8.1259.2.14 uint16_t unpack_omaDmFotaTlv_t::versionlength

## 8.1260 unpack_omaDmNotificationsTlv_t Struct Reference

### Data Fields

- uint8_t notification
- uint16_t sessionStatus

### 8.1260.1 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. |

### 8.1260.2 Field Documentation

#### 8.1260.2.1 uint8_t unpack_omaDmNotificationsTlv_t::notification

#### 8.1260.2.2 uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus

## 8.1261 unpack_pds_GetPDSDefaults_t Struct Reference

### Data Fields

- uint32_t * pOperation
- uint8_t * pTimeout
- uint32_t * pInterval
- uint32_t * pAccuracy

### 8.1261.1 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>• NULL pointer - Invalid data. |
| *pTimeout* | • Maximum amount of time (seconds) to work on each fix, maximum is 255<br>• NULL pointer - Invalid data. |
| *pInterval* | • Interval (seconds) between fix requests<br>• NULL pointer - Invalid data. |
| *pAccuracy* | • Preferred accuracy threshold (meters)<br>• NULL pointer - Invalid data. |

### 8.1261.2 Field Documentation

#### 8.1261.2.1 uint32_t* unpack_pds_GetPDSDefaults_t::pAccuracy

#### 8.1261.2.2 uint32_t* unpack_pds_GetPDSDefaults_t::pInterval

#### 8.1261.2.3 uint32_t* unpack_pds_GetPDSDefaults_t::pOperation

#### 8.1261.2.4 uint8_t* unpack_pds_GetPDSDefaults_t::pTimeout

## 8.1262 unpack_pds_GetPDSState_t Struct Reference

### Data Fields

- uint32_t * pEnabledStatus
- uint32_t * pTrackingStatus

### 8.1262.1 Detailed Description

Structure contain parameters of current PDS state.

**Parameters**

| | |
|---|---|
| *pEnabledStatus* | • Current PDS state<br>– 0 - disable<br>– 1 - enable<br>• NULL pointer - Invalid data. |
| *pTrackingStatus* | • Current PDS tracking session state<br>• Values:<br>– 0x00 - Unknown<br>– 0x01 - Inactive<br>– 0x02 - Active<br>• NULL pointer - Invalid data. |

### 8.1262.2 Field Documentation

#### 8.1262.2.1 uint32_t* unpack_pds_GetPDSState_t::pEnabledStatus

#### 8.1262.2.2 uint32_t* unpack_pds_GetPDSState_t::pTrackingStatus

## 8.1263 unpack_pds_GetPortAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t * pbAuto

### 8.1263.1 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 |

- NULL pointer - Invalid data.

### 8.1263.2 Field Documentation

#### 8.1263.2.1 uint32_t* unpack_pds_GetPortAutomaticTracking_t::pbAuto

## 8.1264 unpack_pds_GetServiceAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t * pbAuto

### 8.1264.1 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>• NULL pointer - Invalid data. |

### 8.1264.2 Field Documentation

#### 8.1264.2.1 uint32_t* unpack_pds_GetServiceAutomaticTracking_t::pbAuto

## 8.1265 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference

### Data Fields

- uint32_t * pbEnabled
- uint16_t * pInterval

### 8.1265.1 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>• NULL pointer - Invalid data. |
| *pInterval* | • Interval (hours) between XTRA downloads<br>• NULL pointer - Invalid data. |

### 8.1265.2 Field Documentation

#### 8.1265.2.1 uint32_t* unpack_pds_GetXTRAAutomaticDownload_t::pbEnabled

#### 8.1265.2.2 uint16_t* unpack_pds_GetXTRAAutomaticDownload_t::pInterval

## 8.1266 unpack_pds_GetXTRANetwork_t Struct Reference

### Data Fields

- uint32_t * pPreference

### 8.1266.1 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>• NULL pointer - Invalid data. |

### 8.1266.2 Field Documentation

#### 8.1266.2.1 uint32_t* unpack_pds_GetXTRANetwork_t::pPreference

## 8.1267 unpack_pds_GetXTRAValidity_t Struct Reference

### Data Fields

- uint16_t * pGPSWeek
- uint16_t * pGPSWeekOffset
- uint16_t * pDuration

### 8.1267.1 Detailed Description

Structure contain the parameter for XTRA database validity period

**Parameters**

| | |
|---|---|
| *pGPSWeek* | • Starting GPS week of validity period<br>• NULL pointer - Invalid data. |
| *pGPSWeek-Offset* | • Starting GPS week offset (minutes) of validity period<br>• NULL pointer - Invalid data. |
| *pDuration* | • Length of validity period (hours)<br>• NULL pointer - Invalid data. |

### 8.1267.2 Field Documentation

#### 8.1267.2.1 uint16_t* unpack_pds_GetXTRAValidity_t::pDuration

#### 8.1267.2.2 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeek

#### 8.1267.2.3 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeekOffset

## 8.1268 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

### 8.1268.1 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). |
| *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 |
| *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 |

| | |
|---|---|
| *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>• 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 |

### 8.1268.2 Field Documentation

#### 8.1268.2.1 double unpack_pds_SetEventReport_Ind_t::dLatitude

#### 8.1268.2.2 double unpack_pds_SetEventReport_Ind_t::dLongitude

#### 8.1268.2.3 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLatitude

#### 8.1268.2.4 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLongitude

#### 8.1268.2.5 uint8_t unpack_pds_SetEventReport_Ind_t::has_PositionDataNMEA

#### 8.1268.2.6 uint8_t unpack_pds_SetEventReport_Ind_t::has_posSrc

#### 8.1268.2.7 uint8_t unpack_pds_SetEventReport_Ind_t::has_SessionStatus

#### 8.1268.2.8 char unpack_pds_SetEventReport_Ind_t::PositionDataNMEA[200]

#### 8.1268.2.9 uint32_t unpack_pds_SetEventReport_Ind_t::posSrc

#### 8.1268.2.10 uint8_t unpack_pds_SetEventReport_Ind_t::SessionStatus

## 8.1269 unpack_pds_SetPdsState_Ind_t Struct Reference

### Data Fields

- uint16_t TlvPresent
- uint32_t EnabledStatus
- uint32_t TrackingStatus

### 8.1269.1 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 |
| *TrackingStatus* | • Tracking session state.<br>• Values<br>– 0x00 - Unknown<br>– 0x01 - Inactive<br>– 0x02 - Active |

### 8.1269.2 Field Documentation

#### 8.1269.2.1 uint32_t unpack_pds_SetPdsState_Ind_t::EnabledStatus

#### 8.1269.2.2 uint16_t unpack_pds_SetPdsState_Ind_t::TlvPresent

#### 8.1269.2.3 uint32_t unpack_pds_SetPdsState_Ind_t::TrackingStatus

## 8.1270 unpack_pds_SLQSGetAGPSConfig_t Struct Reference

### Data Fields

- uint32_t * pServerAddress
- uint32_t * pServerPort
- uint8_t * pServerURL
- uint8_t * pServerURLLength

### 8.1270.1 Detailed Description

Structure contain the parameter for PDS AGPS (MS-based) configuration unpack.

**Parameters**

| | |
|---|---|
| *pServerAddress* | • IPv4 address of AGPS server. "0" if not set |
| *pServerPort* | • Port number of AGPS server. "0" if not set |
| *pServerURL* | • URL of the AGPS server. "0" if not set |
| *pServerURL-Length* | • URL length of AGPS server. "0" if not set |

### 8.1270.2 Field Documentation

#### 8.1270.2.1 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerAddress

#### 8.1270.2.2 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerPort

#### 8.1270.2.3 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURL

#### 8.1270.2.4 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURLLength

## 8.1271 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

### 8.1271.1 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 |
| *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 |
| *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) |
| *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) |
| *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) |

| | |
|---|---|
| *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) |
| *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) |
| *TimeStmp_tow_-ms* | • Time stamp in GPS time of week( in milliseconds) |
| *TimeStmp_gps-_week* | • GPS week number |
| *Time_uncert_ms* | • Time uncertainty (in milliseconds). The uncertainty is provided at 99 percent confidence. |
| *Iono_valid* | • Iono validity.<br>• Values:<br>– 0 - Invalid<br>– 1 - Valid |
| *gps_ephemeris-_sv_msk* | • GPS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. |
| *gps_almanac_-sv_msk* | • GPS SV mask for almanac; if the bit is set, almanac for that SV is available. |
| *gps_health_sv_-msk* | • GPS SV mask for health; if the bit is set, health for that SV is available. |
| *gps_visible_sv_-msk* | • GPS SV mask for visible Svs; if the bit is set, the SV is available. |
| *glo_ephemeris_-sv_msk* | • GLONASS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. |
| *glo_almanac_sv-_msk* | • GLONASS SV mask for almanac; if the bit is set, almanac for that SV is available. |
| *glo_health_sv_-msk* | • GLONASS SV mask for health; if the bit is set, health for that SV is available. |
| *glo_visible_sv_-msk* | • GLONASS SV mask for visible SVs; if the bit is set, the SV is available. |
| *sbas_-ephemeris_sv_-msk* | • SBAS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. |

| | |
|---|---|
| *sbas_almanac_sv_msk* | • SBAS SV mask for almanac; if the bit is set, almanac for that SV is available. |
| *sbas_health_sv_msk* | • SBAS SV mask for health; if the bit is set, health for that SV is available. |
| *sbas_visible_sv_msk* | • SBAS SV mask for visible SVs; if the bit is set, the SV is available. |
| *xtra_start_gps_week* | • Current XTRA information is valid starting from this GPS week number |
| *xtra_start_gps_minutes* | • Current XTRA information is valid starting from the GPS minutes with the GPS week |
| *xtra_valid_duration_hours* | • XTRA information is valid for this many hours starting from the specified GPS week/minutes |

### 8.1271.2 Field Documentation

#### 8.1271.2.1 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Altitude

#### 8.1271.2.2 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::EngineState

#### 8.1271.2.3 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_almanac_sv_msk

#### 8.1271.2.4 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_ephemeris_sv_msk

#### 8.1271.2.5 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_health_sv_msk

#### 8.1271.2.6 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_visible_sv_msk

#### 8.1271.2.7 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_almanac_sv_msk

#### 8.1271.2.8 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_ephemeris_sv_msk

#### 8.1271.2.9 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_health_sv_msk

#### 8.1271.2.10 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_visible_sv_msk

#### 8.1271.2.11 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::HorizontalUncertainty

#### 8.1271.2.12 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::Iono_valid

#### 8.1271.2.13 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Latitude

#### 8.1271.2.14 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Longitude

#### 8.1271.2.15 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_almanac_sv_msk

#### 8.1271.2.16 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_ephemeris_sv_msk

#### 8.1271.2.17 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_health_sv_msk

#### 8.1271.2.18 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_visible_sv_msk

#### 8.1271.2.19 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Time_uncert_ms

#### 8.1271.2.20 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_gps_week

#### 8.1271.2.21 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_tow_ms

#### 8.1271.2.22 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::ValidMask

#### 8.1271.2.23 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::VerticalUncertainty

#### 8.1271.2.24 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_minutes

#### 8.1271.2.25 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_week

#### 8.1271.2.26 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_valid_duration_hours

## 8.1272 unpack_qmi_t Struct Reference

### Data Fields

- enum msgtype type
- uint16_t msgid
- uint16_t xid

### 8.1272.1 Detailed Description

qmi response context

**Parameters**

| | | |
|---|---|---|
| out | *type* | message type |
| out | *msgid* | message id |
| out | *xid* | transaction id |

### 8.1272.2 Field Documentation

#### 8.1272.2.1 uint16_t unpack_qmi_t::msgid

#### 8.1272.2.2 enum msgtype unpack_qmi_t::type

#### 8.1272.2.3 uint16_t unpack_qmi_t::xid

## 8.1273 unpack_qos_dataRate_t Struct Reference

### Data Fields

- uint32_t dataRateMax
- uint32_t guaranteedRate

### 8.1273.1 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) |

### 8.1273.2 Field Documentation

#### 8.1273.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

#### 8.1273.2.2 uint32_t unpack_qos_dataRate_t::guaranteedRate

## 8.1274 unpack_qos_IPv4Addr_t Struct Reference

### Data Fields

- uint32_t addr
- uint32_t subnetMask

### 8.1274.1 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 |

### 8.1274.2 Field Documentation

#### 8.1274.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

#### 8.1274.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

## 8.1275 unpack_qos_IPv6Addr_t Struct Reference

### Data Fields

- uint8_t addr [16]
- uint8_t prefixLen

### 8.1275.1 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. |

### 8.1275.2 Field Documentation

#### 8.1275.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

#### 8.1275.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

## 8.1276 unpack_qos_IPv6TrafCls_t Struct Reference

### Data Fields

- uint8_t val
- uint8_t mask

### 8.1276.1 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 |

### 8.1276.2 Field Documentation

#### 8.1276.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

#### 8.1276.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

## 8.1277 unpack_qos_pktErrRate_t Struct Reference

### Data Fields

- uint16_t multiplier
- uint16_t exponent

### 8.1277.1 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) |

### 8.1277.2 Field Documentation

#### 8.1277.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

#### 8.1277.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

## 8.1278 unpack_qos_Port_t Struct Reference

### Data Fields

- uint16_t port
- uint16_t range

### 8.1278.1 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 |

### 8.1278.2 Field Documentation

#### 8.1278.2.1 uint16_t unpack_qos_Port_t::port

#### 8.1278.2.2 uint16_t unpack_qos_Port_t::range

## 8.1279 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

### 8.1279.1 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. |

### 8.1279.2 Field Documentation

#### 8.1279.2.1 uint8_t unpack_qos_QosFlowInfo_t::BearerID

#### 8.1279.2.2 uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available

#### 8.1279.2.3 uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available

#### 8.1279.2.4 uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters

#### 8.1279.2.5 uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters

#### 8.1279.2.6 unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState

#### 8.1279.2.7 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]

#### 8.1279.2.8 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted

#### 8.1279.2.9 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]

#### 8.1279.2.10 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted

## 8.1280 unpack_qos_QosFlowInfoState_t Struct Reference

### Data Fields

- uint32_t id
- uint8_t isNewFlow
- uint8_t state

### 8.1280.1 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 |

### 8.1280.2 Field Documentation

#### 8.1280.2.1 uint32_t unpack_qos_QosFlowInfoState_t::id

#### 8.1280.2.2 uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow

#### 8.1280.2.3 uint8_t unpack_qos_QosFlowInfoState_t::state

## 8.1281 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference

### Data Fields

- uint8_t NWQoSStatus

### 8.1281.1 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>• 0xff - Invalid Data. |

### 8.1281.2 Field Documentation

#### 8.1281.2.1 uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus

## 8.1282 unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

### Data Fields

- uint32_t apnId
- uint8_t ambr_ul
- uint8_t ambr_dl
- uint8_t ambr_ul_ext
- uint8_t ambr_dl_ext
- uint8_t ambr_ul_ext2
- uint8_t ambr_dl_ext2

### 8.1282.1 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 |
| *ambr_ul* | • APN AMBR uplink<br>• APN AMBR uplink values from 1 kbps to 8640 kbps |
| *ambr_dl* | • APN AMBR downlink<br>• APN AMBR downlink values from 1 kbps to 8640 kbps |
| *ambr_ul_ext* | • Extended APN AMBR uplink<br>• APN AMBR uplink values from 8700 kbps to 256 Mbps |
| *ambr_dl_ext* | • Extended APN AMBR downlink<br>• APN AMBR downlink values from 8700 kbps to 256 Mbps |

| | |
|---|---|
| *ambr_ul_ext2* | • Second extended APN AMBR uplink<br>• APN AMBR uplink values from 256 Mbps to 65280 Mbps |
| *ambr_dl_ext2* | • Second extended APN AMBR downlink<br>• APN AMBR downlink values from 256 Mbps to 65280 Mbps |

### 8.1282.2 Field Documentation

#### 8.1282.2.1 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl

#### 8.1282.2.2 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext

#### 8.1282.2.3 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext2

#### 8.1282.2.4 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul

#### 8.1282.2.5 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext

#### 8.1282.2.6 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext2

#### 8.1282.2.7 uint32_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

## 8.1283 unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference

### Data Fields

- uint32_t apnId
- uint32_t total_tx_pkt
- uint32_t total_tx_pkt_drp
- uint32_t total_rx_pkt
- uint64_t total_tx_bytes
- uint64_t total_tx_bytes_drp
- uint64_t total_rx_bytes
- uint32_t numQosFlow
- unpack_QosFlowStat_t qosFlow [10]

### 8.1283.1 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 |
| *total_tx_pkt* | • sum of all packets sent |

| | |
|---|---|
| *total_tx_pkt_drp* | • sum of all(TX) packets dropped |
| *total_rx_pkt* | • sum of all packets received |
| *total_tx_bytes* | • sum of all bytes sent |
| *total_tx_bytes_drp* | • sum of all(TX) bytes dropped |
| *total_rx_bytes* | • number of received bytes for the QoS flow ID |
| *numQosFlow* | • pointer to number of QoS flow Stat |
| *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 |

### 8.1283.2 Field Documentation

#### 8.1283.2.1 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId

#### 8.1283.2.2 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow

#### 8.1283.2.3 unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]

#### 8.1283.2.4 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes

#### 8.1283.2.5 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt

#### 8.1283.2.6 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes

#### 8.1283.2.7 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp

#### 8.1283.2.8 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt

#### 8.1283.2.9 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp

## 8.1284 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

### Data Fields

- uint8_t NumFlows
- unpack_qos_QosFlowInfo_t QosFlowInfo [8]

### 8.1284.1 Detailed Description

Structure with QoS event details

**Parameters**

| | |
|---|---|
| *NumFlows* | • Number of QoS flows available |
| *QosFlowInfo* | • The Qos flow details, please check unpack_qos_QosFlowInfo_t for more information<br>• See LITEQMI_MAX_QOS_FLOWS for more information |

### 8.1284.2 Field Documentation

#### 8.1284.2.1 uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows

#### 8.1284.2.2 unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]

## 8.1285 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference

### Data Fields

- uint8_t status

### 8.1285.1 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 |

**Note**

- Technology Supported: CDMA

### 8.1285.2 Field Documentation

#### 8.1285.2.1 uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status

## 8.1286 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference

### Data Fields

- uint16_t event

### 8.1286.1 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 |

### 8.1286.2 Field Documentation

#### 8.1286.2.1 uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event

## 8.1287 unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference

### Data Fields

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

### 8.1287.1 Detailed Description

Structure with QoS status indication details

**Parameters**

| | |
|---|---|
| *id* | • Index identifying the QoS flow whose status is being reported |
| *status* | Current QoS flow status:<br>• 0x01 - QMI_QOS_STATUS_ACTIVATED<br>• 0x02 - QMI_QOS_STATUS_SUSPENDED<br>• 0x03 - QMI_QOS_STATUS_GONE |
| *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 |

| | |
|---|---|
| *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>• 0xff - Invalid Data. |

### 8.1287.2 Field Documentation

#### 8.1287.2.1 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event

#### 8.1287.2.2 uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id

#### 8.1287.2.3 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason

#### 8.1287.2.4 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status

## 8.1288 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

### 8.1288.1 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 |

## 8.1288.2 Field Documentation

### 8.1288.2.1 uint32_t unpack_qos_swiQosFilter_t::EspSpi

### 8.1288.2.2 uint16_t unpack_qos_swiQosFilter_t::Id

### 8.1288.2.3 uint8_t unpack_qos_swiQosFilter_t::index

### 8.1288.2.4 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr

### 8.1288.2.5 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr

### 8.1288.2.6 unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos

8.1288.2.7 **unpack_qos_IPv6Addr_t** unpack_qos_swiQosFilter_t::IPv6DstAddr

8.1288.2.8 uint32_t unpack_qos_swiQosFilter_t::IPv6Label

8.1288.2.9 **unpack_qos_IPv6Addr_t** unpack_qos_swiQosFilter_t::IPv6SrcAddr

8.1288.2.10 **unpack_qos_IPv6TrafCls_t** unpack_qos_swiQosFilter_t::IPv6TrafCls

8.1288.2.11 uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available

8.1288.2.12 uint8_t unpack_qos_swiQosFilter_t::is_Id_Available

8.1288.2.13 uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available

8.1288.2.14 uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available

8.1288.2.15 uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available

8.1288.2.16 uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available

8.1288.2.17 uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available

8.1288.2.18 uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available

8.1288.2.19 uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available

8.1288.2.20 uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available

8.1288.2.21 uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available

8.1288.2.22 uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available

8.1288.2.23 uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available

8.1288.2.24 uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available

8.1288.2.25 uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available

8.1288.2.26 uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available

8.1288.2.27 uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available

8.1288.2.28 uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto

8.1288.2.29 uint16_t unpack_qos_swiQosFilter_t::Precedence

8.1288.2.30 **unpack_qos_Port_t** unpack_qos_swiQosFilter_t::TCPDstPort

8.1288.2.31 **unpack_qos_Port_t** unpack_qos_swiQosFilter_t::TCPSrcPort

8.1288.2.32 **unpack_qos_Port_t** unpack_qos_swiQosFilter_t::TranDstPort

8.1288.2.33 **unpack_qos_Port_t** unpack_qos_swiQosFilter_t::TranSrcPort

8.1288.2.34 **unpack_qos_Port_t** unpack_qos_swiQosFilter_t::UDPDstPort

#### 8.1288.2.35 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort

#### 8.1288.2.36 uint8_t unpack_qos_swiQosFilter_t::version

## 8.1289 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

### 8.1289.1 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_3GPPRes-ResidualBER* | • 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_3GPPTra-HdlPri* | • 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 |

## 8.1289.2 Field Documentation

### 8.1289.2.1 unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate

### 8.1289.2.2 uint8_t unpack_qos_swiQosFlow_t::index

8.1289.2.3 uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available

8.1289.2.4 uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available

8.1289.2.5 uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available

8.1289.2.6 uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available

8.1289.2.7 uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available

8.1289.2.8 uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available

8.1289.2.9 uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available

8.1289.2.10 uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available

8.1289.2.11 uint8_t unpack_qos_swiQosFlow_t::is_TokenBucket_Available

8.1289.2.12 uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available

8.1289.2.13 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available

8.1289.2.14 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available

8.1289.2.15 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available

8.1289.2.16 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available

8.1289.2.17 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available

8.1289.2.18 uint32_t unpack_qos_swiQosFlow_t::Jitter

8.1289.2.19 uint32_t unpack_qos_swiQosFlow_t::Latency

8.1289.2.20 uint8_t unpack_qos_swiQosFlow_t::LteQci

8.1289.2.21 uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz

8.1289.2.22 uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz

8.1289.2.23 **unpack_qos_pktErrRate_t** unpack_qos_swiQosFlow_t::PktErrRate

8.1289.2.24 uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2

8.1289.2.25 **unpack_qos_tokenBucket_t** unpack_qos_swiQosFlow_t::TokenBucket

8.1289.2.26 uint8_t unpack_qos_swiQosFlow_t::TrafficClass

8.1289.2.27 uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri

8.1289.2.28 uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn

8.1289.2.29 uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER

8.1289.2.30 uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd

#### 8.1289.2.31 uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri

## 8.1290 unpack_qos_tokenBucket_t Struct Reference

### Data Fields

- uint32_t peakRate
- uint32_t tokenRate
- uint32_t bucketSz

### 8.1290.1 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 |

### 8.1290.2 Field Documentation

#### 8.1290.2.1 uint32_t unpack_qos_tokenBucket_t::bucketSz

#### 8.1290.2.2 uint32_t unpack_qos_tokenBucket_t::peakRate

#### 8.1290.2.3 uint32_t unpack_qos_tokenBucket_t::tokenRate

## 8.1291 unpack_qos_Tos_t Struct Reference

### Data Fields

- uint8_t val
- uint8_t mask

### 8.1291.1 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. |

### 8.1291.2 Field Documentation

#### 8.1291.2.1 uint8_t unpack_qos_Tos_t::mask

#### 8.1291.2.2 uint8_t unpack_qos_Tos_t::val

## 8.1292 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

### 8.1292.1 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 |

### 8.1292.2 Field Documentation

#### 8.1292.2.1 uint32_t unpack_QosFlowStat_t::bearerId

#### 8.1292.2.2 uint64_t unpack_QosFlowStat_t::tx_bytes

#### 8.1292.2.3 uint64_t unpack_QosFlowStat_t::tx_bytes_drp

#### 8.1292.2.4 uint32_t unpack_QosFlowStat_t::tx_pkt

#### 8.1292.2.5 uint32_t unpack_QosFlowStat_t::tx_pkt_drp

## 8.1293 unpack_rms_GetSMSWake_t Struct Reference

### Data Fields

- uint32_t enabled
- uint32_t wake_mask

### 8.1293.1 Detailed Description

Unpack get SMS wake parameters

**Parameters**

| | |
|---|---|
| *enabled* | • SMS wake functionality enabled<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xffffffff - invalid value. |
| *wake_mask* | • SMS wake mask to search for incoming messages (only relevant when enabled) |

### 8.1293.2 Field Documentation

#### 8.1293.2.1 uint32_t unpack_rms_GetSMSWake_t::enabled

#### 8.1293.2.2 uint32_t unpack_rms_GetSMSWake_t::wake_mask

## 8.1294 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

### 8.1294.1 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. |

### 8.1294.2 Field Documentation

#### 8.1294.2.1 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv

#### 8.1294.2.2 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::RxOkByteCountTlv

#### 8.1294.2.3 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxOkConutTlv

#### 8.1294.2.4 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxDropConutTlv

#### 8.1294.2.5 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::TxOkByteCountTlv

#### 8.1294.2.6 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxOkConutTlv

## 8.1295 unpack_sar_SLQSGetRfSarState_t Struct Reference

### Data Fields

- uint32_t * pSarRFState

### 8.1295.1 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>• NULL pointer - Invalid data. |

### 8.1295.2 Field Documentation

#### 8.1295.2.1 uint32_t* unpack_sar_SLQSGetRfSarState_t::pSarRFState

## 8.1296 unpack_sms_GetSMSCAddress_t Struct Reference

### Data Fields

- uint8_t addressSize
- uint8_t * pSMSCAddress
- uint8_t typeSize
- uint8_t * pSMSCType

### 8.1296.1 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. |
| *pSMSCAddress* | • The SMS center address represented as a NULL terminated string. |
| *typeSize* | • The maximum number of characters (including NULL terminator) that the SMS center address type array can contain. |
| *pSMSCType* | • The SMS center address type represented as a NULL terminated string. |

### 8.1296.2 Field Documentation

#### 8.1296.2.1 uint8_t unpack_sms_GetSMSCAddress_t::addressSize

#### 8.1296.2.2 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCAddress

#### 8.1296.2.3 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCType

#### 8.1296.2.4 uint8_t unpack_sms_GetSMSCAddress_t::typeSize

## 8.1297 unpack_sms_SaveSMS_t Struct Reference

### Data Fields

- uint32_t * pMessageIndex

### 8.1297.1 Detailed Description

This structure contains unpack save SMS parameter.

**Parameters**

| | |
|---|---|
| *pMessageIndex* | • The message index assigned by the device |

### 8.1297.2 Field Documentation

#### 8.1297.2.1 uint32_t* unpack_sms_SaveSMS_t::pMessageIndex

## 8.1298 unpack_sms_SendSMS_t Struct Reference

### Data Fields

- uint16_t messageID
- uint32_t messageFailureCode

### 8.1298.1 Detailed Description

This structure contains unpack send SMS parameters.

**Parameters**

| | |
|---|---|
| *messageID* | • WMS message ID |
| *messageFailure-Code* | • pointer to message failure code. If cause code is not provided, then value will be 0xFFFFFFFF |

### 8.1298.2 Field Documentation

#### 8.1298.2.1 uint32_t unpack_sms_SendSMS_t::messageFailureCode

#### 8.1298.2.2 uint16_t unpack_sms_SendSMS_t::messageID

## 8.1299 unpack_sms_SetNewSMSCallback_ind_t Struct Reference

### Data Fields

- newMTMessageTlv NewMMTlv
- transferRouteMessageTlv TRMessageTlv
- messageModeTlv MMTlv
- sMSEtwsMessageTlv ETWSTlv
- eTWSPLMNInfoTlv ETWSPLMNTlv
- sMSCAddressTlv SMSCTlv
- sMSOnIMSTlv IMSTlv

### 8.1299.1 Detailed Description

This structure contains unpack new SMS callback indication.

**Parameters**

| | |
|---|---|
| *NewMMTlv* | • MT message |
| *TRMessageTlv* | • Transfer Route MT Message<br>• See transferRouteMessageTlv for more information |
| *MMTlv* | • Message mode<br>• See messageModeTlv for more information |
| *ETWSTlv* | • ETWS Message<br>• See sMSEtwsMessageTlv for more information |
| *ETWSPLMNTlv* | • ETWS PLMN Information<br>• See eTWSPLMNInfoTlv for more information |
| *SMSCTlv* | • SMSC Address<br>• See sMSCAddressTlv for more information |
| *IMSTlv* | • SMS on IMS<br>• See sMSOnIMSTlv for more information |

### 8.1299.2 Field Documentation

#### 8.1299.2.1 eTWSPLMNInfoTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv

#### 8.1299.2.2 sMSEtwsMessageTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv

#### 8.1299.2.3 sMSOnIMSTlv unpack_sms_SetNewSMSCallback_ind_t::IMSTlv

#### 8.1299.2.4 messageModeTlv unpack_sms_SetNewSMSCallback_ind_t::MMTlv

#### 8.1299.2.5 newMTMessageTlv unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv

#### 8.1299.2.6 sMSCAddressTlv unpack_sms_SetNewSMSCallback_ind_t::SMSCTlv

#### 8.1299.2.7 transferRouteMessageTlv unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv

## 8.1300 unpack_sms_SetNewSMSCallback_t Struct Reference

### 8.1300.1 Detailed Description

This structure contains unpack set new SMS callback.

**Parameters**

| | |
|---|---|
| *NULL* | dummy prototype. |

## 8.1301 unpack_sms_SLQSDeleteSMS_t Struct Reference

### 8.1301.1 Detailed Description

This structure contains unpack delete SMS parameters.

**Parameters**

| | |
|---|---|
| *NULL* | unpack prototype. |

## 8.1302 unpack_sms_SLQSGetIndicationRegister_t Struct Reference

### Data Fields

- sms_getIndicationReg * pGetIndicationRegInfo

### 8.1302.1 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 |

### 8.1302.2 Field Documentation

#### 8.1302.2.1 sms_getIndicationReg* unpack_sms_SLQSGetIndicationRegister_t::pGetIndicationRegInfo

## 8.1303 unpack_sms_SLQSGetMessageWaiting_t Struct Reference

### Data Fields

- sms_getMsgWaitingInfo * pGetMsgWaitingInfoResp

### 8.1303.1 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>• NULL pointer - Invalid data. |

### 8.1303.2 Field Documentation

#### 8.1303.2.1 sms_getMsgWaitingInfo* unpack_sms_SLQSGetMessageWaiting_t::pGetMsgWaitingInfoResp

## 8.1304 unpack_sms_SLQSGetSMS_t Struct Reference

### Data Fields

- uint32_t messageTag
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t message [2048]

### 8.1304.1 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 |
| *messageFormat* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *messageSize* | • Upon input the maximum number of bytes that can be written to the message array. |

- Upon successful output the actual number of bytes written to the message array.

**Parameters**

| | |
|---|---|
| *message* | • The message contents array |

### 8.1304.2 Field Documentation

#### 8.1304.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[2048]

#### 8.1304.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

#### 8.1304.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

#### 8.1304.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

## 8.1305 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

### Data Fields

- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

### 8.1305.1 Detailed Description

Structure contain parameters that Provides Information about the SMS BroadcastConfiguration

**Parameters**

| | |
|---|---|
| *pBroadcast-Config* | • The data for 3GPP Broadcast Information(Optional).<br>• NULL pointer - Invalid data. |
| *pCDMA-BroadcastConfig* | • The data for 3GPP2 Broadcast Information(Optional).<br>• NULL pointer - Invalid data. |

### 8.1305.2 Field Documentation

#### 8.1305.2.1 sms_qaQmi3GPPBroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pBroadcastConfig

#### 8.1305.2.2 sms_qaQmi3GPP2BroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pCDMABroadcast-Config

## 8.1306 unpack_sms_SLQSGetSMSList_t Struct Reference

### Data Fields

- uint32_t messageListSize
- qmiSmsMessageList messageList [255]

### 8.1306.1 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. |
| *messageList* | • Message List<br>• See qmiSmsMessageList for more information |

### 8.1306.2 Field Documentation

#### 8.1306.2.1 qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]

#### 8.1306.2.2 uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize

## 8.1307 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference

### Data Fields

- sms_getTransLayerInfo * pGetTransLayerInfo

### 8.1307.1 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 |

### 8.1307.2 Field Documentation

#### 8.1307.2.1 sms_getTransLayerInfo* unpack_sms_SLQSGetTransLayerInfo_t::pGetTransLayerInfo

## 8.1308 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference

### Data Fields

- sms_getTransNWRegInfo * pGetTransNWRegInfo

### 8.1308.1 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 |

### 8.1308.2 Field Documentation

#### 8.1308.2.1 sms_getTransNWRegInfo* unpack_sms_SLQSGetTransNWRegInfo_t::pGetTransNWRegInfo

## 8.1309 unpack_sms_SLQSModifySMSStatus_t Struct Reference

### 8.1309.1 Detailed Description

This structure contains unpack modify SMS status parameters.

**Parameters**

| | |
|---|---|
| *NULL* | unpack prototype. |

## 8.1310 unpack_sms_SLQSNWRegInfoCallback_ind_t Struct Reference

### Data Fields

- uint8_t NWRegStat

### 8.1310.1 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 |

**Note**

None

### 8.1310.2 Field Documentation

#### 8.1310.2.1 uint8_t unpack_sms_SLQSNWRegInfoCallback_ind_t::NWRegStat

## 8.1311 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

### Data Fields

- sms_maxStorageSizeResp * pMaxStorageSizeResp

### 8.1311.1 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>• NULL pointer - Invalid data. |

### 8.1311.2 Field Documentation

#### 8.1311.2.1 sms_maxStorageSizeResp∗ unpack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeResp

## 8.1312 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference

### Data Fields

- sms_msgProtocolResp ∗ pMessageProtocol

### 8.1312.1 Detailed Description

Structure contain Parameters that get the message protocol currently in use for the WMS client.

**Parameters**

| | |
|---|---|
| *pMessage-Protocol* | • Pointer to smsMsgprotocolResp<br>– See sms_msgProtocolResp for more information |

### 8.1312.2 Field Documentation

#### 8.1312.2.1 sms_msgProtocolResp∗ unpack_sms_SLQSSmsGetMessageProtocol_t::pMessageProtocol

## 8.1313 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference

### Data Fields

- uint32_t ∗ pSmsStorage

### 8.1313.1 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>• NULL pointer - Invalid data. |

### 8.1313.2 Field Documentation

#### 8.1313.2.1 uint32_t* unpack_sms_SLQSSwiGetSMSStorage_t::pSmsStorage

## 8.1314 unpack_sms_SLQSTransLayerInfoCallback_ind_t Struct Reference

### Data Fields

- uint8_t regInd
- sms_transLayerInfo * pTransLayerInfo

### 8.1314.1 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 |
| *pTransLayerInfo* | • Optional parameter<br>• NULL when response Transport layer information TLV field absent<br>• See sms_transLayerInfo for more information |

**Note**

None

### 8.1314.2 Field Documentation

#### 8.1314.2.1 sms_transLayerInfo* unpack_sms_SLQSTransLayerInfoCallback_ind_t::pTransLayerInfo

#### 8.1314.2.2 uint8_t unpack_sms_SLQSTransLayerInfoCallback_ind_t::regInd

## 8.1315 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

### 8.1315.1 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 |
| *messageID* | • Unique ID assigned by WMS for non-retry messages. |
| *causeCode* | • WMS cause code |
| *errorClass* | • Error Class<br>• Values:<br>– 0x00 - ERROR_CLASS_TEMPORARY<br>– 0x01 - ERROR_CLASS_PERMANENT |
| *RPCause* | • GW RP cause |
| *TPCause* | • GW TP Cause |
| *msgDelFailure-Type* | • Message delivery failure type<br>• Values:<br>– 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY<br>– 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT |
| *msgDelFailure-Cause* | • Message delivery failure cause<br>• Values:<br>– 0x00 - WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL |
| *alphaIDLen* | • Number of sets of the pAlphaID |
| *pAlphaID* | • Alpha ID<br>• NULL when Alpha ID information TLV field absent |
| *userData* | • Identifies the request associated with this indication. |

### 8.1315.2 Field Documentation

#### 8.1315.2.1 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::alphaIDLen

#### 8.1315.2.2 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::causeCode

#### 8.1315.2.3 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::errorClass

#### 8.1315.2.4 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::messageID

#### 8.1315.2.5 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::msgDelFailureCause

#### 8.1315.2.6 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::msgDelFailureType

#### 8.1315.2.7 uint8_t* unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::pAlphaID

#### 8.1315.2.8 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::RPCause

#### 8.1315.2.9 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::sendStatus

#### 8.1315.2.10 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::TPCause

#### 8.1315.2.11 uint32_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::userData

## 8.1316 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference

### Data Fields

- uint8_t storageType
- uint8_t messageMode

### 8.1316.1 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 |
| *messageMode* | • 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

### 8.1316.2 Field Documentation

#### 8.1316.2.1 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode

#### 8.1316.2.2 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType

## 8.1317 unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t Struct Reference

### Data Fields

- uint8_t numInstances
- sms_messageWaitingInfoContent msgWaitInfo [0xFF]

### 8.1317.1 Detailed Description

This structure holds information related to message waiting information indication

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of sets of the elements in structure sms_messageWaitingInfoContent |
| *msgWaitInfo* | • Pointer to structure of sms_messageWaitingInfoContent.<br>– See sms_messageWaitingInfoContent for more information. |

### 8.1317.2 Field Documentation

#### 8.1317.2.1 sms_messageWaitingInfoContent unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::msgWaitInfo[0xFF]

#### 8.1317.2.2 uint8_t unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::numInstances

## 8.1318 unpack_swidms_SLQSSwiDmsGetMTU_t Struct Reference

### Data Fields

- swidms_mtuSize3gppTlv * pMTUSize3gpp
- swidms_hrpdMTUSizeTlv * pHrpdMTUSize
- swidms_ehrpdMTUSizeTlv * pEhrpdMTUSize
- swidms_usbMTUSizeTlv * pUsbMTUSize

### 8.1318.1 Detailed Description

This structure contains the Get MTU Response parameter.

**Parameters**

| | |
|---|---|
| *pMTUSize3gpp* | [Optional]<br>• See swidms_mtuSize3gppTlv for more information |
| *pHrpdMTUSize* | [Optional]<br>• See swidms_hrpdMTUSizeTlv for more information |
| *pEhrpdMTUSize* | [Optional]<br>• See swidms_ehrpdMTUSizeTlv for more information |

| | |
|---|---|
| *pUsbMTUSize* | [Optional]<br>• See swidms_usbMTUSizeTlv for more information |

### 8.1318.2 Field Documentation

#### 8.1318.2.1 swidms_ehrpdMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pEhrpdMTUSize

#### 8.1318.2.2 swidms_hrpdMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pHrpdMTUSize

#### 8.1318.2.3 swidms_mtuSize3gppTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pMTUSize3gpp

#### 8.1318.2.4 swidms_usbMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pUsbMTUSize

## 8.1319 unpack_swidms_SLQSSwiDmsGetUsbNetNum_t Struct Reference

### Data Fields

- uint8_t usbNetNum

### 8.1319.1 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 |

### 8.1319.2 Field Documentation

#### 8.1319.2.1 uint8_t unpack_swidms_SLQSSwiDmsGetUsbNetNum_t::usbNetNum

## 8.1320 unpack_swidms_SLQSSwiDmsSetMTU_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1320.1 Detailed Description

This structure is used to store unpack_swidms_SLQSSwiDmsSetMTU_t parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |

### 8.1320.2 Field Documentation

#### 8.1320.2.1 uint16_t unpack_swidms_SLQSSwiDmsSetMTU_t::TlvResult

## 8.1321 unpack_swiloc_SwiLocGetAutoStart_t Struct Reference

### Data Fields

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

### 8.1321.1 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>– 0xff - not reported by modem |
| *function_reported* | • 0 - not reported by modem<br>• l - 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 |
| *fix_type_reported* | • 0 - not reported by modem<br>• l - reported by modem |
| *max_time* | • Maximum time allowed for the receiver to get a fix in seconds<br>• Valid range: 1-255 |

| | |
|---|---|
| *max_time_reported* | • 0 - not reported by modem<br>• l - reported by modem |
| *max_dist* | • Maximum uncertainty of a fix measured by distance in meters<br>• Valid range: 1 - 4294967280 |
| *max_dist_reported* | • 0 - not reported by modem<br>• l - reported by modem |
| *fix_rate* | • Time between fixes in seconds<br>• Valid range: 1–65535 |
| *fix_rate_reported* | • 0 - not reported by modem<br>• l - reported by modem |

### 8.1321.2 Field Documentation

#### 8.1321.2.1 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate

#### 8.1321.2.2 int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported

#### 8.1321.2.3 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type

#### 8.1321.2.4 int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported

#### 8.1321.2.5 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function

#### 8.1321.2.6 int unpack_swiloc_SwiLocGetAutoStart_t::function_reported

#### 8.1321.2.7 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist

#### 8.1321.2.8 int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported

#### 8.1321.2.9 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time

#### 8.1321.2.10 int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported

## 8.1322 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference

### Data Fields

- uint32_t eventType
- unpack_omaDmFotaTlv_t SessionInfoFota
- unpack_omaDmConfigTlv_t SessionInfoConfig
- unpack_omaDmNotificationsTlv_t SessionInfoNotification

### 8.1322.1 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. |
| *SessionInfo-Fota[OUT]* | • See unpack_omaDmFotaTlv_t for more information |
| *SessionInfo-Config[OUT]* | • See unpack_omaDmConfigTlv_t for more information |
| *SessionInfo-Notification[OUT]* | • See unpack_omaDmNotificationsTlv_t for more information |

### 8.1322.2 Field Documentation

#### 8.1322.2.1 uint32_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::eventType

#### 8.1322.2.2 unpack_omaDmConfigTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoConfig

#### 8.1322.2.3 unpack_omaDmFotaTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoFota

#### 8.1322.2.4 unpack_omaDmNotificationsTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfo-Notification

## 8.1323 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

### 8.1323.1 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 |

### 8.1323.2 Field Documentation

#### 8.1323.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

#### 8.1323.2.2 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength

#### 8.1323.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength

#### 8.1323.2.4 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]

#### 8.1323.2.5 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgName[255]

#### 8.1323.2.6 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgNameLength

#### 8.1323.2.7 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::RetryCount

#### 8.1323.2.8 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionState

#### 8.1323.2.9 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

#### 8.1323.2.10 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Severity

#### 8.1323.2.11 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Source[255]

#### 8.1323.2.12 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SourceLength

#### 8.1323.2.13 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Status

#### 8.1323.2.14 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Time[255]

#### 8.1323.2.15 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::TimeLength

#### 8.1323.2.16 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus

## 8.1324 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference

### Data Fields

- uint32_t OMADMEnabled
- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t Autosdm
- uint8_t FwAutoCheck

### 8.1324.1 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. |

### 8.1324.2 Field Documentation

#### 8.1324.2.1 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::Autosdm

#### 8.1324.2.2 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAdownload

#### 8.1324.2.3 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAUpdate

#### 8.1324.2.4 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FwAutoCheck

#### 8.1324.2.5 uint32_t unpack_swioma_SLQSOMADMGetSettings_t::OMADMEnabled

## 8.1325 unpack_swioma_SLQSOMADMStartSession_t Struct Reference

### Data Fields

- uint32_t FwAvailability

### 8.1325.1 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. |

### 8.1325.2 Field Documentation

#### 8.1325.2.1 uint32_t unpack_swioma_SLQSOMADMStartSession_t::FwAvailability

## 8.1326 unpack_uim_ChangePin_t Struct Reference

### Data Fields

- uim_remainingRetries * pRemainingRetries
- uim_encryptedPIN1 * pEncryptedPIN1
- uint32_t * pIndicationToken
- uint16_t TlvResult

### 8.1326.1 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.

### 8.1326.2 Field Documentation

#### 8.1326.2.1 uim_encryptedPIN1* unpack_uim_ChangePin_t::pEncryptedPIN1

#### 8.1326.2.2 uint32_t* unpack_uim_ChangePin_t::pIndicationToken

#### 8.1326.2.3 uim_remainingRetries* unpack_uim_ChangePin_t::pRemainingRetries

#### 8.1326.2.4 uint16_t unpack_uim_ChangePin_t::Tlvresult

## 8.1327 unpack_uim_GetCardStatus_t Struct Reference

### Data Fields

- uim_cardStatus * pCardStatus
- uim_hotSwapStatus * pHotSwapStatus
- uint16_t Tlvresult

### 8.1327.1 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.

### 8.1327.2 Field Documentation

#### 8.1327.2.1 uim_cardStatus* unpack_uim_GetCardStatus_t::pCardStatus

#### 8.1327.2.2 uim_hotSwapStatus* unpack_uim_GetCardStatus_t::pHotSwapStatus

#### 8.1327.2.3 uint16_t unpack_uim_GetCardStatus_t::Tlvresult

## 8.1328 unpack_uim_ReadTransparent_t Struct Reference

### Data Fields

- uim_cardResult * pCardResult
- uim_readResult * pReadResult
- uint32_t * pIndicationToken
- uint8_t * pEncryptedData
- uint16_t Tlvresult

### 8.1328.1 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.

### 8.1328.2 Field Documentation

#### 8.1328.2.1 uim_cardResult* unpack_uim_ReadTransparent_t::pCardResult

#### 8.1328.2.2 uint8_t* unpack_uim_ReadTransparent_t::pEncryptedData

#### 8.1328.2.3 uint32_t* unpack_uim_ReadTransparent_t::pIndicationToken

#### 8.1328.2.4 uim_readResult* unpack_uim_ReadTransparent_t::pReadResult

#### 8.1328.2.5 uint16_t unpack_uim_ReadTransparent_t::Tlvresult

## 8.1329 unpack_uim_SetPinProtection_t Struct Reference

### Data Fields

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

### 8.1329.1 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.

### 8.1329.2 Field Documentation

#### 8.1329.2.1 uim_encryptedPIN1∗ unpack_uim_SetPinProtection_t::pEncryptedPIN1

#### 8.1329.2.2 uint32_t∗ unpack_uim_SetPinProtection_t::pIndicationToken

#### 8.1329.2.3 uim_remainingRetries∗ unpack_uim_SetPinProtection_t::pRemainingRetries

#### 8.1329.2.4 uint16_t unpack_uim_SetPinProtection_t::TlvresultI

## 8.1330 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference

### Data Fields

- slots_t slotsstatusChange
- uint8_t bNumberOfPhySlots

### 8.1330.1 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. |

### 8.1330.2 Field Documentation

#### 8.1330.2.1 uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots

#### 8.1330.2.2 slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange

## 8.1331 unpack_uim_SLQSUIMAuthenticate_t Struct Reference

### Data Fields

- uim_cardResult * pCardResult
- uim_authenticateResult * pAuthenticateResult
- uint32_t * pIndicationToken

### 8.1331.1 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.

### 8.1331.2 Field Documentation

#### 8.1331.2.1 uim_authenticateResult* unpack_uim_SLQSUIMAuthenticate_t::pAuthenticateResult

#### 8.1331.2.2 uim_cardResult* unpack_uim_SLQSUIMAuthenticate_t::pCardResult

#### 8.1331.2.3 uint32_t* unpack_uim_SLQSUIMAuthenticate_t::pIndicationToken

## 8.1332 unpack_uim_SLQSUIMDepersonalization_t Struct Reference

### Data Fields

- uim_remainingRetries * pRemainingRetries

### 8.1332.1 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.

### 8.1332.2 Field Documentation

#### 8.1332.2.1 uim_remainingRetries* unpack_uim_SLQSUIMDepersonalization_t::pRemainingRetries

## 8.1333 unpack_uim_SLQSUIMEventRegister_t Struct Reference

### Data Fields

- uint32_t eventMask

### 8.1333.1 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. |

### 8.1333.2 Field Documentation

#### 8.1333.2.1 uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask

## 8.1334 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference

### Data Fields

- uint8_t * pAutoSelection
- uim_personalizationStatus * pPersonalizationStatus
- uint8_t * pHaltSubscription

### 8.1334.1 Detailed Description

This structure contains information of the response parameters associated with Get Configuration.

**Parameters**

| | |
|---|---|
| *pAuto-Selection(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. |
| *p-Personalization-Status(optional)* | • See uim_personalizationStatus for more information.<br>• NULL pointer - Invalid data. |
| *pHalt-Subscription(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. |

### 8.1334.2 Field Documentation

#### 8.1334.2.1 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pAutoSelection

#### 8.1334.2.2 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pHaltSubscription

#### 8.1334.2.3 uim_personalizationStatus* unpack_uim_SLQSUIMGetConfiguration_t::pPersonalizationStatus

## 8.1335 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference

### Data Fields

- uim_cardResult * pCardResult
- uim_fileAttributes * pFileAttributes
- uint32_t * pIndicationToken

### 8.1335.1 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.

### 8.1335.2 Field Documentation

#### 8.1335.2.1 uim_cardResult* unpack_uim_SLQSUIMGetFileAttributes_t::pCardResult

#### 8.1335.2.2 uim_fileAttributes* unpack_uim_SLQSUIMGetFileAttributes_t::pFileAttributes

#### 8.1335.2.3 uint32_t* unpack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

## 8.1336 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference

### Data Fields

- uint8_t * pNumberOfPhySlot
- slots_t * pUimSlotsStatus

### 8.1336.1 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.. |

### 8.1336.2 Field Documentation

#### 8.1336.2.1 uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot

#### 8.1336.2.2 slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus

## 8.1337 unpack_uim_SLQSUIMRefreshCallback_Ind_t Struct Reference

### Data Fields

- uint8_t TlvPresent
- uim_refreshevent refreshEvent

### 8.1337.1 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 |

### 8.1337.2 Field Documentation

#### 8.1337.2.1 uim_refreshevent unpack_uim_SLQSUIMRefreshCallback_Ind_t::refreshEvent

#### 8.1337.2.2 uint8_t unpack_uim_SLQSUIMRefreshCallback_Ind_t::TlvPresent

## 8.1338 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

### Data Fields

- uim_refreshevent * pRefreshEvent

### 8.1338.1 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. |

### 8.1338.2 Field Documentation

#### 8.1338.2.1 uim_refreshevent* unpack_uim_SLQSUIMRefreshGetLastEvent_t::pRefreshEvent

## 8.1339 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference

### Data Fields

- uim_cardStatus * pCardStatus

### 8.1339.1 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. |

### 8.1339.2 Field Documentation

#### 8.1339.2.1 uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus

## 8.1340 unpack_uim_UnblockPin_t Struct Reference

### Data Fields

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

### 8.1340.1 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. |

### 8.1340.2 Field Documentation

#### 8.1340.2.1 uim_encryptedPIN1* unpack_uim_UnblockPin_t::pEncryptedPIN1

#### 8.1340.2.2 uint32_t* unpack_uim_UnblockPin_t::pIndicationToken

#### 8.1340.2.3 uim_remainingRetries* unpack_uim_UnblockPin_t::pRemainingRetries

#### 8.1340.2.4 uint16_t unpack_uim_UnblockPin_t::Tlvresult

## 8.1341 unpack_uim_VerifyPin_t Struct Reference

### Data Fields

- uim_remainingRetries * pRemainingRetries
- uim_encryptedPIN1 * pEncryptedPIN1
- uint32_t * pIndicationToken
- uint16_t TlvResult

#### 8.1341.1 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.

#### 8.1341.2 Field Documentation

##### 8.1341.2.1 uim_encryptedPIN1* unpack_uim_VerifyPin_t::pEncryptedPIN1

##### 8.1341.2.2 uint32_t* unpack_uim_VerifyPin_t::pIndicationToken

##### 8.1341.2.3 uim_remainingRetries* unpack_uim_VerifyPin_t::pRemainingRetries

##### 8.1341.2.4 uint16_t unpack_uim_VerifyPin_t::TlvResult

### 8.1342 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

### 8.1342.1 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. |

### 8.1342.2 Field Documentation

#### 8.1342.2.1 voice_arrCallInfo unpack_voice_allCallStatusCallback_ind_t::arrCallInfomation

#### 8.1342.2.2 voice_arrAlertingPattern* unpack_voice_allCallStatusCallback_ind_t::pArrAlertingPattern

#### 8.1342.2.3 voice_arrAlertingType* unpack_voice_allCallStatusCallback_ind_t::pArrAlertingType

#### 8.1342.2.4 voice_arrAlphaID* unpack_voice_allCallStatusCallback_ind_t::pArrAlphaID

#### 8.1342.2.5 voice_arrCalledPartyNum* unpack_voice_allCallStatusCallback_ind_t::pArrCalledPartyNum

#### 8.1342.2.6 voice_arrCallEndReason* unpack_voice_allCallStatusCallback_ind_t::pArrCallEndReason

#### 8.1342.2.7 voice_arrConnectPartyNum* unpack_voice_allCallStatusCallback_ind_t::pArrConnectPartyNum

#### 8.1342.2.8 voice_arrDiagInfo* unpack_voice_allCallStatusCallback_ind_t::pArrDiagInfo

#### 8.1342.2.9 voice_arrRedirPartyNum* unpack_voice_allCallStatusCallback_ind_t::pArrRedirPartyNum

#### 8.1342.2.10 voice_arrRemotePartyName* unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyName

#### 8.1342.2.11 voice_arrRemotePartyNum* unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyNum

#### 8.1342.2.12 voice_arrSvcOption* unpack_voice_allCallStatusCallback_ind_t::pArrSvcOption

## 8.1343 unpack_voice_DTMFEventCallback_ind_t Struct Reference

### Data Fields

- voice_DTMFInfo DTMFInformation
- uint8_t * pOnLength
- uint8_t * pOffLength

### 8.1343.1 Detailed Description

This structure contains the parameters passed for DTMF event indication by the device.

**Parameters**

| | |
|---|---|
| *DTMFInformation(mandatory)* | See voice_DTMFInfo for more information. |
| *pOnLength(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 |
| *pOffLength(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

### 8.1343.2 Field Documentation

#### 8.1343.2.1 voice_DTMFInfo unpack_voice_DTMFEventCallback_ind_t::DTMFInformation

#### 8.1343.2.2 uint8_t* unpack_voice_DTMFEventCallback_ind_t::pOffLength

#### 8.1343.2.3 uint8_t* unpack_voice_DTMFEventCallback_ind_t::pOnLength

## 8.1344 unpack_voice_OTASPStatusCallback_ind_t Struct Reference

### Data Fields

- uint8_t callID
- uint8_t OTASPStatus

### 8.1344.1 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) |

### 8.1344.2 Field Documentation

#### 8.1344.2.1 uint8_t unpack_voice_OTASPStatusCallback_ind_t::callID

#### 8.1344.2.2 uint8_t unpack_voice_OTASPStatusCallback_ind_t::OTASPStatus

## 8.1345 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

### 8.1345.1 Detailed Description

This structure contains the parameters of USS response.

Parameters

| | |
|---|---|
| *pfailureCause* | • Failure Cause.<br>• 0 - QMI_FAILURE_CAUSE_OFFLINE - Phone is offline<br>• 20 - QMI_FAILURE_CAUSE_CDMA_LOCK - Phone is CDMA locked until a power cycle; CDMA only<br>• 21 - QMI_FAILURE_CAUSE_NO_SRV - Phone has no service<br>• 22 - QMI_FAILURE_CAUSE_FADE - Call has ended abnormally<br>• 23 - QMI_FAILURE_CAUSE_INTERCEPT - Received intercept from the base station; originating only; CDMA only<br>• 24 - QMI_FAILURE_CAUSE_REORDER - Received reorder from the base station; originating only; CDMA only<br>• 25 - QMI_FAILURE_CAUSE_REL_NORMAL - Received release from the base station; no reason was given<br>• 26 - QMI_FAILURE_CAUSE_REL_SO_REJ - Received release from the base station; SO reject; CDMA only<br>• 27 - QMI_FAILURE_CAUSE_INCOM_CALL - Received incoming call from the base station<br>• 28 - QMI_FAILURE_CAUSE_ALERT_STOP - Received alert stop from the base station; incoming only; CDMA only<br>• 29 - QMI_FAILURE_CAUSE_CLIENT_END - Client ended the call<br>• 30 - QMI_FAILURE_CAUSE_ACTIVATION - Received end activation; OTASP call only; CDMA only<br>• 31 - QMI_FAILURE_CAUSE_MC_ABORT - MC aborted the origination/conversation; CDMA only<br>• 32 - QMI_FAILURE_CAUSE_MAX_ACCESS_PROBE - Maximum access probes were transmitted; CDMA only<br>• 33 - QMI_FAILURE_CAUSE_PSIST_N - Persistence test failure; FEATURE_JCDMA only; CDMA only<br>• 34 - QMI_FAILURE_CAUSE_UIM_NOT_PRESENT - R-UIM is not present<br>• 35 - QMI_FAILURE_CAUSE_ACC_IN_PROG - Access attempt is already in progress<br>• 36 - QMI_FAILURE_CAUSE_ACC_FAIL - Access failure for a reason other than the above<br>• 37 - QMI_FAILURE_CAUSE_RETRY_ORDER - Received retry order; originating only; IS 2000; CDMA only<br>• 38 - QMI_FAILURE_CAUSE_CCS_NOT_SUPPORTED_BYBS - Concurrent service is not supported by the base station<br>• 39 - QMI_FAILURE_CAUSE_NO_RESPONSE_FROM_BS - No response was received from the base station<br>• 40 - QMI_FAILURE_CAUSE_REJECTED_BY_BS - Call was rejected by the base station; CDMA only<br>• 41 - QMI_FAILURE_CAUSE_INCOMPATIBLE - Concurrent services requested were not compatible; CDMA only<br>• 42 - QMI_FAILURE_CAUSE_ACCESS_BLOCK - Access is blocked by the base station; CDMA only<br>• 43 - QMI_FAILURE_CAUSE_ALREADY_IN_TC - Corresponds to CM_CALL_ORIGERR_ALREADY_IN_TC<br>• 44 - QMI_FAILURE_CAUSE_EMERGENCY_FLASHED - Call is ended because an emergency call was flashed over this call; CDMA only<br>• 45 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_GPS - Used if CM is ending a GPS call in preference of a user call<br>• 46 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_SMS - Used if CM is ending an SMS call in preference of a user call<br>• 47 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_DATA - Used if CM is ending a data call in preference of an emergency call<br>• 48 - QMI_FAILURE_CAUSE_REDIR_OR_HANDOFF - Call was rejected because of a redirection or handoff<br>• 49 - QMI_FAILURE_CAUSE_ACCESS_BLOCK_ALL - Access is blocked by the base station for all mobiles; KDDI-specific; CDMA only<br>• 50 - QMI_FAILURE_CAUSE_OTASP_SPC_ERR - To support OTASP SPC Error indication<br>• 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. |

### 8.1345.2 Field Documentation

#### 8.1345.2.1 voice_alphaIDInfo* unpack_voice_SLQSOriginateUSSD_t::pAlphaIDInfo

#### 8.1345.2.2 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCallId

#### 8.1345.2.3 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCcResultType

#### 8.1345.2.4 voice_ccSUPSType* unpack_voice_SLQSOriginateUSSD_t::pCCSuppsType

#### 8.1345.2.5 uint16_t* unpack_voice_SLQSOriginateUSSD_t::pfailureCause

#### 8.1345.2.6 struct voice_USSInfo* unpack_voice_SLQSOriginateUSSD_t::pUSSDInfo

## 8.1346 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference

### Data Fields

• uint8_t * pCallId

### 8.1346.1 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. |

### 8.1346.2 Field Documentation

#### 8.1346.2.1 uint8_t* unpack_voice_SLQSVoiceAnswerCall_t::pCallId

## 8.1347 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference

### Data Fields

- uint8_t * pCallID

### 8.1347.1 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. |

### 8.1347.2 Field Documentation

#### 8.1347.2.1 uint8_t* unpack_voice_SLQSVoiceBurstDTMF_t::pCallID

## 8.1348 unpack_voice_SLQSVoiceDialCall_t Struct Reference

### Data Fields

- uint8_t * pCallID
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- voice_ccSUPSType * pCCSUPSType

### 8.1348.1 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. |

### 8.1348.2 Field Documentation

#### 8.1348.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceDialCall_t::pAlphaIDInfo

#### 8.1348.2.2 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCallID

#### 8.1348.2.3 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCCResultType

#### 8.1348.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceDialCall_t::pCCSUPSType

## 8.1349 unpack_voice_SLQSVoiceEndCall_t Struct Reference

### Data Fields

- uint8_t * pCallId

### 8.1349.1 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. |

### 8.1349.2 Field Documentation

#### 8.1349.2.1 uint8_t* unpack_voice_SLQSVoiceEndCall_t::pCallId

## 8.1350 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

### 8.1350.1 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. |

### 8.1350.2 Field Documentation

#### 8.1350.2.1 voice_arrAlertingPattern∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingPattern

#### 8.1350.2.2 voice_arrAlertingType∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingType

#### 8.1350.2.3 voice_arrAlphaID∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlphaID

#### 8.1350.2.4 voice_arrCalledPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCalledPartyNum

#### 8.1350.2.5 voice_arrCallEndReason∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallEndReason

#### 8.1350.2.6 voice_arrCallInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallInfo

#### 8.1350.2.7 voice_arrConnectPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrConnectPartyNum

#### 8.1350.2.8 voice_arrDiagInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrDiagInfo

#### 8.1350.2.9 voice_arrRedirPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRedirPartyNum

#### 8.1350.2.10 voice_arrRemotePartyName∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyName

#### 8.1350.2.11 voice_arrRemotePartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyNum

#### 8.1350.2.12 voice_arrSvcOption∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrSvcOption

#### 8.1350.2.13 voice_arrUUSInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrUUSInfo

#### 8.1350.2.14 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pOTASPStatus

#### 8.1350.2.15 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pVoicePrivacy

## 8.1351 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

### 8.1351.1 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.

### 8.1351.2 Field Documentation

#### 8.1351.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallBarring_t::pAlphaIDInfo

#### 8.1351.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCallID

#### 8.1351.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCCResType

#### 8.1351.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallBarring_t::pCCSUPSType

#### 8.1351.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallBarring_t::pFailCause

#### 8.1351.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

## 8.1352 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

### 8.1352.1 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.

### 8.1352.2 Field Documentation

#### 8.1352.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pAlphaIDInfo

#### 8.1352.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCallID

#### 8.1352.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCResType

#### 8.1352.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCSUPSType

#### 8.1352.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pFailCause

#### 8.1352.2.6 voice_getCallFWExtInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWExtInfo

#### 8.1352.2.7 voice_getCallFWInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWInfo

## 8.1353 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

### 8.1353.1 Detailed Description

This structure contains information of the unpack parameters associated with a call.

**Parameters**

| | |
|---|---|
| *pCall-Info(optional)* | • See voice_callInfo for more information.<br>• NULL pointer - Invalid data. |
| *pRemoteParty-Num(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. |

### 8.1353.2 Field Documentation

#### 8.1353.2.1 uint32_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertingPattern

#### 8.1353.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertType

#### 8.1353.2.3 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pAlphaIDInfo

#### 8.1353.2.4 voice_callInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pCallInfo

#### 8.1353.2.5 voice_connectNumInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pConnectNumInfo

#### 8.1353.2.6 voice_diagInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pDiagInfo

#### 8.1353.2.7 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pOTASPStatus

#### 8.1353.2.8 voice_remotePartyName* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyName

#### 8.1353.2.9 voice_remotePartyNum* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyNum

#### 8.1353.2.10 uint16_t* unpack_voice_SLQSVoiceGetCallInfo_t::pSrvOpt

#### 8.1353.2.11 voice_UUSInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pUUSInfo

#### 8.1353.2.12 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pVoicePrivacy

## 8.1354 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

### 8.1354.1 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.

### 8.1354.2 Field Documentation

#### 8.1354.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallWaiting_t::pAlphaIDInfo

#### 8.1354.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCallID

#### 8.1354.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCResType

#### 8.1354.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCSUPSType

#### 8.1354.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pFailCause

#### 8.1354.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

## 8.1355 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

### 8.1355.1 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.

### 8.1355.2 Field Documentation

#### 8.1355.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIP_t::pAlphaIDInfo

#### 8.1355.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCallID

#### 8.1355.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCCResType

#### 8.1355.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIP_t::pCCSUPSType

#### 8.1355.2.5 voice_CLIPResp* unpack_voice_SLQSVoiceGetCLIP_t::pCLIPResp

#### 8.1355.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIP_t::pFailCause

## 8.1356 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

### 8.1356.1 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.

### 8.1356.2 Field Documentation

#### 8.1356.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIR_t::pAlphaIDInfo

#### 8.1356.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCallID

#### 8.1356.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCCResType

#### 8.1356.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIR_t::pCCSUPSType

#### 8.1356.2.5 voice_CLIRResp* unpack_voice_SLQSVoiceGetCLIR_t::pCLIRResp

#### 8.1356.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIR_t::pFailCause

## 8.1357 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

### 8.1357.1 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.

### 8.1357.2 Field Documentation

#### 8.1357.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCNAP_t::pAlphaIDInfo

#### 8.1357.2.2 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCallID

#### 8.1357.2.3 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCCResType

#### 8.1357.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCNAP_t::pCCSUPSType

#### 8.1357.2.5 voice_CNAPResp* unpack_voice_SLQSVoiceGetCNAP_t::pCNAPResp

#### 8.1357.2.6 uint16_t* unpack_voice_SLQSVoiceGetCNAP_t::pFailCause

## 8.1358 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

### 8.1358.1 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.

### 8.1358.2 Field Documentation

#### 8.1358.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLP_t::pAlphaIDInfo

#### 8.1358.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCallID

#### 8.1358.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCCResType

#### 8.1358.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLP_t::pCCSUPSType

#### 8.1358.2.5 voice_COLPResp* unpack_voice_SLQSVoiceGetCOLP_t::pCOLPResp

#### 8.1358.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLP_t::pFailCause

## 8.1359 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

### 8.1359.1 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.

### 8.1359.2 Field Documentation

#### 8.1359.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLR_t::pAlphaIDInfo

#### 8.1359.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCallID

#### 8.1359.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCCResType

#### 8.1359.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLR_t::pCCSUPSType

#### 8.1359.2.5 voice_COLRResp* unpack_voice_SLQSVoiceGetCOLR_t::pCOLRResp

#### 8.1359.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLR_t::pFailCause

## 8.1360 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

### 8.1360.1 Detailed Description

This structure contains Voice Get Configuration Response Parameters.

**Parameters**

| | |
|---|---|
| *pAutoAnswer-Stat(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. |
| *pAirTimer-Cnt(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_CODEC_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.

### 8.1360.2 Field Documentation

#### 8.1360.2.1 voice_airTimer* unpack_voice_SLQSVoiceGetConfig_t::pAirTimerCnt

#### 8.1360.2.2 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pAutoAnswerStat

#### 8.1360.2.3 voice_curAMRConfig* unpack_voice_SLQSVoiceGetConfig_t::pCurAMRConfig

#### 8.1360.2.4 voice_prefVoiceSO* unpack_voice_SLQSVoiceGetConfig_t::pCurPrefVoiceSO

#### 8.1360.2.5 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurrTTYMode

#### 8.1360.2.6 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoiceDomainPref

#### 8.1360.2.7 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoicePrivacyPref

#### 8.1360.2.8 voice_roamTimer* unpack_voice_SLQSVoiceGetConfig_t::pRoamTimerCnt

## 8.1361 unpack_voice_SLQSVoiceManageCalls_t Struct Reference

### Data Fields

- uint16_t * pFailCause

### 8.1361.1 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. |

### 8.1361.2 Field Documentation

#### 8.1361.2.1 uint16_t* unpack_voice_SLQSVoiceManageCalls_t::pFailCause

## 8.1362 unpack_voice_SLQSVoiceSendFlash_t Struct Reference

### Data Fields

- uint8_t * pCallID

### 8.1362.1 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. |

### 8.1362.2 Field Documentation

#### 8.1362.2.1 uint8_t* unpack_voice_SLQSVoiceSendFlash_t::pCallID

## 8.1363 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

### Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.1363.1 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.

### 8.1363.2 Field Documentation

#### 8.1363.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pAlphaIDInfo

#### 8.1363.2.2 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCallID

#### 8.1363.2.3 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCResType

#### 8.1363.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCSUPSType

#### 8.1363.2.5 uint16_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pFailCause

## 8.1364 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

### 8.1364.1 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.

### 8.1364.2 Field Documentation

#### 8.1364.2.1 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAirTimerStatus

#### 8.1364.2.2 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAutoAnsStatus

#### 8.1364.2.3 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSOStatus

#### 8.1364.2.4 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pRoamTimerStatus

#### 8.1364.2.5 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pTTYConfigStatus

#### 8.1364.2.6 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pVoiceDomainPrefStatus

## 8.1365 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference

### Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.1365.1 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. |

### 8.1365.2 Field Documentation

#### 8.1365.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetSUPSService_t::pAlphaIDInfo

#### 8.1365.2.2 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCallID

#### 8.1365.2.3 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCCResultType

#### 8.1365.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetSUPSService_t::pCCSUPSType

#### 8.1365.2.5 uint16_t* unpack_voice_SLQSVoiceSetSUPSService_t::pFailCause

## 8.1366 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference

### Data Fields

- uint8_t * pCallID

### 8.1366.1 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. |

### 8.1366.2 Field Documentation

#### 8.1366.2.1 uint8_t* unpack_voice_SLQSVoiceStartContDTMF_t::pCallID

## 8.1367 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference

### Data Fields

- uint8_t callID

### 8.1367.1 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 |

### 8.1367.2 Field Documentation

#### 8.1367.2.1 uint8_t unpack_voice_SLQSVoiceStopContDTMF_t::callID

## 8.1368 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

### 8.1368.1 Detailed Description

This structure contains the parameters passed for SUPS info indication by the device.

**Parameters**

| | |
|---|---|
| *SUPSInformation(mandatory)* | See voice_SUPSInfo for more information. |
| *pSvcClass(optional)* | • Service class is a combination (sum) of information class constants (optional)<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes. |

| | |
|---|---|
| *pReason(optional)* | • See qaGobiApiTableCallControlReturnReasons.h for return reasons. |
| *pCallFWNum(optional)* | • Call forwarding number to be registered with the network.<br>• ASCII String, NULL terminated. |
| *pCallFWTimerVal(optional)* | • Call Forwarding No Reply Timer.<br>– Range: 5 to 30 in steps of 5. |
| *pUSSInfo(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. |
| *pAlphaIDInfo(optional)* | • See voice_alphaIDInfo for more information. |
| *pCallBarPasswd(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. |
| *pNewPwdData(optional)* | • See voice_newPwdData for more information. |
| *pDataSrc(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. |
| *pFailCause(optional)* | • Supplementary services failure cause.<br>• See qaGobiApiTableVoiceCallEndReasons.h for more information. |
| *pCallFwdInfo(optional)* | • See voice_getCallFWInfo for more information. |
| *pCLIRstatus(optional)* | • See voice_CLIRResp for more information. |
| *pCLIPstatus(optional)* | • See voice_CLIPResp for more information. |
| *pCOLPstatus(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

### 8.1368.2 Field Documentation

#### 8.1368.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pAlphaIDInfo

#### 8.1368.2.2 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallBarPasswd

#### 8.1368.2.3 voice_getCallFWInfo* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFwdInfo

#### 8.1368.2.4 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWNum

#### 8.1368.2.5 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWTimerVal

#### 8.1368.2.6 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallID

#### 8.1368.2.7 voice_CLIPResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIPstatus

#### 8.1368.2.8 voice_CLIRResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIRstatus

#### 8.1368.2.9 voice_CNAPResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCNAPstatus

#### 8.1368.2.10 voice_COLPResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLPstatus

#### 8.1368.2.11 voice_COLRResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLRstatus

#### 8.1368.2.12 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pDataSrc

#### 8.1368.2.13 uint16_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pFailCause

#### 8.1368.2.14 voice_newPwdData* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pNewPwdData

#### 8.1368.2.15 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pReason

#### 8.1368.2.16 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pSvcClass

#### 8.1368.2.17 struct voice_USSInfo* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pUSSInfo

#### 8.1368.2.18 voice_SUPSInfo unpack_voice_SLQSVoiceSUPSCallback_ind_t::SUPSInformation

## 8.1369 unpack_voice_SUPSNotificationCallback_ind_t Struct Reference

### Data Fields

- uint8_t callID

- uint8_t notifType
- uint16_t * pCUGIndex
- voice_ECTNum * pECTNum

### 8.1369.1 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

### 8.1369.2 Field Documentation

#### 8.1369.2.1 uint8_t unpack_voice_SUPSNotificationCallback_ind_t::callID

#### 8.1369.2.2 uint8_t unpack_voice_SUPSNotificationCallback_ind_t::notifType

#### 8.1369.2.3 uint16_t* unpack_voice_SUPSNotificationCallback_ind_t::pCUGIndex

#### 8.1369.2.4 voice_ECTNum* unpack_voice_SUPSNotificationCallback_ind_t::pECTNum

## 8.1370 unpack_voice_USSDNotificationCallback_ind_t Struct Reference

### Data Fields

- uint8_t notification_Type
- voice_USSDNotificationNetworkInfo USSDNotificationNetworkInfo

### 8.1370.1 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) |

### 8.1370.2 Field Documentation

#### 8.1370.2.1 uint8_t unpack_voice_USSDNotificationCallback_ind_t::notification_Type

#### 8.1370.2.2 voice_USSDNotificationNetworkInfo unpack_voice_USSDNotificationCallback_ind_t::USSDNotificationNetworkInfo

## 8.1371 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

### 8.1371.1 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 |

### 8.1371.2 Field Documentation

#### 8.1371.2.1 uint8_t unpack_voice_VoiceInfoRecCallback_ind_t::callID

#### 8.1371.2.2 voice_calledPartyInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCalledPartyInfo

#### 8.1371.2.3 voice_callerIDInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerIDInfo

#### 8.1371.2.4 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerNameInfo

#### 8.1371.2.5 voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallingPartyInfo

#### 8.1371.2.6 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallWaitInd

#### 8.1371.2.7 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCLIRCause

#### 8.1371.2.8 voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pConnectNumInfo

#### 8.1371.2.9 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pDispInfo

#### 8.1371.2.10 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDispInfo

#### 8.1371.2.11 voice_extDispRecInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDispRecInfo

#### 8.1371.2.12 voice_lineCtrlInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pLineCtrlInfo

#### 8.1371.2.13 voice_NSSAudioCtrl* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSAudioCtrl

#### 8.1371.2.14 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSRelease

#### 8.1371.2.15 voice_redirNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pRedirNumInfo

#### 8.1371.2.16 voice_signalInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pSignalInfo

## 8.1372 unpack_voice_voicePrivacyChangeCallback_ind_t Struct Reference

### Data Fields

- uint8_t callID
- uint8_t voicePrivacy

### 8.1372.1 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

### 8.1372.2 Field Documentation

#### 8.1372.2.1 uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::callID

#### 8.1372.2.2 uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::voicePrivacy

## 8.1373 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference

### Data Fields

- wds_DHCPProfileIdTlv ProfileIdTlv
- wds_DHCPLeaseStateTlv DHCPv4LeaseStateTlv
- wds_IPv4AdTlv IPv4AddrTlv
- wds_DHCPLeaseOptTlv DHCPv4LeaseOptTlv

### 8.1373.1 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 |

### 8.1373.2 Field Documentation

#### 8.1373.2.1 wds_DHCPLeaseOptTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseOptTlv

#### 8.1373.2.2 wds_DHCPLeaseStateTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseStateTlv

#### 8.1373.2.3 wds_IPv4AdTlv unpack_wds_DHCPv4ClientLease_ind_t::IPv4AddrTlv

#### 8.1373.2.4 wds_DHCPProfileIdTlv unpack_wds_DHCPv4ClientLease_ind_t::ProfileIdTlv

## 8.1374 unpack_wds_GetAutoconnect_t Struct Reference

### Data Fields

- uint32_t * psetting

### 8.1374.1 Detailed Description

auto connect data session setting parameter.

**Parameters**

| | |
|---|---|
| *pSetting* | • NDIS auto connect setting<br>– 0 - Disabled<br>– 1 - Enabled |

### 8.1374.2 Field Documentation

#### 8.1374.2.1 uint32_t* unpack_wds_GetAutoconnect_t::psetting

## 8.1375 unpack_wds_GetByteTotals_t Struct Reference

### Data Fields

- uint64_t * pTXTotalBytes
- uint64_t * pRXTotalBytes

### 8.1375.1 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. |

### 8.1375.2 Field Documentation

#### 8.1375.2.1 uint64_t* unpack_wds_GetByteTotals_t::pRXTotalBytes

#### 8.1375.2.2 uint64_t* unpack_wds_GetByteTotals_t::pTXTotalBytes

## 8.1376 unpack_wds_GetConnectionRate_t Struct Reference

### Data Fields

- uint32_t currentChannelTXRate
- uint32_t currentChannelRXRate
- uint32_t maxChannelTXRate
- uint32_t maxChannelRXRate

### 8.1376.1 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. |

### 8.1376.2 Field Documentation

#### 8.1376.2.1 uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate

#### 8.1376.2.2 uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate

#### 8.1376.2.3 uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate

#### 8.1376.2.4 uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate

## 8.1377 unpack_wds_GetDataBearerTechnology_t Struct Reference

### Data Fields

- uint32_t * pDataBearer

### 8.1377.1 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 |

### 8.1377.2 Field Documentation

#### 8.1377.2.1 uint32_t* unpack_wds_GetDataBearerTechnology_t::pDataBearer

## 8.1378 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]

### 8.1378.1 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 |

### 8.1378.2 Field Documentation

#### 8.1378.2.1 int8_t unpack_wds_GetDefaultProfile_t::apnname[255]

#### 8.1378.2.2 uint8_t unpack_wds_GetDefaultProfile_t::apnsize

#### 8.1378.2.3 uint32_t unpack_wds_GetDefaultProfile_t::auth

#### 8.1378.2.4 uint32_t unpack_wds_GetDefaultProfile_t::ipaddr

#### 8.1378.2.5 uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6

#### 8.1378.2.6 int8_t unpack_wds_GetDefaultProfile_t::name[255]

#### 8.1378.2.7 uint8_t unpack_wds_GetDefaultProfile_t::namesize

#### 8.1378.2.8 uint32_t unpack_wds_GetDefaultProfile_t::pdptype

#### 8.1378.2.9 uint32_t unpack_wds_GetDefaultProfile_t::pridns

#### 8.1378.2.10 uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6

#### 8.1378.2.11 uint32_t unpack_wds_GetDefaultProfile_t::secdns

#### 8.1378.2.12 uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6

#### 8.1378.2.13 int8_t unpack_wds_GetDefaultProfile_t::username[255]

#### 8.1378.2.14 uint8_t unpack_wds_GetDefaultProfile_t::usersize

## 8.1379 unpack_wds_GetDefaultProfileNum_t Struct Reference

### Data Fields

- uint8_t index

### 8.1379.1 Detailed Description

This structure contains unpack Get Default Profile Number information.

**Parameters**

| | |
|---|---|
| *index* | profile index |

### 8.1379.2 Field Documentation

#### 8.1379.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index

## 8.1380 unpack_wds_GetDormancyState_t Struct Reference

### Data Fields

- uint32_t dormancyState

### 8.1380.1 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 |

### 8.1380.2 Field Documentation

#### 8.1380.2.1 uint32_t unpack_wds_GetDormancyState_t::dormancyState

## 8.1381 unpack_wds_GetLastMobileIPError_t Struct Reference

### Data Fields

- uint32_t error

### 8.1381.1 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 |

### 8.1381.2 Field Documentation

#### 8.1381.2.1 uint32_t unpack_wds_GetLastMobileIPError_t::error

## 8.1382 unpack_wds_GetMobileIP_t Struct Reference

### Data Fields

- uint32_t mipMode

### 8.1382.1 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 |

### 8.1382.2 Field Documentation

#### 8.1382.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

## 8.1383 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

### 8.1383.1 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 |

### 8.1383.2 Field Documentation

#### 8.1383.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

#### 8.1383.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAAState

#### 8.1383.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address

#### 8.1383.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled

#### 8.1383.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI

#### 8.1383.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HAState

#### 8.1383.2.7 int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]

#### 8.1383.2.8 uint8_t unpack_wds_GetMobileIPProfile_t::naiSize

#### 8.1383.2.9 uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA

#### 8.1383.2.10 uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling

#### 8.1383.2.11 uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA

## 8.1384 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

### 8.1384.1 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. |

### 8.1384.2 Field Documentation

#### 8.1384.2.1 uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount

#### 8.1384.2.2 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount

#### 8.1384.2.3 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOKBytesLastCall

#### 8.1384.2.4 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors

#### 8.1384.2.5 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows

#### 8.1384.2.6 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses

#### 8.1384.2.7 uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount

#### 8.1384.2.8 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount

#### 8.1384.2.9 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOKBytesLastCall

#### 8.1384.2.10 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors

#### 8.1384.2.11 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows

#### 8.1384.2.12 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses

## 8.1385 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

### 8.1385.1 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. |

### 8.1385.2 Field Documentation

#### 8.1385.2.1 uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount

#### 8.1385.2.2 uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount

#### 8.1385.2.3 uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall

#### 8.1385.2.4 uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors

#### 8.1385.2.5 uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows

#### 8.1385.2.6 uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses

#### 8.1385.2.7 uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount

#### 8.1385.2.8 uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount

#### 8.1385.2.9 uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall

#### 8.1385.2.10 uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors

#### 8.1385.2.11 uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows

#### 8.1385.2.12 uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses

## 8.1386 unpack_wds_GetSessionDuration_t Struct Reference

### Data Fields

- uint64_t callDuration

### 8.1386.1 Detailed Description

This structure contains unpack get session duration information.

**Parameters**

| | |
|---|---|
| *callDuration* | • Duration of the current packet session in milliseconds |

### 8.1386.2 Field Documentation

#### 8.1386.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

## 8.1387 unpack_wds_GetSessionState_t Struct Reference

### Data Fields

- uint32_t connectionStatus

### 8.1387.1 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. |

### 8.1387.2 Field Documentation

#### 8.1387.2.1 uint32_t unpack_wds_GetSessionState_t::connectionStatus

## 8.1388 unpack_wds_RMSetTransferStatistics_t Struct Reference

### 8.1388.1 Detailed Description

This structure contains unpack fetch current data system transfer statistics information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.1389 unpack_wds_SetMobileIPProfile_t Struct Reference

### 8.1389.1 Detailed Description

This structure contains set mobile IP profile information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.1390 unpack_wds_SLQSCreateProfile_t Struct Reference

### Data Fields

- PackCreateProfileOut * pCreateProfileOut
- uint8_t * pProfileID
- uint16_t Tlvresult

### 8.1390.1 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 |

### 8.1390.2 Field Documentation

#### 8.1390.2.1 PackCreateProfileOut* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut

#### 8.1390.2.2 uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID

#### 8.1390.2.3 uint16_t unpack_wds_SLQSCreateProfile_t::Tlvresult

## 8.1391 unpack_wds_SLQSDeleteProfile_t Struct Reference

### Data Fields

- uint16_t extendedErrorCode

### 8.1391.1 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. |

### 8.1391.2 Field Documentation

#### 8.1391.2.1 uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode

## 8.1392 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

### 8.1392.1 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

### 8.1392.2 Field Documentation

#### 8.1392.2.1 wds_ChannelRateTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CRTlv

#### 8.1392.2.2 wds_ConnStatusTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CSTlv

#### 8.1392.2.3 wds_DataBearTechTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DBTTlv

#### 8.1392.2.4 wds_DormStatTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DSTlv

#### 8.1392.2.5 wds_LastMdmCallEndRsnTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::LMCERTlv

#### 8.1392.2.6 wds_RXBytesOKTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::RXBOTlv

#### 8.1392.2.7 wds_TXBytesOKTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::TXBOTlv

## 8.1393 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

### 8.1393.1 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. |

## 8.1393.2 Field Documentation

### 8.1393.2.1 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease

### 8.1393.2.2 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled

### 8.1393.2.3 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile

### 8.1393.2.4 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]

#### 8.1393.2.5 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen

#### 8.1393.2.6 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]

## 8.1394 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

### Data Fields

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

### 8.1394.1 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 |

### 8.1394.2 Field Documentation

#### 8.1394.2.1 currNetworkInfo unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]

#### 8.1394.2.2 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen

#### 8.1394.2.3 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork

## 8.1395 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

### 8.1395.1 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. |

### 8.1395.2 Field Documentation

#### 8.1395.2.1 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_rx_rate

#### 8.1395.2.2 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_tx_rate

#### 8.1395.2.3 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_rx_rate

#### 8.1395.2.4 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_tx_rate

## 8.1396 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference

### Data Fields

- uint8_t dataBearerMask
- qmiWDSDataBearerTechnology curDataBearerTechnology
- qmiWDSDataBearerTechnology lastCallDataBearerTechnology

### 8.1396.1 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 |

### 8.1396.2 Field Documentation

#### 8.1396.2.1 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::curDataBearerTechnology

#### 8.1396.2.2 uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask

#### 8.1396.2.3 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology

## 8.1397 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

### 8.1397.1 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 |

### 8.1397.2 Field Documentation

#### 8.1397.2.1 uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason

#### 8.1397.2.2 dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate

#### 8.1397.2.3 connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus

#### 8.1397.2.4 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech

#### 8.1397.2.5 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus

#### 8.1397.2.6 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech

#### 8.1397.2.7 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt

#### 8.1397.2.8 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt

#### 8.1397.2.9 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive

#### 8.1397.2.10 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount

#### 8.1397.2.11 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount

## 8.1398 unpack_wds_SLQSGetProfileSettings_t Struct Reference

### Data Fields

- UnPackGetProfileSettingOut * pProfileSettings
- uint8_t ProfileType
- uint16_t Tlvresult

### 8.1398.1 Detailed Description

This structure contains the profile setting information of parameter pOutput for API unpack_wds_SLQSGetProfile-Settings

**Parameters**

| | |
|---|---|
| *pProfileSettings* | - Profile Settings |
| *ProfileType* | - Profile Type<br>• 0 - Profile 3GPP<br>• 1 - Profile 3GPP2 |
| *Tlvresult* | - unpack Tlv Result. |

### 8.1398.2 Field Documentation

#### 8.1398.2.1 UnPackGetProfileSettingOut* unpack_wds_SLQSGetProfileSettings_t::pProfileSettings

#### 8.1398.2.2 uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType

#### 8.1398.2.3 uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult

## 8.1399 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

### 8.1399.1 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. |
| *ProfileID* | 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 |

### 8.1399.2 Field Documentation

#### 8.1399.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

#### 8.1399.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

#### 8.1399.2.3 struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList

8.1399.2.4 struct **wds_GPRSQoS** unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

8.1399.2.5 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4

8.1399.2.6 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag

8.1399.2.7 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference

8.1399.2.8 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4

8.1399.2.9 struct **wds_IPV6AddressInfo** unpack_wds_SLQSGetRuntimeSettings_t::IPV6AddrInfo

8.1399.2.10 struct **wds_IPV6GWAddressInfo** unpack_wds_SLQSGetRuntimeSettings_t::IPV6GWAddrInfo

8.1399.2.11 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu

8.1399.2.12 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO

8.1399.2.13 struct **wds_PCSCFFQDNAddressList** unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList

8.1399.2.14 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPType

8.1399.2.15 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4

8.1399.2.16 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]

8.1399.2.17 struct **wds_ProfileIdentifier** unpack_wds_SLQSGetRuntimeSettings_t::ProfileID

8.1399.2.18 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]

8.1399.2.19 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4

8.1399.2.20 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]

8.1399.2.21 struct **wds_PCSCFIPv4ServerAddressList** unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList

8.1399.2.22 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4

8.1399.2.23 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology

8.1399.2.24 **LibPackUMTSQoS** unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS

8.1399.2.25 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]

## 8.1400 unpack_wds_SLQSModifyProfile_t Struct Reference

### Data Fields

- uint16_t * pExtErrorCode

### 8.1400.1 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. |

### 8.1400.2 Field Documentation

#### 8.1400.2.1 uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode

## 8.1401 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.1401.1 Detailed Description

This structure contains unpack Set IP family preference information

**Parameters**

| | |
|---|---|
| *Tlvresult* | unpack result |

### 8.1401.2 Field Documentation

#### 8.1401.2.1 uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

## 8.1402 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

### 8.1402.1 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. |

### 8.1402.2 Field Documentation

#### 8.1402.2.1 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID

#### 8.1402.2.2 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status

#### 8.1402.2.3 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily

#### 8.1402.2.4 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd

#### 8.1402.2.5 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason

#### 8.1402.2.6 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName

#### 8.1402.2.7 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason

#### 8.1402.2.8 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType

## 8.1403 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

### 8.1403.1 Detailed Description

This structure contains unpack set WDS event callback information.

**Parameters**

| | |
|---|---|
| *xferStatAvail* | transfer statistic available |
| *tx_bytes* | transmit bytes |
| *rx_bytes* | received bytes |
| *tx_pkts* | transmit packets |
| *rx_pkts* | received packets |
| *mipstatAvail* | Mobile IP status available |
| *mipStatus* | Mobile IP status |
| *dBTechAvail* | Data Bearer technology available |
| *dBTechnology* | Data Bearer technology |
| *dormancyStat-Avail* | Dormancy status available |
| *dormancyStatus* | Dormancy status |
| *currDBTechAvail* | Current Data Bearer technology available |
| *ratMask* | RAT mask to indicate type of technology |
| *soMask* | SO mask to indicate the service type |
| *dataSysStatAvail* | Data System Status available |
| *prefNetwork* | preferred network |
| *currNWInfo* | Current Network Info |
| *dBtechExtAvail* | Data bearer technology extended available |
| *dBtechnologyExt* | Data bearer technology Extended. |
| *dBTechExtRat-Value* | Data bearer technology Extended RAT Value |
| *dBTechExtSo-Mask* | Data bearer technology Extended SO Mask |

### 8.1403.2 Field Documentation

#### 8.1403.2.1 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail

#### 8.1403.2.2 wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]

#### 8.1403.2.3 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail

#### 8.1403.2.4 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail

#### 8.1403.2.5 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechExtAvail

#### 8.1403.2.6 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtRatValue

#### 8.1403.2.7 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtSoMask

#### 8.1403.2.8 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology

#### 8.1403.2.9 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechnologyExt

#### 8.1403.2.10 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail

#### 8.1403.2.11 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus

#### 8.1403.2.12 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail

8.1403.2.13 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus

8.1403.2.14 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen

8.1403.2.15 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork

8.1403.2.16 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask

8.1403.2.17 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes

8.1403.2.18 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts

8.1403.2.19 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask

8.1403.2.20 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes

8.1403.2.21 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts

8.1403.2.22 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail

## 8.1404 unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

### Data Fields

- wdsDhcpv4HwConfig * pHwConfig
- wdsDhcpv4OptionList * pRequestOptionList

### 8.1404.1 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. |

### 8.1404.2 Field Documentation

8.1404.2.1 **wdsDhcpv4HwConfig*** unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig

8.1404.2.2 **wdsDhcpv4OptionList*** unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList

## 8.1405 unpack_wds_SLQSSGetLoopback_t Struct Reference

### Data Fields

- uint8_t ByteLoopbackMode
- uint8_t ByteLoopbackMultiplier

#### 8.1405.1 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. |

#### 8.1405.2 Field Documentation

##### 8.1405.2.1 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode

##### 8.1405.2.2 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier

## 8.1406 unpack_wds_SLQSStartDataSession_t Struct Reference

### Data Fields

- uint32_t * psid
- uint32_t * pFailureReason
- uint32_t * pVerboseFailReasonType
- uint32_t * pVerboseFailureReason

#### 8.1406.1 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. |

### 8.1406.2 Field Documentation

#### 8.1406.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

#### 8.1406.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

#### 8.1406.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

#### 8.1406.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

## 8.1407 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]

### 8.1407.1 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. |

### 8.1407.2 Field Documentation

#### 8.1407.2.1 int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]

#### 8.1407.2.2 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId

#### 8.1407.2.3 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

#### 8.1407.2.4 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address

#### 8.1407.2.5 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress

#### 8.1407.2.6 ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address

#### 8.1407.2.7 ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress

#### 8.1407.2.8 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address

#### 8.1407.2.9 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]

#### 8.1407.2.10 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address

#### 8.1407.2.11 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]

#### 8.1407.2.12 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address

#### 8.1407.2.13 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]

#### 8.1407.2.14 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address

#### 8.1407.2.15 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]

## 8.1408 UnPackGetProfileSettingOut Struct Reference

### Data Fields

- UnpackQmiProfileInfo curProfile
- uint16_t * pExtErrCode

### 8.1408.1 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. |

### 8.1408.2 Field Documentation

#### 8.1408.2.1 UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile

#### 8.1408.2.2 uint16_t* UnPackGetProfileSettingOut::pExtErrCode

## 8.1409 unpackWdsProfileParam Union Reference

### Data Fields

- LibpackProfile3GPP SlqsProfile3GPP
- LibpackProfile3GPP2 SlqsProfile3GPP2

### 8.1409.1 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 |
| *SlqsProfile3GPP2* | 3GPP2 profile<br>• See LibpackProfile3GPP2 |

### 8.1409.2 Field Documentation

#### 8.1409.2.1 LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP

#### 8.1409.2.2 LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2

## 8.1410 urlAddr Struct Reference

### Data Fields

- CHAR address [256]
- BYTE TlvPresent

### 8.1410.1 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 |

### 8.1410.2 Field Documentation

#### 8.1410.2.1 CHAR urlAddr::address[256]

#### 8.1410.2.2 BYTE urlAddr::TlvPresent

## 8.1411 URLAddrInfo Struct Reference

### Data Fields

- CHAR urlAddr [256]

### 8.1411.1 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 |

### 8.1411.2 Field Documentation

#### 8.1411.2.1 CHAR URLAddrInfo::urlAddr[256]

## 8.1412 USBCompConfig Struct Reference

### Data Fields

- BYTE * pUSBComp

### 8.1412.1 Detailed Description

This structure is used to store USB composition information

**Parameters**

| | |
|---|---|
| *pUSBComp[IN]* | • 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 |

### 8.1412.2 Field Documentation

#### 8.1412.2.1 BYTE* USBCompConfig::pUSBComp

## 8.1413 USBCompParams Struct Reference

### Data Fields

- BYTE * pUSBComp
- BYTE * pNumSupUSBComps
- BYTE * pSupUSBComps

### 8.1413.1 Detailed Description

This structure is used to store retrieved USB Composition

**Parameters**

| | |
|---|---|
| *pUSBComp[OUT]* | • 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<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 |
| *pNumSupUSBComps[OUT]* | • Number of supported USB compositions in the parameter to follow<br>• Range - 0-255 |

| | |
|---|---|
| *pSupUSB-Comps[OUT]* | • 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 |

### 8.1413.2 Field Documentation

#### 8.1413.2.1 BYTE* USBCompParams::pNumSupUSBComps

#### 8.1413.2.2 BYTE* USBCompParams::pSupUSBComps

#### 8.1413.2.3 BYTE* USBCompParams::pUSBComp

## 8.1414 usbMTUSizeTlv Struct Reference

### Data Fields

- BYTE TlvPresent
- WORD UsbMTUSize

### 8.1414.1 Detailed Description

This structure contains the parameters for USB MTU Size

**Parameters**

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

| | |
|---|---|
| *UsbMTUSize* | USB MTU size<br>• Size of USB MTU |

### 8.1414.2 Field Documentation

#### 8.1414.2.1 BYTE usbMTUSizeTlv::TlvPresent

#### 8.1414.2.2 WORD usbMTUSizeTlv::UsbMTUSize

## 8.1415 USSDNoWaitIndicationInfo Struct Reference

### Data Fields

- BYTE * pError
- BYTE * pFailureCause
- struct USSInfo * pUSSDData
- alphaIDInfo * pAlphaIdentifier

### 8.1415.1 Detailed Description

Contains the parameters passed for USSDNoWaitIndicationCallback by the device.

**Parameters**

| | |
|---|---|
| *pError* | • Type of Error (if any) |
| *pFailureCause* | • Supplementary services failure cause |
| *pUSSDData* | • USS Data from Network.<br>• See USSInfo for more details. |

### 8.1415.2 Field Documentation

#### 8.1415.2.1 alphaIDInfo* USSDNoWaitIndicationInfo::pAlphaIdentifier

#### 8.1415.2.2 BYTE* USSDNoWaitIndicationInfo::pError

#### 8.1415.2.3 BYTE* USSDNoWaitIndicationInfo::pFailureCause

#### 8.1415.2.4 struct USSInfo* USSDNoWaitIndicationInfo::pUSSDData

## 8.1416 USSDRespFNetwork Struct Reference

### Data Fields

- char * pTypeCode

- char * pRespData

### 8.1416.1 Detailed Description

This structure contains the response from the network

**Parameters**

| | |
|---|---|
| *pTypeCode* | "0" USSD-Notify – text in pRespData "1" USSD-Request – text in pRespData "2" Session terminated by network "3" other local client (eg, SIM Toolkit) has responded "4" Operation not supported "5" Network timeout |
| *pRespData* | • points to a message string received from the network |

### 8.1416.2 Field Documentation

#### 8.1416.2.1 char* USSDRespFNetwork::pRespData

#### 8.1416.2.2 char* USSDRespFNetwork::pTypeCode

## 8.1417 USSInfo Struct Reference

### Data Fields

- BYTE ussDCS
- BYTE ussLen
- BYTE ussData [182]

### 8.1417.1 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 |

### 8.1417.2 Field Documentation

#### 8.1417.2.1 BYTE USSInfo::ussData[182]

#### 8.1417.2.2 BYTE USSInfo::ussDCS

#### 8.1417.2.3 BYTE USSInfo::ussLen

## 8.1418 USSResp Struct Reference

### Data Fields

- WORD * pfailureCause
- alphaIDInfo * pAlphaIDInfo
- struct USSInfo * pUSSDInfo
- BYTE * pCcResultType
- BYTE * pCallId
- ccSUPSType * pCCSuppsType

### 8.1418.1 Field Documentation

#### 8.1418.1.1 alphaIDInfo* USSResp::pAlphaIDInfo

#### 8.1418.1.2 BYTE* USSResp::pCallId

#### 8.1418.1.3 BYTE* USSResp::pCcResultType

#### 8.1418.1.4 ccSUPSType* USSResp::pCCSuppsType

#### 8.1418.1.5 WORD* USSResp::pfailureCause

#### 8.1418.1.6 struct USSInfo* USSResp::pUSSDInfo

## 8.1419 UUSInfo Struct Reference

### Data Fields

- BYTE UUSType
- BYTE UUSDcs
- BYTE UUSDatalen
- BYTE UUSData [255]

### 8.1419.1 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_-DESCRIPTION-_LENGTH]* | • UUS data encoded as per coding scheme |

### 8.1419.2 Field Documentation

#### 8.1419.2.1 BYTE UUSInfo::UUSData[255]

#### 8.1419.2.2 BYTE UUSInfo::UUSDatalen

#### 8.1419.2.3 BYTE UUSInfo::UUSDcs

#### 8.1419.2.4 BYTE UUSInfo::UUSType

## 8.1420 verifyUIMPIN Struct Reference

### Data Fields

- BYTE pinID
- BYTE pinLen
- BYTE pinVal [255]

### 8.1420.1 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. |

### 8.1420.2 Field Documentation

#### 8.1420.2.1 BYTE verifyUIMPIN::pinID

#### 8.1420.2.2 BYTE verifyUIMPIN::pinLen

#### 8.1420.2.3 BYTE verifyUIMPIN::pinVal[255]

## 8.1421 voice_airTimer Struct Reference

### Data Fields

- uint8_t namID
- uint32_t airTimerValue

### 8.1421.1 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. |

### 8.1421.2 Field Documentation

#### 8.1421.2.1 uint32_t voice_airTimer::airTimerValue

#### 8.1421.2.2 uint8_t voice_airTimer::namID

## 8.1422 voice_allCallsAlphaIDInfo Struct Reference

### Data Fields

- uint8_t callID
- voice_alphaIDInfo AlphaIDInfo

#### 8.1422.1 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. |

#### 8.1422.2 Field Documentation

##### 8.1422.2.1 voice_alphaIDInfo voice_allCallsAlphaIDInfo::AlphaIDInfo

##### 8.1422.2.2 uint8_t voice_allCallsAlphaIDInfo::callID

## 8.1423 voice_allCallsDiagInfo Struct Reference

### Data Fields

- uint8_t callID
- voice_diagInfo DiagInfo

#### 8.1423.1 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. |

#### 8.1423.2 Field Documentation

##### 8.1423.2.1 uint8_t voice_allCallsDiagInfo::callID

##### 8.1423.2.2 voice_diagInfo voice_allCallsDiagInfo::DiagInfo

## 8.1424 voice_allCallsUUSInfo Struct Reference

### Data Fields

- uint8_t callID
- voice_UUSInfo uusInfo

#### 8.1424.1 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. |

#### 8.1424.2 Field Documentation

##### 8.1424.2.1 uint8_t voice_allCallsUUSInfo::callID

##### 8.1424.2.2 voice_UUSInfo voice_allCallsUUSInfo::uusInfo

## 8.1425 voice_alphaIDInfo Struct Reference

### Data Fields

- uint8_t alphaDcs
- uint8_t alphaLen
- uint8_t alphaText [255]

#### 8.1425.1 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_DESCRIPTION_LENGTH]* | • Data encoded as per the alpha_dcs |

### 8.1425.2 Field Documentation

#### 8.1425.2.1 uint8_t voice_alphaIDInfo::alphaDcs

#### 8.1425.2.2 uint8_t voice_alphaIDInfo::alphaLen

#### 8.1425.2.3 uint8_t voice_alphaIDInfo::alphaText[255]

## 8.1426 voice_arrAlertingPattern Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint32_t alertingPattern [20]

### 8.1426.1 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_NO_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 |

### 8.1426.2 Field Documentation

#### 8.1426.2.1 uint32_t voice_arrAlertingPattern::alertingPattern[20]

#### 8.1426.2.2 uint8_t voice_arrAlertingPattern::callID[20]

#### 8.1426.2.3 uint8_t voice_arrAlertingPattern::numInstances

## 8.1427 voice_arrAlertingType Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint8_t AlertingType [20]

### 8.1427.1 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 |

### 8.1427.2 Field Documentation

#### 8.1427.2.1 uint8_t voice_arrAlertingType::AlertingType[20]

#### 8.1427.2.2 uint8_t voice_arrAlertingType::callID[20]

#### 8.1427.2.3 uint8_t voice_arrAlertingType::numInstances

## 8.1428 voice_arrAlphaID Struct Reference

### Data Fields

- uint8_t numInstances
- voice_allCallsAlphaIDInfo allCallsAlphaIDInfoArr [20]

### 8.1428.1 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. |

### 8.1428.2 Field Documentation

#### 8.1428.2.1 voice_allCallsAlphaIDInfo voice_arrAlphaID::allCallsAlphaIDInfoArr[20]

#### 8.1428.2.2 uint8_t voice_arrAlphaID::numInstances

## 8.1429 voice_arrCalledPartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_peerNumberInfo CalledPartyNum [20]

### 8.1429.1 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. |

### 8.1429.2 Field Documentation

#### 8.1429.2.1 voice_peerNumberInfo voice_arrCalledPartyNum::CalledPartyNum[20]

#### 8.1429.2.2 uint8_t voice_arrCalledPartyNum::numInstances

## 8.1430 voice_arrCallEndReason Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t callEndReason [20]

### 8.1430.1 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_N-O_OF_CALLS]* | • Array of Call End Reason .<br>• See Table9 qaGobiApiTableVoiceCallEndReasons.h for a list of valid voice-related call end reasons |

### 8.1430.2 Field Documentation

#### 8.1430.2.1 uint16_t voice_arrCallEndReason::callEndReason[20]

#### 8.1430.2.2 uint8_t voice_arrCallEndReason::callID[20]

#### 8.1430.2.3 uint8_t voice_arrCallEndReason::numInstances

## 8.1431 voice_arrCallInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_getAllCallInformation getAllCallInfo [20]

### 8.1431.1 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[V-OICE_MAX_NO-_OF_CALLS]* | • Array of CallInfo.<br>• See voice_getAllCallInformation for more information. |

### 8.1431.2 Field Documentation

#### 8.1431.2.1 voice_getAllCallInformation voice_arrCallInfo::getAllCallInfo[20]

#### 8.1431.2.2 uint8_t voice_arrCallInfo::numInstances

## 8.1432 voice_arrConnectPartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_peerNumberInfo ConnectedPartyNum [20]

### 8.1432.1 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. |

### 8.1432.2 Field Documentation

#### 8.1432.2.1 voice_peerNumberInfo voice_arrConnectPartyNum::ConnectedPartyNum[20]

#### 8.1432.2.2 uint8_t voice_arrConnectPartyNum::numInstances

## 8.1433 voice_arrDiagInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_allCallsDiagInfo DiagInfo [20]

### 8.1433.1 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. |

### 8.1433.2 Field Documentation

#### 8.1433.2.1 voice_allCallsDiagInfo voice_arrDiagInfo::DiagInfo[20]

#### 8.1433.2.2 uint8_t voice_arrDiagInfo::numInstances

## 8.1434 voice_arrRedirPartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_peerNumberInfo RedirPartyNum [20]

### 8.1434.1 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_NO_OF_CALLS]* | • Array of RedirPartyNum.<br>• See voice_peerNumberInfo for more information. |

### 8.1434.2 Field Documentation

#### 8.1434.2.1 uint8_t voice_arrRedirPartyNum::numInstances

#### 8.1434.2.2 voice_peerNumberInfo voice_arrRedirPartyNum::RedirPartyNum[20]

## 8.1435 voice_arrRemotePartyName Struct Reference

### Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyName GetAllCallRmtPtyName [20]

### 8.1435.1 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. |
| *GetAllCallRmtPtyName[MAX_NO_OF_CALLS]* | • Array of remotePartyName.<br>• See voice_getAllCallRmtPtyName for more information. |

### 8.1435.2 Field Documentation

#### 8.1435.2.1 voice_getAllCallRmtPtyName voice_arrRemotePartyName::GetAllCallRmtPtyName[20]

#### 8.1435.2.2 uint8_t voice_arrRemotePartyName::numInstances

## 8.1436 voice_arrRemotePartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyNum RmtPtyNum [20]

### 8.1436.1 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. |

### 8.1436.2 Field Documentation

#### 8.1436.2.1 uint8_t voice_arrRemotePartyNum::numInstances

#### 8.1436.2.2 voice_getAllCallRmtPtyNum voice_arrRemotePartyNum::RmtPtyNum[20]

## 8.1437 voice_arrSvcOption Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t srvOption [20]

### 8.1437.1 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. |

### 8.1437.2 Field Documentation

#### 8.1437.2.1 uint8_t voice_arrSvcOption::callID[20]

#### 8.1437.2.2 uint8_t voice_arrSvcOption::numInstances

#### 8.1437.2.3 uint16_t voice_arrSvcOption::srvOption[20]

## 8.1438 voice_arrUUSInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_allCallsUUSInfo AllCallsUUSInfo [20]

### 8.1438.1 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. |

### 8.1438.2 Field Documentation

#### 8.1438.2.1 voice_allCallsUUSInfo voice_arrUUSInfo::AllCallsUUSInfo[20]

#### 8.1438.2.2 uint8_t voice_arrUUSInfo::numInstances

## 8.1439 voice_burstDTMFInfo Struct Reference

### Data Fields

- uint8_t * pCallID
- uint8_t digitCnt
- uint8_t pDigitBuff [255]

### 8.1439.1 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 |

### 8.1439.2 Field Documentation

#### 8.1439.2.1 uint8_t voice_burstDTMFInfo::digitCnt

#### 8.1439.2.2 uint8_t* voice_burstDTMFInfo::pCallID

#### 8.1439.2.3 uint8_t voice_burstDTMFInfo::pDigitBuff[255]

## 8.1440 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]

### 8.1440.1 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. |

### 8.1440.2 Field Documentation

#### 8.1440.2.1 uint8_t voice_calledPartyInfo::number[255]

#### 8.1440.2.2 uint8_t voice_calledPartyInfo::numLen

#### 8.1440.2.3 uint8_t voice_calledPartyInfo::numPlan

#### 8.1440.2.4 uint8_t voice_calledPartyInfo::numType

#### 8.1440.2.5 uint8_t voice_calledPartyInfo::PI

#### 8.1440.2.6 uint8_t voice_calledPartyInfo::SI

## 8.1441 voice_calledPartySubAdd Struct Reference

### Data Fields

- uint8_t extBit
- uint8_t subAddrType
- uint8_t oddEvenInd
- uint8_t subAddrLen
- uint8_t subAddr [255]

### 8.1441.1 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. |

### 8.1441.2 Field Documentation

#### 8.1441.2.1 uint8_t voice_calledPartySubAdd::extBit

#### 8.1441.2.2 uint8_t voice_calledPartySubAdd::oddEvenInd

#### 8.1441.2.3 uint8_t voice_calledPartySubAdd::subAddr[255]

#### 8.1441.2.4 uint8_t voice_calledPartySubAdd::subAddrLen

#### 8.1441.2.5 uint8_t voice_calledPartySubAdd::subAddrType

## 8.1442 voice_callerIDInfo Struct Reference

### Data Fields

- uint8_t PI
- uint8_t callerIDLen
- uint8_t callerID [255]

### 8.1442.1 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. |

### 8.1442.2 Field Documentation

#### 8.1442.2.1 uint8_t voice_callerIDInfo::callerID[255]

#### 8.1442.2.2 uint8_t voice_callerIDInfo::callerIDLen

#### 8.1442.2.3 uint8_t voice_callerIDInfo::PI

## 8.1443 voice_callFwdTypeAndPlan Struct Reference

### Data Fields

- uint8_t numberType
- uint8_t numberPlan

### 8.1443.1 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 |

### 8.1443.2 Field Documentation

#### 8.1443.2.1 uint8_t voice_callFwdTypeAndPlan::numberPlan

#### 8.1443.2.2 uint8_t voice_callFwdTypeAndPlan::numberType

## 8.1444 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]

### 8.1444.1 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. |

### 8.1444.2 Field Documentation

#### 8.1444.2.1 uint8_t voice_callFWExtInfo::noReplyTimer

#### 8.1444.2.2 uint8_t voice_callFWExtInfo::number[255]

#### 8.1444.2.3 uint8_t voice_callFWExtInfo::numLen

#### 8.1444.2.4 uint8_t voice_callFWExtInfo::numPlan

#### 8.1444.2.5 uint8_t voice_callFWExtInfo::numType

#### 8.1444.2.6 uint8_t voice_callFWExtInfo::PI

#### 8.1444.2.7 uint8_t voice_callFWExtInfo::SI

#### 8.1444.2.8 uint8_t voice_callFWExtInfo::SvcClass

#### 8.1444.2.9 uint8_t voice_callFWExtInfo::SvcStatus

## 8.1445 voice_callFWInfo Struct Reference

### Data Fields

- uint8_t SvcStatus
- uint8_t SvcClass
- uint8_t numLen
- uint8_t number [255]
- uint8_t noReplyTimer

### 8.1445.1 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. |

### 8.1445.2 Field Documentation

#### 8.1445.2.1 uint8_t voice_callFWInfo::noReplyTimer

#### 8.1445.2.2 uint8_t voice_callFWInfo::number[255]

#### 8.1445.2.3 uint8_t voice_callFWInfo::numLen

#### 8.1445.2.4 uint8_t voice_callFWInfo::SvcClass

#### 8.1445.2.5 uint8_t voice_callFWInfo::SvcStatus

## 8.1446 voice_callInfo Struct Reference

### Data Fields

- uint8_t callID
- uint8_t callState
- uint8_t callType
- uint8_t direction
- uint8_t mode

### 8.1446.1 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 |

## 8.1446.2 Field Documentation

### 8.1446.2.1 uint8_t voice_callInfo::callID

#### 8.1446.2.2 uint8_t voice_callInfo::callState

#### 8.1446.2.3 uint8_t voice_callInfo::callType

#### 8.1446.2.4 uint8_t voice_callInfo::direction

#### 8.1446.2.5 uint8_t voice_callInfo::mode

## 8.1447 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]

### 8.1447.1 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. |

### 8.1447.2 Field Documentation

#### 8.1447.2.1 uint8_t voice_callingPartyInfo::number[255]

#### 8.1447.2.2 uint8_t voice_callingPartyInfo::numLen

#### 8.1447.2.3 uint8_t voice_callingPartyInfo::numPlan

#### 8.1447.2.4 uint8_t voice_callingPartyInfo::numType

#### 8.1447.2.5 uint8_t voice_callingPartyInfo::PI

#### 8.1447.2.6 uint8_t voice_callingPartyInfo::SI

## 8.1448 voice_ccSUPSType Struct Reference

### Data Fields

- uint8_t svcType

- uint8_t reason

### 8.1448.1 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. |

### 8.1448.2 Field Documentation

#### 8.1448.2.1 uint8_t voice_ccSUPSType::reason

#### 8.1448.2.2 uint8_t voice_ccSUPSType::svcType

## 8.1449 voice_CLIPResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.1449.1 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 |

### 8.1449.2 Field Documentation

#### 8.1449.2.1 uint8_t voice_CLIPResp::ActiveStatus

#### 8.1449.2.2 uint8_t voice_CLIPResp::ProvisionStatus

## 8.1450 voice_CLIRResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.1450.1 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 |

### 8.1450.2 Field Documentation

#### 8.1450.2.1 uint8_t voice_CLIRResp::ActiveStatus

#### 8.1450.2.2 uint8_t voice_CLIRResp::ProvisionStatus

## 8.1451 voice_CNAPResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.1451.1 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 |

### 8.1451.2 Field Documentation

#### 8.1451.2.1 uint8_t voice_CNAPResp::ActiveStatus

#### 8.1451.2.2 uint8_t voice_CNAPResp::ProvisionStatus

## 8.1452 voice_COLPResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.1452.1 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 |

### 8.1452.2 Field Documentation

#### 8.1452.2.1 uint8_t voice_COLPResp::ActiveStatus

#### 8.1452.2.2 uint8_t voice_COLPResp::ProvisionStatus

## 8.1453 voice_COLRResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.1453.1 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 |

### 8.1453.2 Field Documentation

#### 8.1453.2.1 uint8_t voice_COLRResp::ActiveStatus

#### 8.1453.2.2 uint8_t voice_COLRResp::ProvisionStatus

## 8.1454 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]

### 8.1454.1 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. |

### 8.1454.2 Field Documentation

#### 8.1454.2.1 uint8_t voice_connectNumInfo::callerID[81]

#### 8.1454.2.2 uint8_t voice_connectNumInfo::callerIDLen

#### 8.1454.2.3 uint8_t voice_connectNumInfo::numPlan

#### 8.1454.2.4 uint8_t voice_connectNumInfo::numPresInd

#### 8.1454.2.5 uint8_t voice_connectNumInfo::numType

#### 8.1454.2.6 uint8_t voice_connectNumInfo::screeningInd

## 8.1455 voice_CUGInfo Struct Reference

### Data Fields

- uint16_t CUGIndex
- uint8_t SuppPrefCUG
- uint8_t SuppOA

### 8.1455.1 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 |

### 8.1455.2 Field Documentation

#### 8.1455.2.1 uint16_t voice_CUGInfo::CUGIndex

#### 8.1455.2.2 uint8_t voice_CUGInfo::SuppOA

#### 8.1455.2.3 uint8_t voice_CUGInfo::SuppPrefCUG

## 8.1456 voice_curAMRConfig Struct Reference

### Data Fields

- uint8_t gsmAmrStat
- uint8_t wcdmaAmrStat

### 8.1456.1 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 |

### 8.1456.2 Field Documentation

#### 8.1456.2.1 uint8_t voice_curAMRConfig::gsmAmrStat

#### 8.1456.2.2 uint8_t voice_curAMRConfig::wcdmaAmrStat

## 8.1457 voice_diagInfo Struct Reference

### Data Fields

- uint8_t diagInfoLen
- uint8_t diagnosticInfo [255]

### 8.1457.1 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. |

### 8.1457.2 Field Documentation

#### 8.1457.2.1 uint8_t voice_diagInfo::diagInfoLen

#### 8.1457.2.2 uint8_t voice_diagInfo::diagnosticInfo[255]

## 8.1458 voice_DTMFInfo Struct Reference

### Data Fields

- uint8_t callID
- uint8_t DTMFEvent

- uint8_t digitCnt
- uint8_t digitBuff [255]

### 8.1458.1 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 |

### 8.1458.2 Field Documentation

#### 8.1458.2.1 uint8_t voice_DTMFInfo::callID

#### 8.1458.2.2 uint8_t voice_DTMFInfo::digitBuff[255]

#### 8.1458.2.3 uint8_t voice_DTMFInfo::digitCnt

#### 8.1458.2.4 uint8_t voice_DTMFInfo::DTMFEvent

## 8.1459 voice_DTMFLengths Struct Reference

### Data Fields

- uint8_t DTMFPulseWidth
- uint8_t DTMFInterdigitInterval

### 8.1459.1 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 |

### 8.1459.2 Field Documentation

#### 8.1459.2.1 uint8_t voice_DTMFLengths::DTMFInterdigitInterval

#### 8.1459.2.2 uint8_t voice_DTMFLengths::DTMFPulseWidth

## 8.1460 voice_ECTNum Struct Reference

### Data Fields

- uint8_t ECTCallState
- uint8_t presentationInd
- uint8_t number [81]

### 8.1460.1 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 |

### 8.1460.2 Field Documentation

#### 8.1460.2.1 uint8_t voice_ECTNum::ECTCallState

#### 8.1460.2.2 uint8_t voice_ECTNum::number[81]

#### 8.1460.2.3 uint8_t voice_ECTNum::presentationInd

## 8.1461 voice_extDispRecInfo Struct Reference

### Data Fields

- uint8_t dispType
- uint8_t extDispInfoLen
- uint8_t extDispInfo [255]

### 8.1461.1 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. |

### 8.1461.2 Field Documentation

#### 8.1461.2.1 uint8_t voice_extDispRecInfo::dispType

#### 8.1461.2.2 uint8_t voice_extDispRecInfo::extDispInfo[255]

#### 8.1461.2.3 uint8_t voice_extDispRecInfo::extDispInfoLen

## 8.1462 voice_getAllCallInformation Struct Reference

### Data Fields

- voice_callInfo CallInfo
- uint8_t isEmpty

- uint8_t ALS

### 8.1462.1 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 |

### 8.1462.2 Field Documentation

#### 8.1462.2.1 uint8_t voice_getAllCallInformation::ALS

#### 8.1462.2.2 voice_callInfo voice_getAllCallInformation::CallInfo

#### 8.1462.2.3 uint8_t voice_getAllCallInformation::isEmpty

## 8.1463 voice_getAllCallRmtPtyName Struct Reference

### Data Fields

- uint8_t callID
- voice_remotePartyName RemotePartyName

### 8.1463.1 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. |

### 8.1463.2 Field Documentation

#### 8.1463.2.1 uint8_t voice_getAllCallRmtPtyName::callID

#### 8.1463.2.2 voice_remotePartyName voice_getAllCallRmtPtyName::RemotePartyName

## 8.1464 voice_getAllCallRmtPtyNum Struct Reference

### Data Fields

- uint8_t callID
- voice_remotePartyNum RemotePartyNum

### 8.1464.1 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. |

### 8.1464.2 Field Documentation

#### 8.1464.2.1 uint8_t voice_getAllCallRmtPtyNum::callID

#### 8.1464.2.2 voice_remotePartyNum voice_getAllCallRmtPtyNum::RemotePartyNum

## 8.1465 voice_getCallFWExtInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_callFWExtInfo CallFWExtInfo [20]

### 8.1465.1 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_N-O_OF_CALLS]* | • Array of CallFWExtInfo.<br>– See voice_callFWExtInfo for more information. |

### 8.1465.2 Field Documentation

#### 8.1465.2.1 voice_callFWExtInfo voice_getCallFWExtInfo::CallFWExtInfo[20]

#### 8.1465.2.2 uint8_t voice_getCallFWExtInfo::numInstances

## 8.1466 voice_getCallFWInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_callFWInfo CallFWInfo [20]

### 8.1466.1 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. |

### 8.1466.2 Field Documentation

#### 8.1466.2.1 voice_callFWInfo voice_getCallFWInfo::CallFWInfo[20]

#### 8.1466.2.2 uint8_t voice_getCallFWInfo::numInstances

## 8.1467 voice_lineCtrlInfo Struct Reference

### Data Fields

- uint8_t polarityIncluded
- uint8_t toggleMode
- uint8_t revPolarity
- uint8_t pwrDenialTime

### 8.1467.1 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 |

### 8.1467.2 Field Documentation

#### 8.1467.2.1 uint8_t voice_lineCtrlInfo::polarityIncluded

#### 8.1467.2.2 uint8_t voice_lineCtrlInfo::pwrDenialTime

#### 8.1467.2.3 uint8_t voice_lineCtrlInfo::revPolarity

#### 8.1467.2.4 uint8_t voice_lineCtrlInfo::toggleMode

## 8.1468 voice_newPwdData Struct Reference

### Data Fields

- uint8_t newPwd [4]
- uint8_t newPwdAgain [4]

### 8.1468.1 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. |

### 8.1468.2 Field Documentation

#### 8.1468.2.1 uint8_t voice_newPwdData::newPwd[4]

#### 8.1468.2.2 uint8_t voice_newPwdData::newPwdAgain[4]

## 8.1469 voice_NSSAudioCtrl Struct Reference

### Data Fields

- uint8_t upLink
- uint8_t downLink

### 8.1469.1 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]. |

### 8.1469.2 Field Documentation

#### 8.1469.2.1 uint8_t voice_NSSAudioCtrl::downLink

#### 8.1469.2.2 uint8_t voice_NSSAudioCtrl::upLink

## 8.1470 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]

### 8.1470.1 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. |

### 8.1470.2 Field Documentation

#### 8.1470.2.1 uint8_t voice_peerNumberInfo::callID

#### 8.1470.2.2 uint8_t voice_peerNumberInfo::number[81]

#### 8.1470.2.3 uint8_t voice_peerNumberInfo::numLen

#### 8.1470.2.4 uint8_t voice_peerNumberInfo::numPI

#### 8.1470.2.5 uint8_t voice_peerNumberInfo::numPlan

#### 8.1470.2.6 uint8_t voice_peerNumberInfo::numSI

#### 8.1470.2.7 uint8_t voice_peerNumberInfo::numType

## 8.1471 voice_prefVoiceSO Struct Reference

### Data Fields

- uint8_t namID
- uint8_t evrcCapability
- uint16_t homePageVoiceSO
- uint16_t homeOrigVoiceSO
- uint16_t roamOrigVoiceSO

### 8.1471.1 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 |

### 8.1471.2 Field Documentation

#### 8.1471.2.1 uint8_t voice_prefVoiceSO::evrcCapability

#### 8.1471.2.2 uint16_t voice_prefVoiceSO::homeOrigVoiceSO

#### 8.1471.2.3 uint16_t voice_prefVoiceSO::homePageVoiceSO

#### 8.1471.2.4 uint8_t voice_prefVoiceSO::namID

#### 8.1471.2.5 uint16_t voice_prefVoiceSO::roamOrigVoiceSO

## 8.1472 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]

### 8.1472.1 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. |

### 8.1472.2 Field Documentation

#### 8.1472.2.1 uint8_t voice_redirNumInfo::number[255]

#### 8.1472.2.2 uint8_t voice_redirNumInfo::numLen

#### 8.1472.2.3 uint8_t voice_redirNumInfo::numPlan

#### 8.1472.2.4 uint8_t voice_redirNumInfo::numType

#### 8.1472.2.5 uint8_t voice_redirNumInfo::PI

#### 8.1472.2.6 uint8_t voice_redirNumInfo::reason

#### 8.1472.2.7 uint8_t voice_redirNumInfo::SI

## 8.1473 voice_remotePartyName Struct Reference

### Data Fields

- uint8_t namePI
- uint8_t codingScheme
- uint8_t nameLen
- uint8_t callerName [255]

### 8.1473.1 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. |

### 8.1473.2 Field Documentation

#### 8.1473.2.1 uint8_t voice_remotePartyName::callerName[255]

#### 8.1473.2.2 uint8_t voice_remotePartyName::codingScheme

#### 8.1473.2.3 uint8_t voice_remotePartyName::nameLen

#### 8.1473.2.4 uint8_t voice_remotePartyName::namePI

## 8.1474 voice_remotePartyNum Struct Reference

### Data Fields

- uint8_t presentationInd
- uint8_t numLen
- uint8_t remPartyNumber [81]

### 8.1474.1 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. |
| *remParty-Number[MAX_-CALL_NO_LEN]* | • Array of numbers in ASCII, NULL ending. |

### 8.1474.2 Field Documentation

#### 8.1474.2.1 uint8_t voice_remotePartyNum::numLen

#### 8.1474.2.2 uint8_t voice_remotePartyNum::presentationInd

#### 8.1474.2.3 uint8_t voice_remotePartyNum::remPartyNumber[81]

## 8.1475 voice_roamTimer Struct Reference

### Data Fields

- uint8_t namID
- uint32_t roamTimerValue

### 8.1475.1 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. |

### 8.1475.2 Field Documentation

#### 8.1475.2.1 uint8_t voice_roamTimer::namID

#### 8.1475.2.2 uint32_t voice_roamTimer::roamTimerValue

## 8.1476 voice_signalInfo Struct Reference

### Data Fields

- uint8_t signalType
- uint8_t alertPitch
- uint8_t signal

### 8.1476.1 Detailed Description

This structure contains Signal Information

**Parameters**

| | |
|---|---|
| *signalType* | • Call identifier for the call. |
| *alertPitch* | • Signal Information |
| *signal* | • Caller ID Information |

### 8.1476.2 Field Documentation

#### 8.1476.2.1 uint8_t voice_signalInfo::alertPitch

#### 8.1476.2.2 uint8_t voice_signalInfo::signal

#### 8.1476.2.3 uint8_t voice_signalInfo::signalType

## 8.1477 voice_SUPSInfo Struct Reference

### Data Fields

- uint8_t svcType
- uint8_t isModByCC

### 8.1477.1 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 |

### 8.1477.2 Field Documentation

#### 8.1477.2.1 uint8_t voice_SUPSInfo::isModByCC

#### 8.1477.2.2 uint8_t voice_SUPSInfo::svcType

## 8.1478 voice_USSDNotificationNetworkInfo Struct Reference

### Data Fields

- uint8_t tlvPresent
- struct voice_USSInfo networkInfo

### 8.1478.1 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. |

### 8.1478.2 Field Documentation

#### 8.1478.2.1 struct voice_USSInfo voice_USSDNotificationNetworkInfo::networkInfo

#### 8.1478.2.2 uint8_t voice_USSDNotificationNetworkInfo::tlvPresent

## 8.1479 voice_USSInfo Struct Reference

### Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

### 8.1479.1 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 |

### 8.1479.2 Field Documentation

#### 8.1479.2.1 uint8_t voice_USSInfo::ussData[182]

#### 8.1479.2.2 uint8_t voice_USSInfo::ussDCS

#### 8.1479.2.3 uint8_t voice_USSInfo::ussLen

## 8.1480 voice_UUSInfo Struct Reference

### Data Fields

- uint8_t UUSType
- uint8_t UUSDcs

- uint8_t UUSDatalen
- uint8_t UUSData [255]

### 8.1480.1 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 |

### 8.1480.2 Field Documentation

#### 8.1480.2.1 uint8_t voice_UUSInfo::UUSData[255]

#### 8.1480.2.2 uint8_t voice_UUSInfo::UUSDatalen

#### 8.1480.2.3 uint8_t voice_UUSInfo::UUSDcs

#### 8.1480.2.4 uint8_t voice_UUSInfo::UUSType

## 8.1481 voiceALSSelectLineInfo Struct Reference

### Data Fields

- BYTE lineValue

#### 8.1481.1 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 |

#### 8.1481.2 Field Documentation

##### 8.1481.2.1 BYTE voiceALSSelectLineInfo::lineValue

## 8.1482 voiceALSSetLineSwitchInfo Struct Reference

### Data Fields

- BYTE switchOption

#### 8.1482.1 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 |

#### 8.1482.2 Field Documentation

##### 8.1482.2.1 BYTE voiceALSSetLineSwitchInfo::switchOption

## 8.1483 voiceAnswerCall Struct Reference

### Data Fields

- BYTE * pCallId

### 8.1483.1 Detailed Description

Contains the parameters passed for SLQSVoiceAnswerCall.

**Parameters**

| | |
|---|---|
| *pCallId[IN/OUT]* | • Unique call identifier for the call that must be answered. |

### 8.1483.2 Field Documentation

#### 8.1483.2.1 BYTE* voiceAnswerCall::pCallId

## 8.1484 voiceBindSubscriptionInfo Struct Reference

### Data Fields

- BYTE subsType

### 8.1484.1 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 |

### 8.1484.2 Field Documentation

#### 8.1484.2.1 BYTE voiceBindSubscriptionInfo::subsType

## 8.1485 voiceBurstDTMFInfo Struct Reference

### Data Fields

- burstDTMFInfo BurstDTMFInfo
- DTMFLengths * pBurstDTMFLengths

### 8.1485.1 Detailed Description

This structure contains parameters of burst Dual-Tone Multifrequency (DTMF)

**Parameters**

| | |
|---|---|
| *BurstDTMFInfo* | • Burst DTMF Information<br>– See burstDTMFInfo for more information |

| | |
|---|---|
| *pBurstDTMF-Lengths* | [optional]<br>• DTMF Lengths<br>– See DTMFLengths for more information |

### 8.1485.2 Field Documentation

#### 8.1485.2.1 burstDTMFInfo voiceBurstDTMFInfo::BurstDTMFInfo

#### 8.1485.2.2 DTMFLengths* voiceBurstDTMFInfo::pBurstDTMFLengths

## 8.1486 voiceCallInfoReq Struct Reference

### Data Fields

- BYTE callID

### 8.1486.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *callID* | • Call identifier for the call queried for information. |

### 8.1486.2 Field Documentation

#### 8.1486.2.1 BYTE voiceCallInfoReq::callID

## 8.1487 voiceCallInfoResp Struct Reference

### Data Fields

- callInfo * pCallInfo
- remotePartyNum * pRemotePartyNum
- WORD * pSrvOpt
- BYTE * pVoicePrivacy
- BYTE * pOTASPStatus
- remotePartyName * pRemotePartyName
- UUSInfo * pUUSInfo
- BYTE * pAlertType
- alphaIDInfo * pAlphaIDInfo
- connectNumInfo * pConnectNumInfo
- diagInfo * pDiagInfo
- ULONG * pAlertingPattern

### 8.1487.1 Detailed Description

This structure contains information of the response parameters associated with a call.

**Parameters**

| | |
|---|---|
| *pCallInfo(optional)* | • See callInfo for more information. |
| *pRemotePartyNum(optional)* | • See remotePartyNum for more information. |
| *pSrvOpt* | • Service option(optional)<br>• Applicable only for 3GPP2 devices.<br>• See Table8 qaGobiApiTableServiceOptions.h for standard service option number assignments. |
| *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 |
| *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 |
| *pRemotePartyName(optional)* | • Applicable only for 3GPP devices.<br>• See remotePartyName for more information. |

| | |
|---|---|
| *pUUSInfo(optional)* | • Applicable only for 3GPP devices.<br>• See UUSInfo for more information. |
| *pAlertType(optional)* | • Alerting type.<br>• Applicable only for 3GPP devices.<br>– 0x00 - ALERTING_LOCAL - Local<br>– 0x01 - ALERTING_REMOTE - Remote<br>– 0xFF - Not Available |
| *pAlphaIDInfo(optional)* | • Applicable only for 3GPP devices.<br>• See alphaIDInfo for more information. |
| *pConnectNumInfo(optional)* | • See connectNumInfo for more information. |
| *pDiagInfo(optional)* | • See diagInfo for more information. |
| *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 |

### 8.1487.2 Field Documentation

#### 8.1487.2.1 ULONG* voiceCallInfoResp::pAlertingPattern

#### 8.1487.2.2 BYTE* voiceCallInfoResp::pAlertType

#### 8.1487.2.3 alphaIDInfo* voiceCallInfoResp::pAlphaIDInfo

#### 8.1487.2.4 callInfo* voiceCallInfoResp::pCallInfo

#### 8.1487.2.5 connectNumInfo* voiceCallInfoResp::pConnectNumInfo

#### 8.1487.2.6 diagInfo* voiceCallInfoResp::pDiagInfo

#### 8.1487.2.7 BYTE* voiceCallInfoResp::pOTASPStatus

#### 8.1487.2.8 remotePartyName* voiceCallInfoResp::pRemotePartyName

#### 8.1487.2.9 remotePartyNum∗ voiceCallInfoResp::pRemotePartyNum

#### 8.1487.2.10 WORD∗ voiceCallInfoResp::pSrvOpt

#### 8.1487.2.11 UUSInfo∗ voiceCallInfoResp::pUUSInfo

#### 8.1487.2.12 BYTE∗ voiceCallInfoResp::pVoicePrivacy

## 8.1488 voiceCallRequestParams Struct Reference

### Data Fields

- BYTE callNumber [81]
- BYTE ∗ pCallType
- BYTE ∗ pCLIRType
- UUSInfo ∗ pUUSInFo
- CUGInfo ∗ pCUGInfo
- BYTE ∗ pEmergencyCategory
- calledPartySubAdd ∗ pCallPartySubAdd
- ULONG ∗ pSvcType

### 8.1488.1 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] |
| *pCallType(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 |
| *pCLIRType(optional)* | • CLIR type values are:<br>– 0x01 - CLIR_SUPPRESSION - Suppression<br>– 0x02 - CLIR_INVOCATION - Invocation |
| *pUUSInFo(optional)* | • Pointer to structure of UUSInfo<br>– See UUSInfo for more information |
| *pCUGInfo(optional)* | • Pointer to structure of CUGInfo<br>– See 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 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 |

### 8.1488.2 Field Documentation

#### 8.1488.2.1 BYTE voiceCallRequestParams::callNumber[81]

#### 8.1488.2.2 calledPartySubAdd* voiceCallRequestParams::pCallPartySubAdd

#### 8.1488.2.3 BYTE* voiceCallRequestParams::pCallType

#### 8.1488.2.4 BYTE* voiceCallRequestParams::pCLIRType

#### 8.1488.2.5 CUGInfo* voiceCallRequestParams::pCUGInfo

#### 8.1488.2.6 BYTE* voiceCallRequestParams::pEmergencyCategory

#### 8.1488.2.7 ULONG* voiceCallRequestParams::pSvcType

#### 8.1488.2.8 UUSInfo* voiceCallRequestParams::pUUSInFo

## 8.1489 voiceCallResponseParams Struct Reference

### Data Fields

- BYTE * pCallID
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- ccSUPSType * pCCSUPSType

### 8.1489.1 Detailed Description

This structure contains Voice Call Response Parameters

**Parameters**

| | |
|---|---|
| *pCallID(optional)* | • Unique call identifier for the dialed call |
| *pAlphaID-Info(optional)* | • Pointer to structure of alphaIDInfo<br>– See alphaIDInfo for more information |
| *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 |
| *pCCSUPS-Type(optional)* | • Pointer to structure of ccSUPSType<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See ccSUPSType for more information |

### 8.1489.2 Field Documentation

#### 8.1489.2.1 alphaIDInfo∗ voiceCallResponseParams::pAlphaIDInfo

#### 8.1489.2.2 BYTE∗ voiceCallResponseParams::pCallID

#### 8.1489.2.3 BYTE∗ voiceCallResponseParams::pCCResultType

#### 8.1489.2.4 ccSUPSType∗ voiceCallResponseParams::pCCSUPSType

## 8.1490 voiceContDTMFinfo Struct Reference

### Data Fields

- BYTE ∗ pCallID
- BYTE DTMFdigit

### 8.1490.1 Detailed Description

This structure contains parameters of continuous DTMF

**Parameters**

| | |
|---|---|
| *pCallID[IN/OUT]* | • 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[IN]* | • DTMF digit in ASCII. |

### 8.1490.2 Field Documentation

#### 8.1490.2.1 BYTE voiceContDTMFinfo::DTMFdigit

#### 8.1490.2.2 BYTE* voiceContDTMFinfo::pCallID

## 8.1491 voiceDTMFEventInfo Struct Reference

### Data Fields

- DTMFInfo DTMFInformation
- BYTE * pOnLength
- BYTE * pOffLength

### 8.1491.1 Detailed Description

This structure contains the parameters passed for SLQSVoiceSetDTMFEventCallBack by the device.

**Parameters**

| | |
|---|---|
| *DTMF-Information(mandatory)* | See 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

### 8.1491.2 Field Documentation

#### 8.1491.2.1 DTMFInfo voiceDTMFEventInfo::DTMFInformation

#### 8.1491.2.2 BYTE* voiceDTMFEventInfo::pOffLength

#### 8.1491.2.3 BYTE* voiceDTMFEventInfo::pOnLength

## 8.1492 voiceFlashInfo Struct Reference

### Data Fields

- BYTE * pCallID
- BYTE * pFlashPayLd
- BYTE * pFlashType

### 8.1492.1 Detailed Description

This structure contains the flash information associated with a call.

**Parameters**

| | |
|---|---|
| *pCallID[IN/OUT]* | • Unique call identifier associated with the current call. |
| *pFlashPayLd[IN](optional)* | • Payload in ASCII to be sent in Flash.<br>• Variable Length, NULL terminated. |
| *pFlashType[IN](optional)* | • Flash type.<br>– 0 - Simple Flash (default)<br>– 1 - Activate answer hold<br>– 2 - Deactivate answer hold |

### 8.1492.2 Field Documentation

#### 8.1492.2.1 BYTE* voiceFlashInfo::pCallID

#### 8.1492.2.2 BYTE* voiceFlashInfo::pFlashPayLd

#### 8.1492.2.3 BYTE* voiceFlashInfo::pFlashType

## 8.1493 voiceGetAllCallInfo Struct Reference

### Data Fields

- arrCallInfo * pArrCallInfo

- arrRemotePartyNum * pArrRemotePartyNum
- arrRemotePartyName * pArrRemotePartyName
- arrAlertingType * pArrAlertingType
- arrUUSInfo * pArrUUSInfo
- arrSvcOption * pArrSvcOption
- BYTE * pOTASPStatus
- BYTE * pVoicePrivacy
- arrCallEndReason * pArrCallEndReason
- arrAlphaID * pArrAlphaID
- arrConnectPartyNum * pArrConnectPartyNum
- arrDiagInfo * pArrDiagInfo
- arrCalledPartyNum * pArrCalledPartyNum
- arrRedirPartyNum * pArrRedirPartyNum
- arrAlertingPattern * pArrAlertingPattern

### 8.1493.1 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 arrCallInfo for more information. |
| *pArrRemotePartyNum(optional)* | • See arrRemotePartyNum for more information. |
| *pArrRemotePartyName(optional)* | • See arrRemotePartyName for more information. |
| *pArrAlertingType(optional)* | • See arrAlertingType for more information. |
| *pArrUUSInfo(optional)* | • See arrUUSInfo for more information. |
| *pArrSvcOption(optional)* | • See arrSvcOption for more information. |

| | |
|---|---|
| *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 |
| *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 |
| *pArrCallEnd-Reason(optional)* | • See arrCallEndReason for more information. |
| *pArrAlphaI-D(optional)* | • See arrAlphaID for more information. |
| *pArrConnect-Party-Num(optional)* | • See arrConnectPartyNum for more information. |
| *pArrDiag-Info(optional)* | • See arrDiagInfo for more information. |
| *pArrCalledParty-Num(optional)* | • See arrCalledPartyNum for more information. |

| | |
|---|---|
| *pArrRedirParty-Num(optional)* | • See arrRedirPartyNum for more information. |
| *pArrAlerting-Pattern(optional)* | • See arrAlertingPattern for more information. |

### 8.1493.2 Field Documentation

#### 8.1493.2.1 arrAlertingPattern∗ voiceGetAllCallInfo::pArrAlertingPattern

#### 8.1493.2.2 arrAlertingType∗ voiceGetAllCallInfo::pArrAlertingType

#### 8.1493.2.3 arrAlphaID∗ voiceGetAllCallInfo::pArrAlphaID

#### 8.1493.2.4 arrCalledPartyNum∗ voiceGetAllCallInfo::pArrCalledPartyNum

#### 8.1493.2.5 arrCallEndReason∗ voiceGetAllCallInfo::pArrCallEndReason

#### 8.1493.2.6 arrCallInfo∗ voiceGetAllCallInfo::pArrCallInfo

#### 8.1493.2.7 arrConnectPartyNum∗ voiceGetAllCallInfo::pArrConnectPartyNum

#### 8.1493.2.8 arrDiagInfo∗ voiceGetAllCallInfo::pArrDiagInfo

#### 8.1493.2.9 arrRedirPartyNum∗ voiceGetAllCallInfo::pArrRedirPartyNum

#### 8.1493.2.10 arrRemotePartyName∗ voiceGetAllCallInfo::pArrRemotePartyName

#### 8.1493.2.11 arrRemotePartyNum∗ voiceGetAllCallInfo::pArrRemotePartyNum

#### 8.1493.2.12 arrSvcOption∗ voiceGetAllCallInfo::pArrSvcOption

#### 8.1493.2.13 arrUUSInfo∗ voiceGetAllCallInfo::pArrUUSInfo

#### 8.1493.2.14 BYTE∗ voiceGetAllCallInfo::pOTASPStatus

#### 8.1493.2.15 BYTE∗ voiceGetAllCallInfo::pVoicePrivacy

## 8.1494 voiceGetCallBarringReq Struct Reference

### Data Fields

- BYTE reason
- BYTE ∗ pSvcClass

### 8.1494.1 Detailed Description

This structure contains Voice Get Call Barring Request 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[IN/OUT]* | • 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 |

### 8.1494.2 Field Documentation

#### 8.1494.2.1 BYTE* voiceGetCallBarringReq::pSvcClass

#### 8.1494.2.2 BYTE voiceGetCallBarringReq::reason

## 8.1495 voiceGetCallBarringResp Struct Reference

### Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

### 8.1495.1 Detailed Description

This structure contains Voice Get Call Barring Response Parameters

**Parameters**

| | |
|---|---|
| *pSvcClass[IN/OUT]* | • 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 |

| | |
|---|---|
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |
| *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 ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1495.2 Field Documentation

#### 8.1495.2.1 alphaIDInfo* voiceGetCallBarringResp::pAlphaIDInfo

#### 8.1495.2.2 BYTE* voiceGetCallBarringResp::pCallID

#### 8.1495.2.3 BYTE* voiceGetCallBarringResp::pCCResType

#### 8.1495.2.4 ccSUPSType* voiceGetCallBarringResp::pCCSUPSType

#### 8.1495.2.5 WORD* voiceGetCallBarringResp::pFailCause

#### 8.1495.2.6 BYTE* voiceGetCallBarringResp::pSvcClass

## 8.1496 voiceGetCallFWReq Struct Reference

### Data Fields

- BYTE Reason
- BYTE * pSvcClass

### 8.1496.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Request 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. |

### 8.1496.2 Field Documentation

#### 8.1496.2.1 BYTE* voiceGetCallFWReq::pSvcClass

#### 8.1496.2.2 BYTE voiceGetCallFWReq::Reason

## 8.1497 voiceGetCallFWResp Struct Reference

### Data Fields

- getCallFWInfo * pGetCallFWInfo
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType
- getCallFWExtInfo * pGetCallFWExtInfo

### 8.1497.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Response Parameters

**Parameters**

| | |
|---|---|
| *pGetCallFWInfo* | • Pointer to structure of getCallFWInfo (optional)<br>– See getCallFWInfo for more information |

| | |
|---|---|
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |
| *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 ccSUPSType for more information |
| *pGetCallFWExt-Info* | • Pointer to structure of getCallFWExtInfo (optional)<br>– See getCallFWExtInfo for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1497.2 Field Documentation

#### 8.1497.2.1 alphaIDInfo∗ voiceGetCallFWResp::pAlphaIDInfo

#### 8.1497.2.2 BYTE∗ voiceGetCallFWResp::pCallID

#### 8.1497.2.3 BYTE∗ voiceGetCallFWResp::pCCResType

#### 8.1497.2.4 ccSUPSType∗ voiceGetCallFWResp::pCCSUPSType

#### 8.1497.2.5 WORD∗ voiceGetCallFWResp::pFailCause

#### 8.1497.2.6 getCallFWExtInfo∗ voiceGetCallFWResp::pGetCallFWExtInfo

#### 8.1497.2.7 getCallFWInfo∗ voiceGetCallFWResp::pGetCallFWInfo

## 8.1498 voiceGetCallWaitInfo Struct Reference

### Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

## 8.1498.1 Detailed Description

This structure contains Voice Get Call Waiting Response Parameters

**Parameters**

| | |
|---|---|
| *pSvcClass[IN/OUT]* | • 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 |
| *pFailCause[OUT]* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo[OUT]* | • Pointer to structure of alphaIDInfo (optional)<br>– See alphaIDInfo for more information |
| *pCCResType[OUT]* | • 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 |
| *pCallID[OUT]* | • 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. |
| *pCCSUPSType[OUT]* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

## 8.1498.2 Field Documentation

#### 8.1498.2.1 alphaIDInfo* voiceGetCallWaitInfo::pAlphaIDInfo

#### 8.1498.2.2 BYTE* voiceGetCallWaitInfo::pCallID

#### 8.1498.2.3 BYTE* voiceGetCallWaitInfo::pCCResType

#### 8.1498.2.4 ccSUPSType* voiceGetCallWaitInfo::pCCSUPSType

#### 8.1498.2.5 WORD* voiceGetCallWaitInfo::pFailCause

#### 8.1498.2.6 BYTE* voiceGetCallWaitInfo::pSvcClass

## 8.1499 voiceGetCLIPResp Struct Reference

### Data Fields

- CLIPResp * pCLIPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

### 8.1499.1 Detailed Description

This structure contains Voice Get Calling Line Identification Presentation(CLIP) Response Parameters

**Parameters**

| | |
|---|---|
| *pCLIPResp* | • Pointer to structure of CLIPResp (optional)<br>– See CLIPResp for more information |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |

| | |
|---|---|
| *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 ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1499.2 Field Documentation

#### 8.1499.2.1 alphaIDInfo* voiceGetCLIPResp::pAlphaIDInfo

#### 8.1499.2.2 BYTE* voiceGetCLIPResp::pCallID

#### 8.1499.2.3 BYTE* voiceGetCLIPResp::pCCResType

#### 8.1499.2.4 ccSUPSType* voiceGetCLIPResp::pCCSUPSType

#### 8.1499.2.5 CLIPResp* voiceGetCLIPResp::pCLIPResp

#### 8.1499.2.6 WORD* voiceGetCLIPResp::pFailCause

## 8.1500 voiceGetCLIRResp Struct Reference

### Data Fields

- CLIRResp * pCLIRResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

### 8.1500.1 Detailed Description

This structure contains Voice Get Calling Line Identification Restriction (CLIR) Response Parameters

**Parameters**

| | |
|---|---|
| *pCLIRResp* | • Pointer to structure of CLIRResp (optional)<br>– See CLIRResp for more information |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |
| *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 ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1500.2 Field Documentation

#### 8.1500.2.1 alphaIDInfo∗ voiceGetCLIRResp::pAlphaIDInfo

#### 8.1500.2.2 BYTE∗ voiceGetCLIRResp::pCallID

#### 8.1500.2.3 BYTE∗ voiceGetCLIRResp::pCCResType

#### 8.1500.2.4 ccSUPSType∗ voiceGetCLIRResp::pCCSUPSType

#### 8.1500.2.5 CLIRResp∗ voiceGetCLIRResp::pCLIRResp

#### 8.1500.2.6 WORD∗ voiceGetCLIRResp::pFailCause

## 8.1501 voiceGetCNAPResp Struct Reference

### Data Fields

- CNAPResp ∗ pCNAPResp
- WORD ∗ pFailCause
- alphaIDInfo ∗ pAlphaIDInfo
- BYTE ∗ pCCResType
- BYTE ∗ pCallID
- ccSUPSType ∗ pCCSUPSType

### 8.1501.1 Detailed Description

This structure contains Voice Get Calling Name Presentation(CNAP) Response Parameters

**Parameters**

| | |
|---|---|
| *pCNAPResp* | • Pointer to structure of CNAPResp (optional)<br>– See CNAPResp for more information |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |
| *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 ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1501.2 Field Documentation

#### 8.1501.2.1 alphaIDInfo* voiceGetCNAPResp::pAlphaIDInfo

#### 8.1501.2.2 BYTE* voiceGetCNAPResp::pCallID

#### 8.1501.2.3 BYTE* voiceGetCNAPResp::pCCResType

#### 8.1501.2.4 ccSUPSType* voiceGetCNAPResp::pCCSUPSType

#### 8.1501.2.5 CNAPResp* voiceGetCNAPResp::pCNAPResp

#### 8.1501.2.6 WORD* voiceGetCNAPResp::pFailCause

## 8.1502 voiceGetCOLPResp Struct Reference

### Data Fields

- COLPResp * pCOLPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

### 8.1502.1 Detailed Description

This structure contains Voice Get Connected Line Identification Presentation(COLP) Response Parameters

**Parameters**

| | |
|---|---|
| *pCOLPResp* | • Pointer to structure of COLPResp (optional)<br>– See COLPResp for more information |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |
| *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 ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1502.2 Field Documentation

#### 8.1502.2.1 alphaIDInfo* voiceGetCOLPResp::pAlphaIDInfo

#### 8.1502.2.2 BYTE* voiceGetCOLPResp::pCallID

#### 8.1502.2.3 BYTE* voiceGetCOLPResp::pCCResType

#### 8.1502.2.4 ccSUPSType* voiceGetCOLPResp::pCCSUPSType

#### 8.1502.2.5 COLPResp* voiceGetCOLPResp::pCOLPResp

#### 8.1502.2.6 WORD* voiceGetCOLPResp::pFailCause

## 8.1503 voiceGetCOLRResp Struct Reference

### Data Fields

- COLRResp * pCOLRResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

### 8.1503.1 Detailed Description

This structure contains Voice Get Connected Line Identification Restriction(COLR) Response Parameters

**Parameters**

| | |
|---|---|
| *pCOLRResp* | • Pointer to structure of COLRResp (optional)<br>– See COLRResp for more information |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |

| | |
|---|---|
| *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 ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1503.2 Field Documentation

#### 8.1503.2.1 alphaIDInfo∗ voiceGetCOLRResp::pAlphaIDInfo

#### 8.1503.2.2 BYTE∗ voiceGetCOLRResp::pCallID

#### 8.1503.2.3 BYTE∗ voiceGetCOLRResp::pCCResType

#### 8.1503.2.4 ccSUPSType∗ voiceGetCOLRResp::pCCSUPSType

#### 8.1503.2.5 COLRResp∗ voiceGetCOLRResp::pCOLRResp

#### 8.1503.2.6 WORD∗ voiceGetCOLRResp::pFailCause

## 8.1504 voiceGetConfigReq Struct Reference

### Data Fields

- BYTE ∗ pAutoAnswer
- BYTE ∗ pAirTimer
- BYTE ∗ pRoamTimer
- BYTE ∗ pTTYMode
- BYTE ∗ pPrefVoiceSO
- BYTE ∗ pAMRStatus
- BYTE ∗ pPrefVoicePrivacy
- BYTE ∗ pNamID
- BYTE ∗ pVoiceDomainPref

### 8.1504.1 Detailed Description

This structure contains Voice Get Configuration Request Parameters

**Parameters**

| | |
|---|---|
| *pAuto-Answer(optional)* | • Indicator to retrieve the Auto Answer Information.<br>– 0x01 - Include auto answer information |
| *pAir-Timer(optional)* | • Indicator to retrieve the Air Timer Information.<br>– 0x01 - Include air calls timer count information<br>• Currently Not Supported. |

| | |
|---|---|
| *pRoam-Timer(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.

## 8.1504.2 Field Documentation

### 8.1504.2.1 BYTE* voiceGetConfigReq::pAirTimer

### 8.1504.2.2 BYTE* voiceGetConfigReq::pAMRStatus

### 8.1504.2.3 BYTE* voiceGetConfigReq::pAutoAnswer

### 8.1504.2.4 BYTE* voiceGetConfigReq::pNamID

### 8.1504.2.5 BYTE* voiceGetConfigReq::pPrefVoicePrivacy

### 8.1504.2.6 BYTE* voiceGetConfigReq::pPrefVoiceSO

#### 8.1504.2.7 BYTE* voiceGetConfigReq::pRoamTimer

#### 8.1504.2.8 BYTE* voiceGetConfigReq::pTTYMode

#### 8.1504.2.9 BYTE* voiceGetConfigReq::pVoiceDomainPref

## 8.1505 voiceGetConfigResp Struct Reference

### Data Fields

- BYTE * pAutoAnswerStat
- airTimer * pAirTimerCnt
- roamTimer * pRoamTimerCnt
- BYTE * pCurrTTYMode
- prefVoiceSO * pCurPrefVoiceSO
- curAMRConfig * pCurAMRConfig
- BYTE * pCurVoicePrivacyPref
- BYTE * pCurVoiceDomainPref

### 8.1505.1 Detailed Description

This structure contains Voice Get Configuration Response Parameters.

**Parameters**

| | |
|---|---|
| *pAutoAnswer-Stat(optional)* | • Auto Answer Status<br>• Value returned is read from NV_AUTO_ANSWER_I.<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>– 0xFF - Not Available |
| *pAirTimer-Cnt(optional)* | • Air Timer Count<br>• Value returned is read from NV_AIR_CNT_I.<br>• See airTimer for more information |
| *pRoamTimer-Cnt(optional)* | • Roam Timer Count<br>• Value returned is read from NV_ROAM_CNT_I.<br>• See roamTimer for more information |
| *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 |

| | |
|---|---|
| *pCurPrefVoiceSO(optional)* | • Current Preferred Voice SO<br>• Value returned is read from NV_PREF_VOICE_SO_I.<br>• See prefVoiceSO for more information |
| *pCurAMRConfig(optional)* | • Current Adaptive Multi-Rate Configuration.<br>• Values returned are read from NV_GSM_ARM_CALL_CONFIG_I and NV_UMTS_AMR_CODEC_PREFERENCE_CONFIG_I.<br>• See curAMRConfig for more information |
| *pCurVoicePrivacyPref(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 |
| *pCurVoiceDomainPref(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 |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1505.2 Field Documentation

#### 8.1505.2.1 airTimer* voiceGetConfigResp::pAirTimerCnt

#### 8.1505.2.2 BYTE* voiceGetConfigResp::pAutoAnswerStat

#### 8.1505.2.3 curAMRConfig* voiceGetConfigResp::pCurAMRConfig

#### 8.1505.2.4 prefVoiceSO* voiceGetConfigResp::pCurPrefVoiceSO

#### 8.1505.2.5 BYTE* voiceGetConfigResp::pCurrTTYMode

#### 8.1505.2.6 BYTE* voiceGetConfigResp::pCurVoiceDomainPref

#### 8.1505.2.7 BYTE* voiceGetConfigResp::pCurVoicePrivacyPref

#### 8.1505.2.8 roamTimer* voiceGetConfigResp::pRoamTimerCnt

## 8.1506 voiceIndicationRegisterInfo Struct Reference

### Data Fields

- BYTE * pRegDTMFEvents
- BYTE * pRegVoicePrivacyEvents
- BYTE * pSuppsNotifEvents

#### 8.1506.1 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.

#### 8.1506.2 Field Documentation

##### 8.1506.2.1 BYTE* voiceIndicationRegisterInfo::pRegDTMFEvents

##### 8.1506.2.2 BYTE* voiceIndicationRegisterInfo::pRegVoicePrivacyEvents

##### 8.1506.2.3 BYTE* voiceIndicationRegisterInfo::pSuppsNotifEvents

## 8.1507 voiceInfoRec Struct Reference

### Data Fields

- BYTE callID
- signalInfo * pSignalInfo
- callerIDInfo * pCallerIDInfo

- BYTE * pDispInfo
- BYTE * pExtDispInfo
- BYTE * pCallerNameInfo
- BYTE * pCallWaitInd
- connectNumInfo * pConnectNumInfo
- connectNumInfo * pCallingPartyInfo
- calledPartyInfo * pCalledPartyInfo
- redirNumInfo * pRedirNumInfo
- BYTE * pCLIRCause
- NSSAudioCtrl * pNSSAudioCtrl
- BYTE * pNSSRelease
- lineCtrlInfo * pLineCtrlInfo
- extDispRecInfo * pExtDispRecInfo

### 8.1507.1 Detailed Description

This structure contains Voice record Information

**Parameters**

| | |
|---|---|
| *callID* | [Mandatory]<br>• Call identifier for the call. |
| *pSignalInfo[-Optional]* | • Signal Information<br>• See signalInfo for more information |
| *pCallerIDInfo[-Optional]* | • Caller ID Information<br>• See 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 connectNumInfo for more information |
| *pCallingParty-Info[Optional]* | • Calling Party Number Information<br>• This structure is having exactly same elements as connectNumInfo<br>• see connectNumInfo for more information |
| *pCalledParty-Info[Optional]* | • Called Party Number Information<br>• see calledPartyInfo for more information |

| | |
|---|---|
| *pRedirNumInfo[-Optional]* | • Redirecting Number Information<br>• see redirNumInfo for more information |
| *pCLIRCause[-Optional]* | • National Supplementary Services - CLIR<br>• see NSSAudioCtrl for more information |
| *pNSSAudioCtrl[-Optional]* | • National Supplementary Services - Audio Control |
| *pNSSRelease[-Optional]* | • National Supplementary Services - Release |
| *pLineCtrlInfo[-Optional]* | • Line Control Information<br>• see lineCtrlInfo for more information |
| *pExtDispRec-Info[Optional]* | • Extended Display Record Information<br>• see extDispRecInfo for more information |

### 8.1507.2 Field Documentation

#### 8.1507.2.1 BYTE voiceInfoRec::callID

#### 8.1507.2.2 calledPartyInfo* voiceInfoRec::pCalledPartyInfo

#### 8.1507.2.3 callerIDInfo* voiceInfoRec::pCallerIDInfo

#### 8.1507.2.4 BYTE* voiceInfoRec::pCallerNameInfo

#### 8.1507.2.5 connectNumInfo* voiceInfoRec::pCallingPartyInfo

#### 8.1507.2.6 BYTE* voiceInfoRec::pCallWaitInd

#### 8.1507.2.7 BYTE* voiceInfoRec::pCLIRCause

#### 8.1507.2.8 connectNumInfo* voiceInfoRec::pConnectNumInfo

#### 8.1507.2.9 BYTE* voiceInfoRec::pDispInfo

#### 8.1507.2.10 BYTE* voiceInfoRec::pExtDispInfo

#### 8.1507.2.11 extDispRecInfo* voiceInfoRec::pExtDispRecInfo

#### 8.1507.2.12 lineCtrlInfo* voiceInfoRec::pLineCtrlInfo

#### 8.1507.2.13 NSSAudioCtrl* voiceInfoRec::pNSSAudioCtrl

#### 8.1507.2.14 BYTE* voiceInfoRec::pNSSRelease

#### 8.1507.2.15 redirNumInfo∗ voiceInfoRec::pRedirNumInfo

#### 8.1507.2.16 signalInfo∗ voiceInfoRec::pSignalInfo

## 8.1508 voiceManageCallsReq Struct Reference

### Data Fields

- BYTE SUPSType
- BYTE ∗ pCallID

### 8.1508.1 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 |

### 8.1508.2 Field Documentation

#### 8.1508.2.1 BYTE∗ voiceManageCallsReq::pCallID

#### 8.1508.2.2 BYTE voiceManageCallsReq::SUPSType

## 8.1509 voiceManageCallsResp Struct Reference

### Data Fields

- WORD * pFailCause

#### 8.1509.1 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 |

#### 8.1509.2 Field Documentation

##### 8.1509.2.1 WORD* voiceManageCallsResp::pFailCause

## 8.1510 voiceOrigUSSDNoWaitInfo Struct Reference

### Data Fields

- struct USSInfo USSInformation

#### 8.1510.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

**Parameters**

| | |
|---|---|
| *USSInformation* | • See USSInfo for more information. |

#### 8.1510.2 Field Documentation

##### 8.1510.2.1 struct USSInfo voiceOrigUSSDNoWaitInfo::USSInformation

## 8.1511 voiceOTASPStatusInfo Struct Reference

### Data Fields

- BYTE callID
- BYTE OTASPStatus

#### 8.1511.1 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) |

### 8.1511.2 Field Documentation

#### 8.1511.2.1 BYTE voiceOTASPStatusInfo::callID

#### 8.1511.2.2 BYTE voiceOTASPStatusInfo::OTASPStatus

## 8.1512 voicePrivacyInfo Struct Reference

### Data Fields

- BYTE callID
- BYTE voicePrivacy

### 8.1512.1 Detailed Description

Contains the parameters passed for SLQSVoiceSetPrivacyChangeCallBack by the device.

**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

### 8.1512.2 Field Documentation

#### 8.1512.2.1 BYTE voicePrivacyInfo::callID

#### 8.1512.2.2 BYTE voicePrivacyInfo::voicePrivacy

## 8.1513 voiceSetAllCallStatusCbkInfo Struct Reference

### Data Fields

- arrCallInfo arrCallInfomation
- arrRemotePartyNum * pArrRemotePartyNum
- arrRemotePartyName * pArrRemotePartyName
- arrAlertingType * pArrAlertingType
- arrSvcOption * pArrSvcOption
- arrCallEndReason * pArrCallEndReason
- arrAlphaID * pArrAlphaID
- arrConnectPartyNum * pArrConnectPartyNum
- arrDiagInfo * pArrDiagInfo
- arrCalledPartyNum * pArrCalledPartyNum
- arrRedirPartyNum * pArrRedirPartyNum
- arrAlertingPattern * pArrAlertingPattern

### 8.1513.1 Detailed Description

This structure contains VoiceCall Information parameters. arrCallInfomation will be populated in case of change in the call information. Other paramters are optional therefore are populated based on device and technology type being used.

**Parameters**

| | |
|---|---|
| *arrCallInfomation* | [mandatory]<br>• Array of Call Information<br>This must be populated if Indication is received See arrCallInfo for more information.<br>– Applicable for both "3GPP/3GPP2" |
| *pArrRemote-PartyNum* | [optional]<br>• Array of Remote Party Name.( NULL when not present)<br>See arrRemotePartyNum for more information.<br>– Applicable only for "3GPP/3GPP2" |
| | |

| | |
|---|---|
| *pArrRemote-PartyName* | [optional]<br>• Array of Alerting Type.( NULL when not present)<br>See arrRemotePartyName for more information.<br>– Applicable only for "3GPP" |
| *pArrAlertingType* | [optional]<br>• Array of Alerting Type( NULL when not present)<br>See arrAlertingType for more information.<br>– Applicable only for "3GPP" |
| *pArrSvcOption* | [optional]<br>• Array of Service Option.(NULL when not present)<br>See arrSvcOption for more information.<br>– Applicable only for "3GPP" |
| *pArrCallEnd-Reason* | [optional]<br>• Array of Call End Reason.( NULL when not present)<br>See arrCallEndReason for more information.<br>– Applicable only for "3GPP" |
| *pArrAlphaID* | [optional]<br>• Array of Alpha Identifier( NULL when not present)<br>See arrAlphaID for more information.<br>– Applicable only for "3GPP" |
| *pArrConnect-PartyNum* | [optional]<br>• Array of Connected Party Number.( NULL when not present)<br>See arrConnectPartyNum for more information.<br>– Applicable for both "3GPP/3GPP2" |
| *pArrDiagInfo* | [optional]<br>• Array of Diagnostic Information.( NULL when not present)<br>See arrDiagInfo for more information.<br>– Applicable only for "3GPP" |
| *pArrCalledParty-Num* | [optional]<br>• Array of Called Party Number.( NULL when not present)<br>See arrCalledPartyNum for more information.<br>– Applicable only for "3GPP" |
| *pArrRedirParty-Num* | [optional]<br>• Array of Redirecting Party Number.( NULL when not present)<br>See arrRedirPartyNum for more information.<br>– Applicable only for "3GPP" |
| *pArrAlerting-Pattern* | [optional]<br>• Array of Alerting Pattern.( NULL when not present)<br>See arrAlertingPattern for more information.<br>– Applicable only for "3GPP" |

**Note**

Optional paramters would be NULL, if not received from the device.

### 8.1513.2 Field Documentation

#### 8.1513.2.1 arrCallInfo voiceSetAllCallStatusCbkInfo::arrCallInfomation

#### 8.1513.2.2 arrAlertingPattern* voiceSetAllCallStatusCbkInfo::pArrAlertingPattern

#### 8.1513.2.3 arrAlertingType* voiceSetAllCallStatusCbkInfo::pArrAlertingType

#### 8.1513.2.4 arrAlphaID* voiceSetAllCallStatusCbkInfo::pArrAlphaID

#### 8.1513.2.5 arrCalledPartyNum* voiceSetAllCallStatusCbkInfo::pArrCalledPartyNum

#### 8.1513.2.6 arrCallEndReason* voiceSetAllCallStatusCbkInfo::pArrCallEndReason

#### 8.1513.2.7 arrConnectPartyNum* voiceSetAllCallStatusCbkInfo::pArrConnectPartyNum

#### 8.1513.2.8 arrDiagInfo* voiceSetAllCallStatusCbkInfo::pArrDiagInfo

#### 8.1513.2.9 arrRedirPartyNum* voiceSetAllCallStatusCbkInfo::pArrRedirPartyNum

#### 8.1513.2.10 arrRemotePartyName* voiceSetAllCallStatusCbkInfo::pArrRemotePartyName

#### 8.1513.2.11 arrRemotePartyNum* voiceSetAllCallStatusCbkInfo::pArrRemotePartyNum

#### 8.1513.2.12 arrSvcOption* voiceSetAllCallStatusCbkInfo::pArrSvcOption

## 8.1514 voiceSetCallBarringPwdInfo Struct Reference

### Data Fields

- BYTE Reason
- BYTE oldPasswd [4]
- BYTE newPasswd [4]
- BYTE newPasswdAgain [4]

### 8.1514.1 Detailed Description

This structure contains Voice Set Call Barring Password Request 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[PASSWORD_LENGTH]* | • Old password.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |
| *newPasswd[PASSWORD_LENGTH]* | • New password.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |
| *newPasswdAgain[PASSWORD_LENGTH]* | • New password Again.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |

### 8.1514.2 Field Documentation

#### 8.1514.2.1 BYTE voiceSetCallBarringPwdInfo::newPasswd[4]

#### 8.1514.2.2 BYTE voiceSetCallBarringPwdInfo::newPasswdAgain[4]

#### 8.1514.2.3 BYTE voiceSetCallBarringPwdInfo::oldPasswd[4]

#### 8.1514.2.4 BYTE voiceSetCallBarringPwdInfo::Reason

## 8.1515 voiceSetCallBarringPwdResp Struct Reference

### Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

### 8.1515.1 Detailed Description

This structure contains Voice Set Call Barring Password Response Parameters

**Parameters**

| | |
|---|---|
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See 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 |
| *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. |
| *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 ccSUPSType for more information |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.1515.2 Field Documentation

#### 8.1515.2.1 alphaIDInfo* voiceSetCallBarringPwdResp::pAlphaIDInfo

#### 8.1515.2.2 BYTE* voiceSetCallBarringPwdResp::pCallID

#### 8.1515.2.3 BYTE* voiceSetCallBarringPwdResp::pCCResType

#### 8.1515.2.4 ccSUPSType* voiceSetCallBarringPwdResp::pCCSUPSType

#### 8.1515.2.5 WORD* voiceSetCallBarringPwdResp::pFailCause

## 8.1516 voiceSetConfigReq Struct Reference

### Data Fields

- BYTE * pAutoAnswer

- airTimer * pAirTimerConfig
- roamTimer * pRoamTimerConfig
- BYTE * pTTYMode
- prefVoiceSO * pPrefVoiceSO
- BYTE * pPrefVoiceDomain

### 8.1516.1 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 airTimer for more information |
| *pRoamTimer-Config* | • Value specified is written to NV_ROAM_CNT_I. (optional)<br>• See 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 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.

### 8.1516.2 Field Documentation

#### 8.1516.2.1 airTimer* voiceSetConfigReq::pAirTimerConfig

#### 8.1516.2.2 BYTE* voiceSetConfigReq::pAutoAnswer

#### 8.1516.2.3 BYTE* voiceSetConfigReq::pPrefVoiceDomain

#### 8.1516.2.4 prefVoiceSO* voiceSetConfigReq::pPrefVoiceSO

#### 8.1516.2.5 roamTimer* voiceSetConfigReq::pRoamTimerConfig

#### 8.1516.2.6 BYTE* voiceSetConfigReq::pTTYMode

## 8.1517 voiceSetConfigResp Struct Reference

### Data Fields

- BYTE * pAutoAnsStatus
- BYTE * pAirTimerStatus
- BYTE * pRoamTimerStatus
- BYTE * pTTYConfigStatus
- BYTE * pPrefVoiceSOStatus
- BYTE * pVoiceDomainPrefStatus

### 8.1517.1 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. |
| *pAirTimerStatus* | • Air Timer Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available. |
| *pRoamTimer-Status* | • Roam Timer Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available. |

| | |
|---|---|
| *pTTYConfig-Status* | • TTY Config Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available. |
| *pPrefVoiceSO-Status* | • Preferred Voice SO Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available. |
| *pVoiceDomain-PrefStatus* | • Voice-Domain Preference Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available. |

**Note**

Parameters which are mentioned as NULL will be ignored.

### 8.1517.2 Field Documentation

#### 8.1517.2.1 BYTE* voiceSetConfigResp::pAirTimerStatus

#### 8.1517.2.2 BYTE* voiceSetConfigResp::pAutoAnsStatus

#### 8.1517.2.3 BYTE* voiceSetConfigResp::pPrefVoiceSOStatus

#### 8.1517.2.4 BYTE* voiceSetConfigResp::pRoamTimerStatus

#### 8.1517.2.5 BYTE* voiceSetConfigResp::pTTYConfigStatus

#### 8.1517.2.6 BYTE* voiceSetConfigResp::pVoiceDomainPrefStatus

## 8.1518 voiceSetPrefPrivacy Struct Reference

### Data Fields

- BYTE privacyPref

### 8.1518.1 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 |

### 8.1518.2 Field Documentation

#### 8.1518.2.1 BYTE voiceSetPrefPrivacy::privacyPref

## 8.1519 voiceSetSUPSServiceReq Struct Reference

### Data Fields

- BYTE voiceSvc
- BYTE reason
- BYTE * pServiceClass
- BYTE * pCallBarringPasswd
- BYTE * pCallForwardingNumber
- BYTE * pTimerVal
- callFwdTypeAndPlan * pCallFwdTypeAndPlan

### 8.1519.1 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 serviceClassInformation 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 callFwdTypeAndPlan for more information |

### 8.1519.2 Field Documentation

#### 8.1519.2.1 BYTE* voiceSetSUPSServiceReq::pCallBarringPasswd

#### 8.1519.2.2 BYTE* voiceSetSUPSServiceReq::pCallForwardingNumber

#### 8.1519.2.3 callFwdTypeAndPlan* voiceSetSUPSServiceReq::pCallFwdTypeAndPlan

#### 8.1519.2.4 BYTE* voiceSetSUPSServiceReq::pServiceClass

#### 8.1519.2.5 BYTE* voiceSetSUPSServiceReq::pTimerVal

#### 8.1519.2.6 BYTE voiceSetSUPSServiceReq::reason

#### 8.1519.2.7 BYTE voiceSetSUPSServiceReq::voiceSvc

## 8.1520 voiceSetSUPSServiceResp Struct Reference

### Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

### 8.1520.1 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 |

| | |
|---|---|
| *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 alphaIDInfo for more information |
| *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 |
| *pCallID* | • Unique call identifier for the dialed call (optional, supply NULL if not required)<br>– 0x00 - if the device does not provide this information |
| *pCCSUPSType* | • Data is present when pCCResultType is present and is other than Voice. (optional, supply NULL if not required)<br>– See ccSUPSType for more information |

### 8.1520.2 Field Documentation

#### 8.1520.2.1 alphaIDInfo* voiceSetSUPSServiceResp::pAlphaIDInfo

#### 8.1520.2.2 BYTE* voiceSetSUPSServiceResp::pCallID

#### 8.1520.2.3 BYTE* voiceSetSUPSServiceResp::pCCResultType

#### 8.1520.2.4 ccSUPSType* voiceSetSUPSServiceResp::pCCSUPSType

#### 8.1520.2.5 WORD* voiceSetSUPSServiceResp::pFailCause

## 8.1521 voiceStopContDTMFinfo Struct Reference

### Data Fields

- BYTE callID

### 8.1521.1 Detailed Description

This structure contains parameters of stop continuous DTMF

**Parameters**

| | |
|---|---|
| *pCallID[IN/OUT]* | • 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 |

### 8.1521.2 Field Documentation

#### 8.1521.2.1 BYTE voiceStopContDTMFinfo::callID

## 8.1522 voiceSUPSInfo Struct Reference

### Data Fields

- SUPSInfo SUPSInformation
- BYTE * pSvcClass
- BYTE * pReason
- BYTE * pCallFWNum
- BYTE * pCallFWTimerVal
- struct USSInfo * pUSSInfo
- BYTE * pCallID
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCallBarPasswd
- newPwdData * pNewPwdData
- BYTE * pDataSrc
- WORD * pFailCause
- getCallFWInfo * pCallFwdInfo
- CLIRResp * pCLIRstatus
- CLIPResp * pCLIPstatus
- COLPResp * pCOLPstatus
- COLRResp * pCOLRstatus
- CNAPResp * pCNAPstatus

### 8.1522.1 Detailed Description

This structure contains the parameters passed for SLQSVoiceSetSUPSCallBack by the device.

**Parameters**

| | |
|---|---|
| *SUPS-Information(mandatory)* | See 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 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 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 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 getCallFWInfo for more information. |
| *pCLI-Rstatus(optional)* | • See CLIRResp for more information. |
| *pCLI-Pstatus(optional)* | • See CLIPResp for more information. |
| *pCOL-Pstatus(optional)* | • See COLPResp for more information. |
| *pCOL-Rstatus(optional)* | • See COLRResp for more information. |

| | |
|---|---|
| *pCNA-Pstatus(optional)* | • See CNAPResp for more information. |

**Note**

None

### 8.1522.2 Field Documentation

#### 8.1522.2.1 alphaIDInfo* voiceSUPSInfo::pAlphaIDInfo

#### 8.1522.2.2 BYTE* voiceSUPSInfo::pCallBarPasswd

#### 8.1522.2.3 getCallFWInfo* voiceSUPSInfo::pCallFwdInfo

#### 8.1522.2.4 BYTE* voiceSUPSInfo::pCallFWNum

#### 8.1522.2.5 BYTE* voiceSUPSInfo::pCallFWTimerVal

#### 8.1522.2.6 BYTE* voiceSUPSInfo::pCallID

#### 8.1522.2.7 CLIPResp* voiceSUPSInfo::pCLIPstatus

#### 8.1522.2.8 CLIRResp* voiceSUPSInfo::pCLIRstatus

#### 8.1522.2.9 CNAPResp* voiceSUPSInfo::pCNAPstatus

#### 8.1522.2.10 COLPResp* voiceSUPSInfo::pCOLPstatus

#### 8.1522.2.11 COLRResp* voiceSUPSInfo::pCOLRstatus

#### 8.1522.2.12 BYTE* voiceSUPSInfo::pDataSrc

#### 8.1522.2.13 WORD* voiceSUPSInfo::pFailCause

#### 8.1522.2.14 newPwdData* voiceSUPSInfo::pNewPwdData

#### 8.1522.2.15 BYTE* voiceSUPSInfo::pReason

#### 8.1522.2.16 BYTE* voiceSUPSInfo::pSvcClass

#### 8.1522.2.17 struct USSInfo* voiceSUPSInfo::pUSSInfo

#### 8.1522.2.18 SUPSInfo voiceSUPSInfo::SUPSInformation

## 8.1523 voiceSUPSNotification Struct Reference

### Data Fields

- BYTE callID
- BYTE notifType
- WORD * pCUGIndex
- ECTNum * pECTNum

### 8.1523.1 Detailed Description

Contains the parameters passed for SLQSVoiceSetSUPSNotificationCallback 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

### 8.1523.2 Field Documentation

#### 8.1523.2.1 BYTE voiceSUPSNotification::callID

#### 8.1523.2.2 BYTE voiceSUPSNotification::notifType

#### 8.1523.2.3 WORD* voiceSUPSNotification::pCUGIndex

#### 8.1523.2.4 ECTNum* voiceSUPSNotification::pECTNum

## 8.1524 wcdmaCellInfo Struct Reference

### Data Fields

- WORD psc
- SHORT cpich_rscp
- SHORT cpich_ecno
- SHORT srxlev

### 8.1524.1 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. |

### 8.1524.2 Field Documentation

#### 8.1524.2.1 SHORT wcdmaCellInfo::cpich_ecno

#### 8.1524.2.2 SHORT wcdmaCellInfo::cpich_rscp

#### 8.1524.2.3 WORD wcdmaCellInfo::psc

#### 8.1524.2.4 SHORT wcdmaCellInfo::srxlev

## 8.1525 WCDMAECIOThresh Struct Reference

### Data Fields

- BYTE WCDMAECIOThreshListLen
- WORD * pWCDMAECIOThreshList

### 8.1525.1 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) |

### 8.1525.2 Field Documentation

#### 8.1525.2.1 WORD* WCDMAECIOThresh::pWCDMAECIOThreshList

#### 8.1525.2.2 BYTE WCDMAECIOThresh::WCDMAECIOThreshListLen

## 8.1526 WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

- ULONG wcdmaRRCState
- BYTE umtsLTENbrCellLen
- umtsLTENbrCell UMTSLTENbrCell [255]

### 8.1526.1 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 umtsLTENbrCell for more information. |

### 8.1526.2 Field Documentation

#### 8.1526.2.1 umtsLTENbrCell WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

#### 8.1526.2.2 BYTE WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

#### 8.1526.2.3 ULONG WCDMAInfoLTENeighborCell::wcdmaRRCState

## 8.1527 wcdmaLongMsgDecodingParams Struct Reference

### Data Fields

- BYTE * pMessage
- BYTE * pSenderAddrLength
- CHAR * pSenderAddr
- BYTE * pTextMsgLength
- CHAR * pTextMsg
- BYTE * pScAddrLength
- CHAR * pScAddr
- BYTE Time [0x09]
- BYTE Date [0x09]
- BYTE * pReferenceNum
- BYTE * pTotalNum
- BYTE * pPartNum
- BOOL * pIsUDHPresent

### 8.1527.1 Detailed Description

Structure contains parameters which need to be decoded from message

**Parameters**

| | |
|---|---|
| *pMessage[IN]* | • Message read off the device via SLQSGetSMS |
| *pSenderAddr-Length[IN/OUT]* | • Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. A length with 24 will be much safe since this address filed can be up to 12 octets (24 bytes) Upon successful output, returns the length of destination address string. |
| *pSenderAddr[O-UT]* | • Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing destination address This field contains the Type of Address. To get the exact address, skip the first two bytes. e.g, 9085290100334, 90 is the Type of Address, indicates international format of phone number |
| *pTextMsg-Length[IN/OUT]* | • Upon input, specifies the number of characters the given text message buffer can accommodate. Upon successful output, returns the number of characters returns in the given text message buffer. |
| *pTextMsg[OUT]* | • Encoded PDU message |
| *pScAddrLength[-IN/OUT]* | • A length with 24 will be much safe since this address filed can be up to 12 octets (24 bytes) Returns NULL-terminated ASCII String containing destination address |
| *pScAddr[OUT]* | • Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing service center address. This SMSC field contains the Type of Address. To get the exact SMSC address, skip the first two bytes. e.g, 9085290100334, 90 is the Type of Address, indicates international format of phone number |
| *pTime[OUT]* | • Time fecthed from message |
| *pReference-Num[OUT]* | • Reference number of the sms |
| *pTotalNum[OUT]* | • Total number of the concatenated message |
| *pPartNum[OUT]* | • Sequence number of the current message |
| *pIsUDHPresent* | • Is User Data Header Prensent in the PDU? If yes, it means it is a<br>• concatenated SMS. |

## 8.1527.2 Field Documentation

### 8.1527.2.1 BYTE wcdmaLongMsgDecodingParams::Date[0x09]

#### 8.1527.2.2 BOOL* wcdmaLongMsgDecodingParams::pIsUDHPresent

#### 8.1527.2.3 BYTE* wcdmaLongMsgDecodingParams::pMessage

#### 8.1527.2.4 BYTE* wcdmaLongMsgDecodingParams::pPartNum

#### 8.1527.2.5 BYTE* wcdmaLongMsgDecodingParams::pReferenceNum

#### 8.1527.2.6 CHAR* wcdmaLongMsgDecodingParams::pScAddr

#### 8.1527.2.7 BYTE* wcdmaLongMsgDecodingParams::pScAddrLength

#### 8.1527.2.8 CHAR* wcdmaLongMsgDecodingParams::pSenderAddr

#### 8.1527.2.9 BYTE* wcdmaLongMsgDecodingParams::pSenderAddrLength

#### 8.1527.2.10 CHAR* wcdmaLongMsgDecodingParams::pTextMsg

#### 8.1527.2.11 BYTE* wcdmaLongMsgDecodingParams::pTextMsgLength

#### 8.1527.2.12 BYTE* wcdmaLongMsgDecodingParams::pTotalNum

#### 8.1527.2.13 BYTE wcdmaLongMsgDecodingParams::Time[0x09]

## 8.1528 wcdmaMsgDecodingParams Struct Reference

### Data Fields

- BYTE * pMessage
- BYTE * pSenderAddrLength
- CHAR * pSenderAddr
- BYTE * pTextMsgLength
- CHAR * pTextMsg
- BYTE * pScAddrLength
- CHAR * pScAddr
- BYTE Time [0x09]
- BYTE Date [0x09]

### 8.1528.1 Detailed Description

Structure contains parameters which need to be decoded from message

**Parameters**

| | |
|---|---|
| *pMessage[IN]* | • Message read off the device via SLQSGetSMS |
| *pSenderAddr-Length[IN/OUT]* | • Upon input, indicates the maximum number of ASCII characters (including NULL termination) that the pSenderAddr buffer can accommodate. A length with 24 will be much safe since this address filed can be up to 12 octets (24 bytes) Upon successful output, returns the length of destination address string. |

| | |
|---|---|
| *pSenderAddr[OUT]* | • Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing destination address |
| *pTextMsgLength[IN/OUT]* | • Upon input, specifies the number of characters the given text message buffer can accommodate. Upon successful output, returns the number of characters returns in the given text message buffer. |
| *pTextMsg[OUT]* | • Encoded PDU message |
| *pScAddrLength[IN/OUT]* | • A length with 24 will be much safe since this address filed can be up to 12 octets (24 bytes) Returns NULL-terminated ASCII String containing destination address |
| *pScAddr[OUT]* | • Note that a length with 24 bytes will be much safe. Returns NULL-terminated ASCII String containing service center address. This SMSC field contains the Type of Address. To get the exact SMSC address, skip the first two bytes. e.g, 9085290100334, 90 is the Type of Address, indicates international format of phone number |
| *pTime[OUT]* | • Time fecthed from message |
| *pDate* | • Date fecthed from message |

### 8.1528.2 Field Documentation

#### 8.1528.2.1 BYTE wcdmaMsgDecodingParams::Date[0x09]

#### 8.1528.2.2 BYTE* wcdmaMsgDecodingParams::pMessage

#### 8.1528.2.3 CHAR* wcdmaMsgDecodingParams::pScAddr

#### 8.1528.2.4 BYTE* wcdmaMsgDecodingParams::pScAddrLength

#### 8.1528.2.5 CHAR* wcdmaMsgDecodingParams::pSenderAddr

#### 8.1528.2.6 BYTE* wcdmaMsgDecodingParams::pSenderAddrLength

#### 8.1528.2.7 CHAR* wcdmaMsgDecodingParams::pTextMsg

#### 8.1528.2.8 BYTE* wcdmaMsgDecodingParams::pTextMsgLength

#### 8.1528.2.9 BYTE wcdmaMsgDecodingParams::Time[0x09]

## 8.1529 wcdmaMsgEncodingParams Struct Reference

### Data Fields

- ULONG messageSize
- CHAR * pDestAddr
- CHAR * pTextMsg
- CHAR * pPDUMessage
- BYTE alphabet

#### 8.1529.1 Detailed Description

Structure contains parameters which need to encoded with message

**Parameters**

| | |
|---|---|
| *messageSize* | • The length of the message contents in bytes |
| *pDestAddr[IN]* | • Gives NULL-terminated ASCII String containing destination address |
| *pTextMsg[IN]* | • Text message to be encoded, maximum limit is 160 charaters |
| *pPDUMessage[-OUT]* | • Encoded PDU message |
| *alphabet[IN]* | • Encoding method to generate the PDU<br>– 0 - 7 bit encoding<br>– 4 - 8 bit encoding<br>– 8 - 16 bit UCS2 encoding<br>– others value will be treated as default 7 bit encoding |

#### 8.1529.2 Field Documentation

##### 8.1529.2.1 BYTE wcdmaMsgEncodingParams::alphabet

##### 8.1529.2.2 ULONG wcdmaMsgEncodingParams::messageSize

##### 8.1529.2.3 CHAR* wcdmaMsgEncodingParams::pDestAddr

##### 8.1529.2.4 CHAR* wcdmaMsgEncodingParams::pPDUMessage

##### 8.1529.2.5 CHAR* wcdmaMsgEncodingParams::pTextMsg

## 8.1530 WCDMARSSIThresh Struct Reference

### Data Fields

- BYTE WCDMARSSIThreshListLen
- WORD * pWCDMARSSIThreshList

### 8.1530.1 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) |

### 8.1530.2 Field Documentation

#### 8.1530.2.1 WORD* WCDMARSSIThresh::pWCDMARSSIThreshList

#### 8.1530.2.2 BYTE WCDMARSSIThresh::WCDMARSSIThreshListLen

## 8.1531 WCDMASysInfo Struct Reference

### Data Fields

- sysInfoCommon sysInfoWCDMA
- BYTE lacValid
- WORD lac
- BYTE cellIdValid
- ULONG cellId
- BYTE regRejectInfoValid
- BYTE rejectSrvDomain
- BYTE rejCause
- BYTE networkIdValid
- BYTE MCC [3]
- BYTE MNC [3]
- BYTE hsCallStatusValid
- BYTE hsCallStatus
- BYTE hsIndValid
- BYTE hsInd
- BYTE pscValid
- WORD psc

### 8.1531.1 Detailed Description

Structure for storing the WCDMA System Information.

**Parameters**

| | |
|---|---|
| *sysInfoWCDMA* | • See 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[PLMN_LE-NGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters |

| | |
|---|---|
| *MNC[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. |
| *hsCallStatusValid* | • 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 |

### 8.1531.2 Field Documentation

#### 8.1531.2.1 ULONG WCDMASysInfo::cellId

#### 8.1531.2.2 BYTE WCDMASysInfo::cellIdValid

#### 8.1531.2.3 BYTE WCDMASysInfo::hsCallStatus

#### 8.1531.2.4 BYTE WCDMASysInfo::hsCallStatusValid

#### 8.1531.2.5 BYTE WCDMASysInfo::hsInd

#### 8.1531.2.6 BYTE WCDMASysInfo::hsIndValid

#### 8.1531.2.7 WORD WCDMASysInfo::lac

#### 8.1531.2.8 BYTE WCDMASysInfo::lacValid

#### 8.1531.2.9 BYTE WCDMASysInfo::MCC[3]

#### 8.1531.2.10 BYTE WCDMASysInfo::MNC[3]

#### 8.1531.2.11 BYTE WCDMASysInfo::networkIdValid

#### 8.1531.2.12 WORD WCDMASysInfo::psc

#### 8.1531.2.13 BYTE WCDMASysInfo::pscValid

#### 8.1531.2.14 BYTE WCDMASysInfo::regRejectInfoValid

#### 8.1531.2.15 BYTE WCDMASysInfo::rejCause

#### 8.1531.2.16 BYTE WCDMASysInfo::rejectSrvDomain

#### 8.1531.2.17 sysInfoCommon WCDMASysInfo::sysInfoWCDMA

## 8.1532 wcdmaUARFCN Struct Reference

### Data Fields

- BYTE status
- ULONG uarfcn

### 8.1532.1 Detailed Description

This structure contains the parameters for WCDMA UARFCN.

**Parameters**

| | |
|---|---|
| *status* | • 0 - Disable<br>• 1 - Enable |
| *uarfcn* | • UARFCN to which UE is locked |

### 8.1532.2 Field Documentation

#### 8.1532.2.1 BYTE wcdmaUARFCN::status

#### 8.1532.2.2 ULONG wcdmaUARFCN::uarfcn

## 8.1533 wds_channelRate Struct Reference

### Data Fields

- uint32_t CurrChanTxRate
- uint32_t CurrChanRxRate

### 8.1533.1 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 |

### 8.1533.2 Field Documentation

#### 8.1533.2.1 uint32_t wds_channelRate::CurrChanRxRate

#### 8.1533.2.2 uint32_t wds_channelRate::CurrChanTxRate

## 8.1534 wds_ChannelRateTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- wds_channelRate ChannelRate

### 8.1534.1 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 |

### 8.1534.2 Field Documentation

#### 8.1534.2.1 wds_channelRate wds_ChannelRateTlv::ChannelRate

#### 8.1534.2.2 uint8_t wds_ChannelRateTlv::TlvPresent

## 8.1535 wds_ConnStatusTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t MDMConnStatus

### 8.1535.1 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 |

### 8.1535.2 Field Documentation

#### 8.1535.2.1 uint8_t wds_ConnStatusTlv::MDMConnStatus

#### 8.1535.2.2 uint8_t wds_ConnStatusTlv::TlvPresent

## 8.1536 wds_currNetworkInfo Struct Reference

### Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

### 8.1536.1 Detailed Description

Network information structure

**Parameters**

| | |
|---|---|
| *NetworkType* | • Values:<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *RATMask* | • Radio Access Technology (RAT) mask to indicate the type of technology.<br>• Values:<br>– 0 - Don't Care<br>– 0x8000 - NULL Bearer<br>• CDMA RAT mask values:<br>– 0x01 - CDMA_1x<br>– 0x02 - EVDO_REV0<br>– 0x05 - HRPD<br>– 0x0B - EHRPD<br>• UMTS RAT mask values:<br>– 0x03 - GPRS<br>– 0x04 - WCDMA<br>– 0x06 - EDGE<br>– 0x07 - HSDPA and WCDMA<br>– 0x08 - HSUPA and WCDMA<br>– 0x08 - 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* | • Service Option (SO) mask to indicate the service option or type of application.<br>• Values:<br>– 0 - Don't Care<br>• CDMA 1x SO mask values:<br>– 0x01 - CDMA_1X_IS95<br>– 0x02 - CDMA_1X_IS2000<br>– 0x04 - CDMA_1X_IS2000_REL_A<br>• CDMA EV-DO Rev 0 SO mask values:<br>– 0x01 - DPA<br>• CDMA EV-DO Rev A SO mask values:<br>– 0x01 - DPA<br>– 0x02 - MFPA<br>– 0x04 - EMPA<br>– 0x08 - EMPA_EHRPD<br>• CDMA EV-DO Rev B SO mask values:<br>– 0x01 - DPA<br>– 0x02 - MFPA<br>– 0x04 - EMPA<br>– 0x08 - EMPA_EHRPD<br>– 0x10 - MMPA<br>– 0x20 - MMPA_EHRPD |

### 8.1536.2 Field Documentation

#### 8.1536.2.1 uint8_t wds_currNetworkInfo::NetworkType

#### 8.1536.2.2 uint32_t wds_currNetworkInfo::RATMask

#### 8.1536.2.3 uint32_t wds_currNetworkInfo::SOMask

## 8.1537 wds_DataBearTechTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t DataBearerTech

### 8.1537.1 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 |

### 8.1537.2 Field Documentation

#### 8.1537.2.1 uint8_t wds_DataBearerTechTlv::DataBearerTech

#### 8.1537.2.2 uint8_t wds_DataBearerTechTlv::TlvPresent

## 8.1538 wds_DataULongLongTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t ullData

### 8.1538.1 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 |

### 8.1538.2 Field Documentation

#### 8.1538.2.1 uint8_t wds_DataULongLongTlv::TlvPresent

#### 8.1538.2.2 uint64_t wds_DataULongLongTlv::ullData

## 8.1539 wds_DataULongTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t ulData

### 8.1539.1 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 |

### 8.1539.2 Field Documentation

#### 8.1539.2.1 uint8_t wds_DataULongTlv::TlvPresent

#### 8.1539.2.2 uint32_t wds_DataULongTlv::ulData

## 8.1540 wds_DHCPLeaseOptTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t numOpt
- wds_DHCPOpt optList [30]
- uint8_t optListData [2048]

### 8.1540.1 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 |

### 8.1540.2 Field Documentation

#### 8.1540.2.1 uint8_t wds_DHCPLeaseOptTlv::numOpt

#### 8.1540.2.2 wds_DHCPOpt wds_DHCPLeaseOptTlv::optList[30]

#### 8.1540.2.3 uint8_t wds_DHCPLeaseOptTlv::optListData[2048]

#### 8.1540.2.4 uint8_t wds_DHCPLeaseOptTlv::TlvPresent

## 8.1541 wds_DHCPLeaseStateTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t leaseState

### 8.1541.1 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 |

### 8.1541.2 Field Documentation

#### 8.1541.2.1 uint8_t wds_DHCPLeaseStateTlv::leaseState

#### 8.1541.2.2 uint8_t wds_DHCPLeaseStateTlv::TlvPresent

## 8.1542 wds_DHCPOpt Struct Reference

### Data Fields

- uint8_t optCode
- uint8_t optValLen
- uint8_t * pOptVal

### 8.1542.1 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 |

### 8.1542.2 Field Documentation

#### 8.1542.2.1 uint8_t wds_DHCPOpt::optCode

#### 8.1542.2.2 uint8_t wds_DHCPOpt::optValLen

#### 8.1542.2.3 uint8_t* wds_DHCPOpt::pOptVal

## 8.1543 wds_DHCPProfileIdTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t profileType
- uint8_t profileId

### 8.1543.1 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) |

### 8.1543.2 Field Documentation

#### 8.1543.2.1 uint8_t wds_DHCPProfileIdTlv::profileId

#### 8.1543.2.2 uint8_t wds_DHCPProfileIdTlv::profileType

#### 8.1543.2.3 uint8_t wds_DHCPProfileIdTlv::TlvPresent

## 8.1544 wds_DHCPv4HWConfig Struct Reference

### Data Fields

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

### 8.1544.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

**Parameters**

| | |
|---|---|
| *hwType* | • HW Type 1 - Ethernet 20 - Serial |
| *chaddrlen* | • chaddrlen |
| *chaddr* | • chaddr. Max size 16 bytes |

### 8.1544.2 Field Documentation

#### 8.1544.2.1 uint8_t wds_DHCPv4HWConfig::chaddr[16]

#### 8.1544.2.2 uint8_t wds_DHCPv4HWConfig::chaddrLen

#### 8.1544.2.3 uint8_t wds_DHCPv4HWConfig::hwType

## 8.1545 wds_DHCPv4Option Struct Reference

### Data Fields

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

### 8.1545.1 Detailed Description

WDS SWI DHCPv4 Option Structure

**Parameters**

| | |
|---|---|
| *optCode* | • Option code<br>– 0 - 255 |
| *optValLen* | • Option value length<br>– 0 - 255 |
| *optVal* | • Option value |

### 8.1545.2 Field Documentation

#### 8.1545.2.1 uint8_t wds_DHCPv4Option::optCode

#### 8.1545.2.2 uint8_t wds_DHCPv4Option::optVal[255]

#### 8.1545.2.3 uint8_t wds_DHCPv4Option::optValLen

## 8.1546 wds_DHCPv4OptionList Struct Reference

### Data Fields

- uint8_t numOpt
- wds_DHCPv4Option * pOptList

### 8.1546.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

**Parameters**

| | |
|---|---|
| *numOpt* | • number of options<br>– 0 - 255 |
| *pOptList* | • pointer to list of DHCP Options |

### 8.1546.2 Field Documentation

#### 8.1546.2.1 uint8_t wds_DHCPv4OptionList::numOpt

#### 8.1546.2.2 wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList

## 8.1547 wds_DHCPv4ProfileId Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileId

### 8.1547.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

**Parameters**

| | |
|---|---|
| *profileType* | • 0 for 3GPP |
| *profileId* | • 1 to 24 for 3GPP profile |

### 8.1547.2 Field Documentation

#### 8.1547.2.1 uint8_t wds_DHCPv4ProfileId::profileId

#### 8.1547.2.2 uint8_t wds_DHCPv4ProfileId::profileType

## 8.1548 wds_Domain Struct Reference

### Data Fields

- uint16_t domainLen
- uint8_t domainName [256]

### 8.1548.1 Detailed Description

This structure contains the DomainName Information

**Parameters**

| | |
|---|---|
| *domainLen* | • length of the recieved Domain name |
| *domainName* | • Domain name(Max 256 characters) |

### 8.1548.2 Field Documentation

#### 8.1548.2.1 uint16_t wds_Domain::domainLen

#### 8.1548.2.2 uint8_t wds_Domain::domainName[256]

## 8.1549 wds_DomainNameList Struct Reference

### Data Fields

- uint8_t numInstances
- struct wds_Domain domain [10]

### 8.1549.1 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.

### 8.1549.2 Field Documentation

#### 8.1549.2.1 struct wds_Domain wds_DomainNameList::domain[10]

#### 8.1549.2.2 uint8_t wds_DomainNameList::numInstances

## 8.1550 wds_DormStatTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t DormancyStat

### 8.1550.1 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 |

### 8.1550.2 Field Documentation

#### 8.1550.2.1 uint8_t wds_DormStatTlv::DormancyStat

#### 8.1550.2.2 uint8_t wds_DormStatTlv::TlvPresent

## 8.1551 wds_GPRSQoS Struct Reference

### Data Fields

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

### 8.1551.1 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.

### 8.1551.2 Field Documentation

#### 8.1551.2.1 uint32_t wds_GPRSQoS::delayClass

#### 8.1551.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

#### 8.1551.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

#### 8.1551.2.4 uint32_t wds_GPRSQoS::precedenceClass

#### 8.1551.2.5 uint32_t wds_GPRSQoS::reliabilityClass

## 8.1552 wds_IPv4AdTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t IPv4Addr

### 8.1552.1 Detailed Description

IPv4 address TLV information

**Parameters**

| | |
|---|---|
| *TlvPresent* | a flag indicates whether TLV presented or not |
| *IPv4Addr* | IPv4 address |

### 8.1552.2 Field Documentation

#### 8.1552.2.1 uint32_t wds_IPv4AdTlv::IPv4Addr

#### 8.1552.2.2 uint8_t wds_IPv4AdTlv::TlvPresent

## 8.1553 wds_IPV6AddressInfo Struct Reference

### Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

### 8.1553.1 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. |

### 8.1553.2 Field Documentation

#### 8.1553.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

#### 8.1553.2.2 uint8_t wds_IPV6AddressInfo::IPV6PrefixLen

## 8.1554 wds_IPV6GWAddressInfo Struct Reference

### Data Fields

- uint8_t gwV6PrefixLen
- uint16_t gwAddressV6 [8]

### 8.1554.1 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.

### 8.1554.2 Field Documentation

#### 8.1554.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

#### 8.1554.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

## 8.1555 wds_LastMdmCallEndRsnTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t CallEndReason

### 8.1555.1 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 |

### 8.1555.2 Field Documentation

#### 8.1555.2.1 uint16_t wds_LastMdmCallEndRsnTlv::CallEndReason

#### 8.1555.2.2 uint8_t wds_LastMdmCallEndRsnTlv::TlvPresent

## 8.1556 wds_PCSCFFQDNAddress Struct Reference

### Data Fields

- uint16_t fqdnLen
- uint8_t fqdnAddr [256]

### 8.1556.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

**Parameters**

| | |
|---|---|
| *fqdnLen* | • length of the received FQDN address |
| *fqdnAddr* | • FQDN address(Max 256 characters) |

### 8.1556.2 Field Documentation

#### 8.1556.2.1 uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]

#### 8.1556.2.2 uint16_t wds_PCSCFFQDNAddress::fqdnLen

## 8.1557 wds_PCSCFFQDNAddressList Struct Reference

### Data Fields

- uint8_t numInstances
- struct wds_PCSCFFQDNAddress pcsfFQDNAddress [10]

### 8.1557.1 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.

### 8.1557.2 Field Documentation

#### 8.1557.2.1 uint8_t wds_PCSCFFQDNAddressList::numInstances

#### 8.1557.2.2 struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]

## 8.1558 wds_PCSCFIPv4ServerAddressList Struct Reference

### Data Fields

- uint8_t numInstances
- uint32_t pscsfIPv4Addr [64]

### 8.1558.1 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.

### 8.1558.2 Field Documentation

#### 8.1558.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

#### 8.1558.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

## 8.1559 wds_ProfileIdentifier Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileIndex

### 8.1559.1 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.

### 8.1559.2 Field Documentation

#### 8.1559.2.1 uint8_t wds_ProfileIdentifier::profileIndex

#### 8.1559.2.2 uint8_t wds_ProfileIdentifier::profileType

## 8.1560 wds_profileInfo Union Reference

### Data Fields

- LibPackprofile_3GPP SlqsProfile3GPP
- LibPackprofile_3GPP2 SlqsProfile3GPP2

### 8.1560.1 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 |

### 8.1560.2 Field Documentation

#### 8.1560.2.1 LibPackprofile_3GPP wds_profileInfo::SlqsProfile3GPP

#### 8.1560.2.2 LibPackprofile_3GPP2 wds_profileInfo::SlqsProfile3GPP2

## 8.1561 wds_RXBytesOKTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t RxByteOKCnt

### 8.1561.1 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 |

### 8.1561.2 Field Documentation

#### 8.1561.2.1 uint64_t wds_RXBytesOKTlv::RxByteOKCnt

#### 8.1561.2.2 uint8_t wds_RXBytesOKTlv::TlvPresent

## 8.1562 wds_transferStatInd Struct Reference

### Data Fields

- uint8_t StatsPeriod
- uint32_t StatsMask

### 8.1562.1 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 |

### 8.1562.2 Field Documentation

#### 8.1562.2.1 uint32_t wds_transferStatInd::StatsMask

#### 8.1562.2.2 uint8_t wds_transferStatInd::StatsPeriod

## 8.1563 wds_TrStatInd Struct Reference

### Data Fields

- uint8_t statsPeriod
- uint32_t statsMask

### 8.1563.1 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 SLQSWdsEventReportCallBack.<br>• All unlisted bits are reserved for future use and must be set to zero. |

### 8.1563.2 Field Documentation

#### 8.1563.2.1 uint32_t wds_TrStatInd::statsMask

#### 8.1563.2.2 uint8_t wds_TrStatInd::statsPeriod

## 8.1564 wds_TXBytesOKTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t TxByteOKCnt

### 8.1564.1 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 |

### 8.1564.2 Field Documentation

#### 8.1564.2.1 uint8_t wds_TXBytesOKTlv::TlvPresent

#### 8.1564.2.2 uint64_t wds_TXBytesOKTlv::TxByteOKCnt

## 8.1565 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

### 8.1565.1 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. |

### 8.1565.2 Field Documentation

#### 8.1565.2.1 uint8_t wds_UMTSMinQoS::deliveryErrSDU

#### 8.1565.2.2 uint32_t wds_UMTSMinQoS::grntDownlinkBitrate

#### 8.1565.2.3 uint32_t wds_UMTSMinQoS::grntUplinkBitrate

#### 8.1565.2.4 uint32_t wds_UMTSMinQoS::maxDownlinkBitrate

#### 8.1565.2.5 uint32_t wds_UMTSMinQoS::maxSDUSize

#### 8.1565.2.6 uint32_t wds_UMTSMinQoS::maxUplinkBitrate

#### 8.1565.2.7 uint8_t wds_UMTSMinQoS::qosDeliveryOrder

#### 8.1565.2.8 uint8_t wds_UMTSMinQoS::resBerRatio

#### 8.1565.2.9 uint8_t wds_UMTSMinQoS::sduErrorRatio

#### 8.1565.2.10 uint8_t wds_UMTSMinQoS::trafficClass

#### 8.1565.2.11 uint32_t wds_UMTSMinQoS::trafficPriority

#### 8.1565.2.12 uint32_t wds_UMTSMinQoS::transferDelay

## 8.1566 WdsByteTotals Struct Reference

### Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsByteTotalsElmnts ByteTotalsElmntsV4
- struct WdsByteTotalsElmnts ByteTotalsElmntsV6

### 8.1566.1 Detailed Description

WDS ByteTotals request data structure

**Parameters**

| | |
|---|---|
| *pV4sessionId* | • The v4 session ID for which the byte totals are to be retrieved<br>• provide a NULL pointer if not applicable |
| *pV6sessionId* | • The v6 session ID for which the byte totals are to be retrieved<br>• provide a NULL pointer if not applicable |
| *ByteTotals-ElmntsV4* | • data structure to be populated with the byte totals for V4 session |
| *ByteTotals-ElmntsV6* | • data structure to be populated with the byte totals for V6 session |

**Note**

At least one of pV4sessionId and pV6sessionId must point to a valid session ID.

### 8.1566.2 Field Documentation

#### 8.1566.2.1 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV4

#### 8.1566.2.2 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV6

#### 8.1566.2.3 ULONG* WdsByteTotals::pV4sessionId

#### 8.1566.2.4 ULONG* WdsByteTotals::pV6sessionId

## 8.1567 WdsByteTotalsElmnts Struct Reference

### Data Fields

- ULONGLONG * pTXTotalBytes
- ULONGLONG * pRXTotalBytes

### 8.1567.1 Detailed Description

WDS Bytes Totals request data structure for individual session

**Parameters**

| | |
|---|---|
| *pTXTotalBytes* | • No of transmitted bytes without error. |
| *pRXTotalBytes* | • No of received bytes without error. |

### 8.1567.2 Field Documentation

#### 8.1567.2.1 ULONGLONG* WdsByteTotalsElmnts::pRXTotalBytes

#### 8.1567.2.2 ULONGLONG* WdsByteTotalsElmnts::pTXTotalBytes

## 8.1568 WdsClientLeaseChange Struct Reference

### Data Fields

- BYTE * pEnableNotification

### 8.1568.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

**Parameters**

| | |
|---|---|
| *pEnable-Notification* | [IN]<br>• Enable Notification or not |

### 8.1568.2 Field Documentation

#### 8.1568.2.1 BYTE* WdsClientLeaseChange::pEnableNotification

## 8.1569 WdsConnectionRate Struct Reference

### Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsConnectionRateElmnts ConnRateElmntsV4
- struct WdsConnectionRateElmnts ConnRateElmntsV6

### 8.1569.1 Detailed Description

WDS ConnectionRate request data structure

**Parameters**

| | |
|---|---|
| *pV4sessionId* | • The v4 session ID for which the connection rate are to be retrieved<br>• provide a NULL pointer if not applicable |

| | |
|---|---|
| *pV6sessionId* | • The v6 session ID for which the connection rate are to be retrieved<br>• provide a NULL pointer if not applicable |
| *ConnRate-ElmntsV4* | • data structure to be populated with the connection rate for V4 session |
| *ConnRate-ElmntsV6* | • data structure to be populated with the connection rate for V6 session |

**Note**

At least one of pV4sessionId and pV6sessionId must point to a valid session ID.

### 8.1569.2 Field Documentation

#### 8.1569.2.1 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV4

#### 8.1569.2.2 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV6

#### 8.1569.2.3 ULONG* WdsConnectionRate::pV4sessionId

#### 8.1569.2.4 ULONG* WdsConnectionRate::pV6sessionId

## 8.1570 WdsConnectionRateElmnts Struct Reference

### Data Fields

- ULONG * pCurrentChannelTXRate
- ULONG * pCurrentChannelRXRate
- ULONG * pMaxChannelTXRate
- ULONG * pMaxChannelRXRate

### 8.1570.1 Detailed Description

WDS Connection rates request data structure for individual session

**Parameters**

| | |
|---|---|
| *pCurrent-ChannelTX-Rate[OUT]* | • Instantaneous channel Tx rate in bits per second. |
| *pCurrent-ChannelRX-Rate[OUT]* | • Instantaneous channel Rx rate in bits per second. |
| *pMaxChannelT-XRate[OUT]* | • Maximum Tx rate that can be assigned to the device by the<br>• serving system in bits per second |
| *pMaxChannelR-XRate[OUT]* | • Maximum Rx rate that can be assigned to the device by the<br>• serving system in bits per second |

### 8.1570.2 Field Documentation

#### 8.1570.2.1 ULONG* WdsConnectionRateElmnts::pCurrentChannelRXRate

#### 8.1570.2.2 ULONG* WdsConnectionRateElmnts::pCurrentChannelTXRate

#### 8.1570.2.3 ULONG* WdsConnectionRateElmnts::pMaxChannelRXRate

#### 8.1570.2.4 ULONG* WdsConnectionRateElmnts::pMaxChannelTXRate

## 8.1571 WdsDHCPv4ClientLeaseInd Struct Reference

### Data Fields

- WdsDHCPv4ProfileId * pProfileId
- BYTE * pLeaseState
- ULONG * pIPv4Addr
- DHCPOptionList * pOptList

### 8.1571.1 Detailed Description

This structure contains DHCPv4 client lease status

**Parameters**

| | |
|---|---|
| *pProfileId* | • Profile Type and Id |
| *pLeaseState* | • Values<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 |
| *pIPv4Addr* | • Values<br>– IPv4 Address |
| *pOptList* | • Option list |

### 8.1571.2 Field Documentation

#### 8.1571.2.1 ULONG* WdsDHCPv4ClientLeaseInd::pIPv4Addr

#### 8.1571.2.2 BYTE* WdsDHCPv4ClientLeaseInd::pLeaseState

#### 8.1571.2.3 DHCPOptionList* WdsDHCPv4ClientLeaseInd::pOptList

#### 8.1571.2.4 WdsDHCPv4ProfileId* WdsDHCPv4ClientLeaseInd::pProfileId

## 8.1572 WdsDHCPv4Config Struct Reference

### Data Fields

- WdsDHCPv4ProfileId * pProfileId
- WdsDHCPv4HWConfig * pHwConfig
- WdsDHCPv4OptionList * pRequestOptionList

### 8.1572.1 Detailed Description

WDS SWI DHCPv4 Config Structure

**Parameters**

| | |
|---|---|
| *pProfileId* | [IN]<br>• pointer to Profile Id structure |
| *pHWConfig* | [IN/OUT]<br>• pointer to HW Config structure |
| *pRequestOption-List* | [IN/OUT]<br>• pointer to Option List structure to be sent in DHCP request |

### 8.1572.2 Field Documentation

#### 8.1572.2.1 WdsDHCPv4HWConfig* WdsDHCPv4Config::pHwConfig

#### 8.1572.2.2 WdsDHCPv4ProfileId* WdsDHCPv4Config::pProfileId

#### 8.1572.2.3 WdsDHCPv4OptionList* WdsDHCPv4Config::pRequestOptionList

## 8.1573 wdsDhcpv4HwConfig Struct Reference

### Data Fields

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

### 8.1573.1 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 |

### 8.1573.2 Field Documentation

#### 8.1573.2.1 uint8_t wdsDhcpv4HwConfig::chaddr[16]

#### 8.1573.2.2 uint8_t wdsDhcpv4HwConfig::chaddrLen

#### 8.1573.2.3 uint8_t wdsDhcpv4HwConfig::hwType

## 8.1574 WdsDHCPv4HWConfig Struct Reference

### Data Fields

- BYTE hwType
- BYTE chaddrLen
- BYTE chaddr [16]

### 8.1574.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

**Parameters**

| | |
|---|---|
| *hwType* | • HW Type 1 - Ethernet 20 - Serial |
| *chaddrlen* | • chaddrlen |
| *chaddr* | • chaddr. Max size 16 bytes |

### 8.1574.2 Field Documentation

#### 8.1574.2.1 BYTE WdsDHCPv4HWConfig::chaddr[16]

#### 8.1574.2.2 BYTE WdsDHCPv4HWConfig::chaddrLen

#### 8.1574.2.3 BYTE WdsDHCPv4HWConfig::hwType

## 8.1575 WdsDHCPv4Option Struct Reference

### Data Fields

- BYTE optCode
- BYTE optValLen
- BYTE optVal [255]

### 8.1575.1 Detailed Description

WDS SWI DHCPv4 Option Structure

**Parameters**

| | |
|---|---|
| *optCode* | • Option code<br>– 0 - 255 |
| *optValLen* | • Option value length<br>– 0 - 255 |
| *optVal* | • Option value |

### 8.1575.2 Field Documentation

#### 8.1575.2.1 BYTE WdsDHCPv4Option::optCode

#### 8.1575.2.2 BYTE WdsDHCPv4Option::optVal[255]

#### 8.1575.2.3 BYTE WdsDHCPv4Option::optValLen

## 8.1576 wdsDhcpv4Option Struct Reference

### Data Fields

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

### 8.1576.1 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 |

### 8.1576.2 Field Documentation

#### 8.1576.2.1 uint8_t wdsDhcpv4Option::optCode

#### 8.1576.2.2 uint8_t wdsDhcpv4Option::optVal[255]

#### 8.1576.2.3 uint8_t wdsDhcpv4Option::optValLen

## 8.1577 WdsDHCPv4OptionList Struct Reference

### Data Fields

- BYTE numOpt
- WdsDHCPv4Option * pOptList

### 8.1577.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

**Parameters**

| | |
|---|---|
| *numOpt* | • number of options<br>– 0 - 255 |
| *pOptList* | • pointer to list of DHCP Options |

### 8.1577.2 Field Documentation

#### 8.1577.2.1 BYTE WdsDHCPv4OptionList::numOpt

#### 8.1577.2.2 WdsDHCPv4Option* WdsDHCPv4OptionList::pOptList

## 8.1578 wdsDhcpv4OptionList Struct Reference

### Data Fields

- uint8_t numOpt
- wdsDhcpv4Option * pOptList

### 8.1578.1 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 |

### 8.1578.2 Field Documentation

#### 8.1578.2.1 uint8_t wdsDhcpv4OptionList::numOpt

#### 8.1578.2.2 wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList

## 8.1579 wdsDhcpv4ProfileId Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileId

### 8.1579.1 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) |

### 8.1579.2 Field Documentation

#### 8.1579.2.1 uint8_t wdsDhcpv4ProfileId::profileId

#### 8.1579.2.2 uint8_t wdsDhcpv4ProfileId::profileType

## 8.1580 WdsDHCPv4ProfileId Struct Reference

### Data Fields

- BYTE profileType
- BYTE profileId

### 8.1580.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

**Parameters**

| | |
|---|---|
| *profileType* | • 0 for 3GPP |
| *profileId* | • 1 to 24 for 3GPP profile |

### 8.1580.2 Field Documentation

#### 8.1580.2.1 BYTE WdsDHCPv4ProfileId::profileId

#### 8.1580.2.2 BYTE WdsDHCPv4ProfileId::profileType

## 8.1581 WDSGetLoopbackData Struct Reference

### Data Fields

- BYTE ByteLoopbackMode
- BYTE ByteLoopbackMultiplier

### 8.1581.1 Detailed Description

This API to Queries Enable/disable Data Loopback Mode and set the value of loopback multiplier.

**Parameters**

| | |
|---|---|
| *pReq* | [IN]<br>• See WDSSetLoopbackData for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

WDS SWI Get Loopback Structure of Packet Data Connection Information.

**Parameters**

| | |
|---|---|
| *ByteLoopback-Mode* | • Loopback Mode.<br>– 0 - Disable<br>– 1 - Enable |
| *ByteLoopback-Multiplier* | • Loopback multiplier. Number of downlink bytes to send for each uplink byte. |

### 8.1581.2 Field Documentation

#### 8.1581.2.1 BYTE WDSGetLoopbackData::ByteLoopbackMode

#### 8.1581.2.2 BYTE WDSGetLoopbackData::ByteLoopbackMultiplier

## 8.1582 WdsIpAddressInfoReq Struct Reference

### Data Fields

- ULONG * pv4sessionId
- ULONG * pv6sessionId
- QmiWdsIpAddressInfo ip

### 8.1582.1 Detailed Description

**Parameters**

| | |
|---|---|
| *pv4sessionId[O-UT]* | • IPv4 session id |
| *pv6sessionId[O-UT]* | • IPv6 session id |
| *ip[OUT]* | • IP address information see QmiWdsIpAddressInfo for more details |

### 8.1582.2 Field Documentation

#### 8.1582.2.1 QmiWdsIpAddressInfo WdsIpAddressInfoReq::ip

#### 8.1582.2.2 ULONG* WdsIpAddressInfoReq::pv4sessionId

#### 8.1582.2.3 ULONG* WdsIpAddressInfoReq::pv6sessionId

## 8.1583 WdsPktStatisticsElmnts Struct Reference

### Data Fields

- ULONG * pTXPacketSuccesses
- ULONG * pRXPacketSuccesses
- ULONG * pTXPacketErrors
- ULONG * pRXPacketErrors
- ULONG * pTXPacketOverflows
- ULONG * pRXPacketOverflows
- ULONGLONG * pTXOkBytesCount
- ULONGLONG * pRXOkBytesCount
- ULONGLONG * pTXOKBytesLastCall
- ULONGLONG * pRXOKBytesLastCall
- ULONG * pTXDroppedCount
- ULONG * pRXDroppedCount

### 8.1583.1 Detailed Description

WDS Pkt Statistics request data structure for individual session

**Parameters**

| | |
|---|---|
| *pTXPacket-Successes* | • Number of transmitted Packets without error. |
| *pRXPacket-Successes* | • Number of received Packets without error. |
| *pTXPacketErrors* | • Number of outgoing packets with framing errors. |
| *pRXPacket-Errors* | • Number of incoming packets with framing errors. |
| *pTXPacket-Overflows* | • Number of packets dropped because Tx buffer overflowed (out of memory). |
| *pRXPacket-Overflows* | • Number of packets dropped because Rx buffer overflowed (out of memory). |
| *pTXOkBytes-Count* | • Number of bytes transmitted without error. |
| *pRXOkBytes-Count* | • Number of bytes received without error. |
| *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 |
| *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 |
| *pTXDropped-Count* | • Number of outgoing packets dropped. |
| *pRXDropped-Count* | • Number of incoming packets dropped. |

### 8.1583.2 Field Documentation

#### 8.1583.2.1 ULONG* WdsPktStatisticsElmnts::pRXDroppedCount

#### 8.1583.2.2 ULONGLONG* WdsPktStatisticsElmnts::pRXOkBytesCount

#### 8.1583.2.3 ULONGLONG* WdsPktStatisticsElmnts::pRXOKBytesLastCall

#### 8.1583.2.4 ULONG* WdsPktStatisticsElmnts::pRXPacketErrors

#### 8.1583.2.5 ULONG* WdsPktStatisticsElmnts::pRXPacketOverflows

#### 8.1583.2.6 ULONG* WdsPktStatisticsElmnts::pRXPacketSuccesses

#### 8.1583.2.7 ULONG* WdsPktStatisticsElmnts::pTXDroppedCount

#### 8.1583.2.8 ULONGLONG* WdsPktStatisticsElmnts::pTXOkBytesCount

#### 8.1583.2.9 ULONGLONG* WdsPktStatisticsElmnts::pTXOKBytesLastCall

#### 8.1583.2.10 ULONG* WdsPktStatisticsElmnts::pTXPacketErrors

#### 8.1583.2.11 ULONG* WdsPktStatisticsElmnts::pTXPacketOverflows

#### 8.1583.2.12 ULONG* WdsPktStatisticsElmnts::pTXPacketSuccesses

## 8.1584 WdsPktStatisticsReq Struct Reference

### Data Fields

- ULONG * pStatMask

### 8.1584.1 Detailed Description

WDS PktStatistics request data structure

**Parameters**

| | |
|---|---|
| *pStatMask* | • Packet Statistics Mask 0x00000001 - Tx packets OK 0x00000002 - Rx packets OK 0x00000004 - Tx packet errors 0x00000008 - Rx packet errors 0x00000010 - Tx overflows 0x00000020 - Rx overflows 0x00000040 - Tx bytes OK 0x00000080 - Rx bytes OK |

### 8.1584.2 Field Documentation

#### 8.1584.2.1 ULONG* WdsPktStatisticsReq::pStatMask

## 8.1585 WdsPktStatisticsResp Struct Reference

### Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsPktStatisticsElmnts PktStatElmntsV4
- struct WdsPktStatisticsElmnts PktStatElmntsV6

### 8.1585.1 Detailed Description

WDS PktStatistics response data structure

**Parameters**

| | |
|---|---|
| *pV4sessionId* | • The v4 session ID for which the byte totals are to be retrieved<br>• provide a NULL pointer if not applicable |
| *pV6sessionId* | • The v6 session ID for which the byte totals are to be retrieved<br>• provide a NULL pointer if not applicable |
| *PktStatElmntsV4* | • data structure to be populated with the Pkt Statistics for V4 session |
| *PktStatElmntsV6* | • data structure to be populated with the Pkt Statistics for V6 session |

**Note**

At least one of pV4sessionId and pV6sessionId must point to a valid session ID.

### 8.1585.2 Field Documentation

#### 8.1585.2.1 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV4

#### 8.1585.2.2 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV6

#### 8.1585.2.3 ULONG* WdsPktStatisticsResp::pV4sessionId

#### 8.1585.2.4 ULONG* WdsPktStatisticsResp::pV6sessionId

## 8.1586 WdsProfileParam Union Reference

### Data Fields

- struct Profile3GPP SlqsProfile3GPP
- struct Profile3GPP2 SlqsProfile3GPP2

### 8.1586.1 Detailed Description

This union WdsProfileParam consist of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

**Parameters**

| | |
|---|---|
| *SlqsProfile3GPP* | • 3GPP profile |
| *SlqsProfile3GP-P2* | • 3GPP2 profile |

### 8.1586.2 Field Documentation

#### 8.1586.2.1 struct Profile3GPP WdsProfileParam::SlqsProfile3GPP

#### 8.1586.2.2 struct Profile3GPP2 WdsProfileParam::SlqsProfile3GPP2

## 8.1587 WdsRunTimeSettings Struct Reference

### Data Fields

- ULONG * v4sessionId
- ULONG * v6sessionId
- struct qmiWdsRunTimeSettings rts

### 8.1587.1 Detailed Description

WDS runtime settings request data structure

**Parameters**

| | |
|---|---|
| *v4sessionId* | • The v4 session ID for which the runtime settings are to be retrieved<br>• provide a NULL pointer if not applicable |
| *v6sessionId* | • The v6 session ID for which the runtime settings are to be retrieved<br>• provide a NULL pointer if not applicable |
| *qmiWdsRunTimeSettings* | • data structure to be populated with the runtime settings |

**Note**

At least one of v4sessionId and v6sessionId must point to a valid session ID.

### 8.1587.2 Field Documentation

#### 8.1587.2.1 struct qmiWdsRunTimeSettings WdsRunTimeSettings::rts

#### 8.1587.2.2 ULONG* WdsRunTimeSettings::v4sessionId

#### 8.1587.2.3 ULONG* WdsRunTimeSettings::v6sessionId

## 8.1588 wdsSetEventReportReq Struct Reference

### Data Fields

- BYTE * pCurrChannelRateInd
- TrStatInd * pTransferStatInd
- BYTE * pDataBearerTechInd
- BYTE * pDormancyStatusInd
- BYTE * pMIPStatusInd
- BYTE * pCurrDataBearerTechInd
- BYTE * pDataCallStatusChangeInd

- BYTE * pCurrPrefDataSysInd
- BYTE * pEVDOPageMonPerChangeInd
- BYTE * pDataSystemStatusChangeInd

### 8.1588.1 Detailed Description

This structure contains the information about the Set Event Report Request parameters.

**Parameters**

| | |
|---|---|
| *pCurrChannelRateInd* | (optional)<br>• Current Channel Rate Indicator.<br>– 0 - Do not report<br>– 1 - Report channel rate when it changes |
| *pTransferStatInd* | (optional)<br>• See TrStatInd for more information. |
| *pDataBearerTechInd* | (optional)<br>• Data Bearer Technology Indicator.<br>– 0 - Do not report<br>– 1 - Report radio interface used for data transfer when it changes |
| *pDormancyStatusInd* | (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 |
| *pCurrDataBearerTechInd* | (optional)<br>• Current Data Bearer Technology Indicator.<br>– 0 - Do not report<br>– 1 - Report current data bearer technology when it changes |
| *pDataCallStatusChangeInd* | (optional)<br>• Data Call Status Change Indicator.<br>– 0 - Do not report<br>– 1 - Report data call status change when it changes |
| *pCurrPrefDataSysInd* | (optional)<br>• Current Preferred Data System Indicator.<br>– 0 - Do not report<br>– 1 - Report preferred data system when it changes |
| *pEVDOPageMonPerChangeInd* | (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.

### 8.1588.2 Field Documentation

#### 8.1588.2.1 BYTE* wdsSetEventReportReq::pCurrChannelRateInd

#### 8.1588.2.2 BYTE* wdsSetEventReportReq::pCurrDataBearerTechInd

#### 8.1588.2.3 BYTE* wdsSetEventReportReq::pCurrPrefDataSysInd

#### 8.1588.2.4 BYTE* wdsSetEventReportReq::pDataBearerTechInd

#### 8.1588.2.5 BYTE* wdsSetEventReportReq::pDataCallStatusChangeInd

#### 8.1588.2.6 BYTE* wdsSetEventReportReq::pDataSystemStatusChangeInd

#### 8.1588.2.7 BYTE* wdsSetEventReportReq::pDormancyStatusInd

#### 8.1588.2.8 BYTE* wdsSetEventReportReq::pEVDOPageMonPerChangeInd

#### 8.1588.2.9 BYTE* wdsSetEventReportReq::pMIPStatusInd

#### 8.1588.2.10 TrStatInd* wdsSetEventReportReq::pTransferStatInd

## 8.1589 WDSSetLoopbackData Struct Reference

### Data Fields

- BYTE * pLoopbackMode
- BYTE * pLoopbackMultiplier

### 8.1589.1 Detailed Description

WDS SWI Set Loopback Structure of Set Loopback Information.

**Parameters**

| | |
|---|---|
| *pLoopbackMode* | • Loopback Mode.<br>– 0 - Disable<br>– 1 - Enable |
| *pLoopback-Multiplier* | • Loopback multiplier. Number of downlink bytes to send for each uplink byte. |

### 8.1589.2 Field Documentation

#### 8.1589.2.1 BYTE* WDSSetLoopbackData::pLoopbackMode

#### 8.1589.2.2 BYTE* WDSSetLoopbackData::pLoopbackMultiplier

## 8.1590 WDSSWICurrentChannelRates Struct Reference

### Data Fields

- unsigned long current_channel_tx_rate
- unsigned long current_channel_rx_rate
- unsigned long max_channel_tx_rate
- unsigned long max_channel_rx_rate

### 8.1590.1 Detailed Description

WDS SWI Current Channel Rates Structure of Packet Data Connection Information.

**Parameters**

| | |
|---|---|
| *current_channel-_tx_rate* | • Current Channel Tx Rate. |
| *current_channel-_rx_rate* | • Current Channel Rx Rate. |
| *max_channel_-tx_rate* | • Max Channel Tx Rate. |
| *max_channel_-rx_rate* | • Max Channel Rx Rate. |

### 8.1590.2 Field Documentation

#### 8.1590.2.1 unsigned long WDSSWICurrentChannelRates::current_channel_rx_rate

#### 8.1590.2.2 unsigned long WDSSWICurrentChannelRates::current_channel_tx_rate

#### 8.1590.2.3 unsigned long WDSSWICurrentChannelRates::max_channel_rx_rate

#### 8.1590.2.4 unsigned long WDSSWICurrentChannelRates::max_channel_tx_rate

# Chapter 9

# File Documentation

## 9.1 apdoxypages.c File Reference

Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages.

### Namespaces

- Tables

### 9.1.1 Detailed Description

Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages.

## 9.2 common.h File Reference

### Data Structures

- struct pack_qmi_t
- struct unpack_qmi_t

### Macros

- #define SDU_HDR_LEN (3)
- #define MINREQBKLEN (2048)
- #define MSGID_AND_LEN (4)
- #define MSGID_DONT_CARE (0xffff)
- #define UNUSEDPARAM(x) (void)x
- #define DEAULT_LOC_TIMEOUT_IN_SEC 2
- #define SDK_VALIDATE_INPUT_PACK_PARAM(pCtx, pBuf, pLen)
- #define SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID(pCtx, pBuf, pLen)
- #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

### 9.2.1 Macro Definition Documentation

#### 9.2.1.1 #define DEAULT_LOC_TIMEOUT_IN_SEC 2

#### 9.2.1.2 #define MINREQBKLEN (2048)

#### 9.2.1.3 #define MSGID_AND_LEN (4)

#### 9.2.1.4 #define MSGID_DONT_CARE (0xffff)

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

**Value:**

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

#### 9.2.1.6 #define SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID( *pCtx, pBuf, pLen* )

**Value:**

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
        {\
        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;
```

#### 9.2.1.7 #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; \
        }
```

#### 9.2.1.8 #define SDU_HDR_LEN (3)

#### 9.2.1.9 #define UNUSEDPARAM( *x* ) (void)x

### 9.2.2 Typedef Documentation

#### 9.2.2.1 typedef void(∗ logger)(uint8_t lvl, const char ∗buff)

### 9.2.3 Enumeration Type Documentation

#### 9.2.3.1 enum eLOG_LEVEL

log levels

**Enumerator**

***eLOG_INFO***
***eLOG_DEBUG***
***eLOG_WARN***
***eLOG_FATAL***

#### 9.2.3.2 enum eQMI_SVC

qmi service

**Enumerator**

***eCTL***
***eWDS***
***eDMS***
***eNAS***
***eQOS***
***eSMS***
***ePDS***
***eVOICE***
***eCAT***
***eUIM***
***eLOC***
***eSAR***
***eIMS***
***eTMD***
***eIMSA***
***eRMS***
***eSWIOMA***
***eAUDIO***
***eSWIM2MCMD***
***eSWIM2MCMD_AVC2***
***eSWIOMAEXT***
***eSWIAUDIO***
***eSWILOC***
***eSWIDMS***

#### 9.2.3.3 enum eTimeout

eTimeout

**Enumerator**

***eTIMEOUT_2_S***

***eTIMEOUT_5_S***

***eTIMEOUT_8_S***

***eTIMEOUT_10_S***

***eTIMEOUT_20_S***

***eTIMEOUT_30_S***

***eTIMEOUT_60_S***

***eTIMEOUT_300_S***

***eTIMEOUT_DEFAULT***

#### 9.2.3.4 enum msgtype

qmi message type

**Enumerator**

***eREQ***

***eRSP***

***eIND***

### 9.2.4 Function Documentation

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

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

#### 9.2.4.3 char* get_version ( )

**Returns**

version string

#### 9.2.4.4 uint16_t helper_get_error_code ( uint8_t * *qmi_resp* )

#### 9.2.4.5 char* helper_get_error_reason ( uint16_t *retVal* )

#### 9.2.4.6 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

#### 9.2.4.7 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

#### 9.2.4.8 uint16_t helper_get_xid ( uint8_t * *qmi_resp* )

#### 9.2.4.9 int helper_set_log_func ( logger *func* )

set log function

#### 9.2.4.10 int helper_set_log_lvl ( uint8_t *lvl* )

set log level

#### 9.2.4.11 char* liteqmi_GetVersion ( )

**Returns**

version string

#### 9.2.4.12 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 |

#### 9.2.4.13 void liteqmi_log ( uint8_t *lvl,* const char * *fmt,* ... )

#### 9.2.4.14 unsigned unpack_result_code_only ( uint8_t * *pMdmResp* )

common handler for unpacking response with TLV type 0x02 only

### 9.2.5 Variable Documentation

#### 9.2.5.1 logger glog

#### 9.2.5.2 uint8_t gloglvl

## 9.3 dms.h File Reference

### Data Structures

- struct unpack_dms_GetModelID_t
- struct unpack_dms_GetIMSI_t
- struct unpack_dms_GetFirmwareInfo_t
- struct unpack_dms_GetPower_t
- struct unpack_dms_GetSerialNumbers_t
- struct unpack_dms_GetHardwareRevision_t
- struct unpack_dms_SLQSGetBandCapability_t
- struct unpack_dms_GetDeviceCapabilities_t
- struct unpack_dms_GetFirmwareRevisions_t
- struct unpack_dms_GetFirmwareRevision_t
- struct unpack_dms_GetDeviceSerialNumbers_t
- struct unpack_dms_GetPRLVersion_t
- struct unpack_dms_GetNetworkTime_t
- struct unpack_dms_GetVoiceNumber_t
- struct unpack_dms_GetDeviceHardwareRev_t
- struct unpack_dms_GetFSN_t
- struct unpack_dms_GetDeviceCap_t
- struct pack_dms_SetPower_t
- struct unpack_dms_SetPower_t
- struct unpack_dms_GetBandCapability_t
- struct unpack_dms_GetUSBComp_t
- struct pack_dms_SetUSBComp_t
- struct unpack_dms_SetUSBComp_t
- struct pack_dms_SetCustFeature_t
- struct unpack_dms_SetCustFeature_t
- struct unpack_dms_GetCustFeature_t
- struct unpack_dms_SetFirmwarePreference_t
- struct unpack_dms_GetCrashAction_t
- struct pack_dms_SetCrashAction_t
- struct unpack_dms_SetCrashAction_t
- struct unpack_dms_GetDeviceMfr_t
- struct pack_dms_SetEventReport_t
- struct unpack_dms_SetEventReport_t
- struct dms_OperatingModeTlv
- struct dms_ActivationStatusTlv
- struct unpack_dms_SetEventReport_ind_t
- struct pack_dms_UIMGetICCID_t
- struct unpack_dms_UIMGetICCID_t
- struct pack_dms_SetCustFeaturesV2_t
- struct unpack_dms_SetCustFeaturesV2_t
- struct pack_dms_GetCustFeaturesV2_t
- struct DMSgetCustomInput
- struct DMScustSettingInfo
- struct DMScustSettingList

- struct DMSgetCustomFeatureV2
- struct unpack_dms_GetCustFeaturesV2_t
- struct unpack_dms_GetActivationState_t
- struct image_info_t
- struct unpack_dms_SLQSSwiGetFirmwareCurr_t
- struct pack_dms_SLQSSwiSetDyingGaspCfg_t
- struct unpack_dms_SLQSSwiSetDyingGaspCfg_t
- struct unpack_dms_SLQSSwiClearDyingGaspStatistics_t
- struct packgetDyingGaspStatistics
- struct unpack_dms_SLQSSwiGetDyingGaspStatistics_t
- struct packgetDyingGaspCfg
- struct unpack_dms_SLQSSwiGetDyingGaspCfg_t
- struct unpack_dms_SLQSDmsSwiGetResetInfo_t
- struct unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t
- struct pack_dms_SLQSDmsSwiIndicationRegister_t
- struct unpack_dms_SLQSDmsSwiIndicationRegister_t
- struct unpack_dms_SLQSSwiGetFwUpdateStatus_t
- struct unpack_dms_GetManufacturer_t
- struct unpack_dms_GetOfflineReason_t
- struct pack_dms_SetActivationStatusCallback_t
- struct unpack_dms_SetActivationStatusCallback_t
- struct pack_dms_UIMSetPINProtection_t
- struct unpack_dms_UIMSetPINProtection_t
- struct pack_dms_UIMUnblockPIN_t
- struct pack_dms_UIMVerifyPIN_t
- struct pack_dms_UIMChangePIN_t
- struct pack_dms_UIMGetControlKeyStatus_t
- struct unpack_dms_UIMGetControlKeyStatus_t
- struct unpack_dms_UIMGetPINStatus_t
- struct pack_dms_UIMSetControlKeyProtection_t
- struct unpack_dms_UIMSetControlKeyProtection_t
- struct pack_dms_UIMUnblockControlKey_t
- struct unpack_dms_UIMUnblockControlKey_t
- struct pack_dms_ResetToFactoryDefaults_t
- struct unpack_dms_ResetToFactoryDefaults_t
- struct pack_dms_ActivateAutomatic_t
- struct eriDataparams
- struct unpack_dms_SLQSGetERIFile_t
- struct unpack_dms_SLQSUIMGetState_t
- struct pack_dms_SLQSSwiGetCrashInfo_t
- struct crashInformation
- struct crashInfoParams
- struct unpack_dms_SLQSSwiGetCrashInfo_t
- struct unpack_dms_SLQSSwiGetHostDevInfo_t
- struct pack_dms_SLQSSwiSetHostDevInfo_t
- struct unpack_dms_SLQSSwiSetHostDevInfo_t
- struct unpack_dms_SLQSSwiGetOSInfo_t
- struct pack_dms_SLQSSwiSetOSInfo_t
- struct unpack_dms_SLQSSwiSetOSInfo_t
- struct unpack_dms_SLQSSwiGetSerialNoExt_t
- 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)

### 9.3.1 Macro Definition Documentation

#### 9.3.1.1 #define ACT_CODE_MAX_SIZE 81

#### 9.3.1.2 #define CK_MAX_SIZE 8

#### 9.3.1.3 #define DMS_IMGDETAILS_LEN 16

#### 9.3.1.4 #define DMS_MAX_CUST_ID_LEN 64

#### 9.3.1.5 #define DMS_MAX_CUST_VALUE_LEN 8

#### 9.3.1.6 #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255

#### 9.3.1.7 #define DMS_MAX_FWUPDATE_REF_STR_SZ 15

#### 9.3.1.8 #define DMS_MAX_RADIO_IFCS_SIZE 64

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

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

#### 9.3.1.11 #define DMS_PM_OFFLINE 0x03 /* Offline */

#### 9.3.1.12 #define DMS_PM_ONLINE 0x00 /* Online */

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

#### 9.3.1.14 #define DMS_PM_RESET 0x04 /* Reset */

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

#### 9.3.1.16 #define DMS_SET_REG_IND_DISABLE 0

#### 9.3.1.17 #define DMS_SET_REG_IND_ENABLE 1

#### 9.3.1.18 #define DMS_SET_REG_IND_NO_CHANGE 2

#### 9.3.1.19 #define DMS_SET_REPORT_DISABLE 0

#### 9.3.1.20 #define DMS_SET_REPORT_ENABLE 1

#### 9.3.1.21 #define DMS_SLQSFWINFO_APPVERSION_SZ 85

#### 9.3.1.22 #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85

#### 9.3.1.23 #define DMS_SLQSFWINFO_CARRIER_SZ 20

#### 9.3.1.24 #define DMS_SLQSFWINFO_CUR_CARR_NAME 17

#### 9.3.1.25 #define DMS_SLQSFWINFO_CUR_CARR_REV 13

#### 9.3.1.26 #define DMS_SLQSFWINFO_MODELID_SZ 20

#### 9.3.1.27 #define DMS_SLQSFWINFO_PACKAGEID_SZ 85

#### 9.3.1.28 #define DMS_SLQSFWINFO_PRIVERSION_SZ 16

#### 9.3.1.29 #define DMS_SLQSFWINFO_SKU_SZ 15

#### 9.3.1.30 #define DMS_SWI_SET_IND_DISABLE 0

#### 9.3.1.31 #define DMS_SWI_SET_IND_ENABLE 1

#### 9.3.1.32 #define DMS_UINT8_MAX_STRING_SZ 255

#### 9.3.1.33 #define DMS_VALID_FSN_LEN 14

#### 9.3.1.34 #define ERI_DATA_MAX_SIZE 1024

#### 9.3.1.35 #define MAX_BUILD_ID_LEN 255

#### 9.3.1.36 #define MEID_MAX_SIZE 8

#### 9.3.1.37 #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160

#### 9.3.1.38 #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20

#### 9.3.1.39 #define SPC_SIZE 6

#### 9.3.1.40 #define UNIQUE_ID_LEN 16

### 9.3.2 Function Documentation

#### 9.3.2.1 int pack_dms_ActivateAutomatic ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_ActivateAutomatic_t * *pReq* )

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

#### 9.3.2.2 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.

#### 9.3.2.3 int pack_dms_GetBandCapability ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Band Capability pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.3.2.4 int pack_dms_GetCrashAction ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Crash Action pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.3.2.5 int pack_dms_GetCustFeature ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use pack_dms_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

#### 9.3.2.6 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

#### 9.3.2.7 int pack_dms_GetDeviceCap ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Device Capabilities pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.3.2.8 int pack_dms_GetDeviceCapabilities ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get device capability pack

**Parameters**

| | | |
|---|---|---|
| in,out | *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

#### 9.3.2.9 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

#### 9.3.2.10 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

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

Get Device Serial Number pack

**Parameters**

| | | |
|---|---|---|
| in,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

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

Get firmware info pack

**Parameters**

| | | |
|---|---|---|
| in,out | *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

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

Get Firmware Revision pack

**Parameters**

| | | |
|---|---|---|
| in,out | *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

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

Get Firmware Revisions pack

**Parameters**

| | | |
|---|---|---|
| in,out | *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

#### 9.3.2.15 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

#### 9.3.2.16 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

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

Get IMSI pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_-xx_xx and all EM74xx firmware versions. Please use pack_uim_ReadTransparent()(EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

**Parameters**

| | | |
|---|---|---|
| in,out | *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

#### 9.3.2.18 int pack_dms_GetManufacturer ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

To get device manufacturer information pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *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

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

Get model id pack

**Parameters**

| | | |
|---|---|---|
| in,out | *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

#### 9.3.2.20 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

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

To get operating mode offline reason pack

**Parameters**

| | | |
|---|---|---|
| in,out | *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

#### 9.3.2.22 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

#### 9.3.2.23 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

#### 9.3.2.24 int pack_dms_GetSerialNumbers ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get serial numbers pack

**Parameters**

| | | |
|---|---|---|
| in,out | *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

#### 9.3.2.25 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

#### 9.3.2.26 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

#### 9.3.2.27 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

#### 9.3.2.28 int pack_dms_SetActivationStatusCallback ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetActivationStatusCallback_t * *reqArg* )

Set activation status pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.3.2.29 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

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

Set Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use pack_dms_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

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

Set Cust Features pack.

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.3.2.32 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

#### 9.3.2.33 int pack_dms_SetFirmwarePreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Set Firmware Preference pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.3.2.34 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

#### 9.3.2.35 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

#### 9.3.2.36 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

#### 9.3.2.37 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

#### 9.3.2.38 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

#### 9.3.2.39 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

#### 9.3.2.40 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

#### 9.3.2.41 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

#### 9.3.2.42 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

#### 9.3.2.43 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

#### 9.3.2.44 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

#### 9.3.2.45 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

#### 9.3.2.46 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

#### 9.3.2.47 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

#### 9.3.2.48 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

#### 9.3.2.49 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

#### 9.3.2.50 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

#### 9.3.2.51 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

#### 9.3.2.52 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

#### 9.3.2.53 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

#### 9.3.2.54 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

#### 9.3.2.55 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

#### 9.3.2.56 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 SWI9X15C_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

#### 9.3.2.57 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

#### 9.3.2.58 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

#### 9.3.2.59 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

#### 9.3.2.60 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

#### 9.3.2.61 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

#### 9.3.2.62 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

#### 9.3.2.63 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

#### 9.3.2.64 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

#### 9.3.2.65 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

### 9.3.2.66 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

### 9.3.2.67 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

#### 9.3.2.68 int unpack_dms_ActivateAutomatic ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_ResetToFactory-Defaults_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

#### 9.3.2.69 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.

#### 9.3.2.70 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

### 9.3.2.71 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

### 9.3.2.72 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

### 9.3.2.73 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

#### 9.3.2.74 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

#### 9.3.2.75 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

#### 9.3.2.76 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

#### 9.3.2.77 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

#### 9.3.2.78 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

#### 9.3.2.79 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

#### 9.3.2.80 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

#### 9.3.2.81 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

#### 9.3.2.82 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

### 9.3.2.83 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

### 9.3.2.84 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

#### 9.3.2.85 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

#### 9.3.2.86 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

#### 9.3.2.87 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

#### 9.3.2.88 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

#### 9.3.2.89 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

#### 9.3.2.90 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

#### 9.3.2.91 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

#### 9.3.2.92 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

#### 9.3.2.93 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

### 9.3.2.94 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

### 9.3.2.95 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

### 9.3.2.96 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

#### 9.3.2.97 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

#### 9.3.2.98 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

#### 9.3.2.99 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

### 9.3.2.100 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

### 9.3.2.101 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

#### 9.3.2.102 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

#### 9.3.2.103 int unpack_dms_SetIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetIndicationRegister_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

#### 9.3.2.104 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

#### 9.3.2.105 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

#### 9.3.2.106 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

#### 9.3.2.107 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

#### 9.3.2.108 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

#### 9.3.2.109 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

#### 9.3.2.110 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

### 9.3.2.111 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

### 9.3.2.112 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

#### 9.3.2.113 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

#### 9.3.2.114 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

#### 9.3.2.115 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

#### 9.3.2.116 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

#### 9.3.2.117 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

#### 9.3.2.118 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

#### 9.3.2.119 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

#### 9.3.2.120 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

#### 9.3.2.121 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

### 9.3.2.122 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

### 9.3.2.123 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

### 9.3.2.124 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

#### 9.3.2.125 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

#### 9.3.2.126 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 SWI9X15C_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

#### 9.3.2.127 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

### 9.3.2.128 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

### 9.3.2.129 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

#### 9.3.2.130 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

#### 9.3.2.131 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

#### 9.3.2.132 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

#### 9.3.2.133 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

#### 9.3.2.134 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

#### 9.3.2.135 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

#### 9.3.2.136 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

#### 9.3.2.137 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

#### 9.3.2.138 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

## 9.4 fms.h File Reference

### Data Structures

- struct CarrierImage_t
- struct pack_fms_GetImagesPreference_t
- struct FMSImageElement
- struct FMSPrefImageList
- struct unpack_fms_GetImagesPreference_t
- struct pack_fms_GetStoredImages_t
- struct FMSImageIdElement
- struct FMSImageIDEntries
- struct FMSImageList
- struct unpack_fms_GetStoredImages_t
- struct pack_fms_SetImagesPreference_t
- struct unpack_fms_SetImagesPreference_t

### Macros

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

### Functions

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

### 9.4.1 Macro Definition Documentation

#### 9.4.1.1 #define FMS_FW_PRI_BUILD_MATCH_LEN 11

#### 9.4.1.2 #define FMS_GOBI_LISTENTRIES_MAX 2

#### 9.4.1.3 #define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100

#### 9.4.1.4 #define FMS_GOBI_MBN_IMG_ID_STR_LEN 16

#### 9.4.1.5 #define FMS_IMAGE_ID_BUILD_ID_LEN 32

#### 9.4.1.6 #define FMS_IMAGE_ID_IMG_ID_LEN 16

#### 9.4.1.7 #define FMS_IMAGE_ID_MAX_ENTRIES 2

#### 9.4.1.8 #define FMS_IMAGE_ID_PRI_IMGTYPE 0x01

#### 9.4.1.9 #define FMS_MAX_IMAGE_ID_ELEMENT 50

#### 9.4.1.10 #define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255

### 9.4.2 Function Documentation

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

This API distills valid Firmware/PRI combinations from GetStoredImages result

**Parameters**

| | | |
|---|---|---|
| in | *pStoredImage-List* | • image list returned from GetStoredImages<br>• See FMSImageList |
| in,out | *pValid-CombinationSize* | • number of combination passed in and returned |
| out | *pValid-Combinations* | • 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

#### 9.4.2.2 int pack_fms_GetImagesPreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_GetImagesPreference_t * *reqArg* )

Get Images Preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

#### 9.4.2.3 int pack_fms_GetStoredImages ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_GetStoredImages_t * *reqArg* )

Get Images Preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

#### 9.4.2.4 int pack_fms_SetImagesPreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_SetImagesPreference_t * *reqArg* )

Set Images Preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

#### 9.4.2.5 int unpack_fms_GetImagesPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_GetImagesPreference_t * *pOutput* )

Get Images Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

#### 9.4.2.6 int unpack_fms_GetStoredImages ( uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_GetStoredImages_t * *pOutput* )

Get Images Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

#### 9.4.2.7 int unpack_fms_SetImagesPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_SetImagesPreference_t * *pOutput* )

Set Images Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

## 9.5 loc.h File Reference

### Data Structures

- struct loc_LocApplicationInfo
- struct loc_SV
- struct loc_SVInfo
- struct loc_GnssData
- struct loc_CellDb
- struct loc_ClkInfo
- struct loc_BdsSV
- struct loc_BdsSVInfo
- struct pack_loc_EventRegister_t
- struct unpack_loc_EventRegister_t
- struct pack_loc_SetExtPowerState_t
- struct unpack_loc_SetExtPowerState_t
- struct pack_loc_Start_t
- struct unpack_loc_Start_t
- struct pack_loc_Stop_t
- struct unpack_loc_Stop_t
- struct pack_loc_SetOperationMode_t
- struct unpack_loc_SetOperationMode_t
- struct pack_loc_Delete_Assist_Data_t
- struct unpack_loc_Delete_Assist_Data_t
- struct loc_precisionDilution
- struct loc_sensorDataUsage
- struct loc_svUsedforFix
- struct loc_gpsTime
- struct unpack_loc_PositionRpt_Ind_t
- struct unpack_loc_EngineState_Ind_t
- struct unpack_loc_SetExtPowerConfig_Ind_t
- struct unpack_loc_SLQSLOCGetBestAvailPos_t
- struct pack_loc_SLQSLOCGetBestAvailPos_t
- struct unpack_loc_BestAvailPos_Ind_t
- 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

## 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)
- int pack_loc_SLQSLOCInjectPosition (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectPosition_t *reqArg)
- int unpack_loc_SLQSLOCInjectPosition (uint8_t *pResp, uint16_t respLen)
- int pack_loc_SLQSLOCSetCradleMountConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCSetCradleMountConfig_t *reqArg)
- int unpack_loc_SLQSLOCSetCradleMountConfig (uint8_t *pResp, uint16_t respLen)
- int pack_loc_SLQSLOCInjectSensorData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectSensorData_t *reqArg)

- int unpack_loc_SLQSLOCInjectSensorData (uint8_t *pResp, uint16_t respLen)
- 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)
- 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)

### 9.5.1 Macro Definition Documentation

#### 9.5.1.1 #define LOC_UINT8_MAX_STRING_SZ 255

#### 9.5.1.2 #define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000

The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.

#### 9.5.1.3 #define LOCEVENTMASKENGINESTATE 0x00000080

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

#### 9.5.1.4 #define LOCEVENTMASKFIXSESSIONSTATE 0x00000100

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

#### 9.5.1.5 #define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.

#### 9.5.1.6 #define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence.

#### 9.5.1.7 #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000

The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, for example, if GPS is turned off or if the network is unavailable.

#### 9.5.1.8 #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000

The control point must enable this mask to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_ CONSTELL_REPORT_CONFIG(Not yet supported).

#### 9.5.1.9 #define LOCEVENTMASKGNSSSVINFO 0x00000002

The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate.

#### 9.5.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040

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

#### 9.5.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020

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

#### 9.5.1.12 #define LOCEVENTMASKINJECTTIMEREQ 0x00000010

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

#### 9.5.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000

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

#### 9.5.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF

Invalid Event Mask

#### 9.5.1.15 #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000

The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing.

#### 9.5.1.16 #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000

The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.

#### 9.5.1.17 #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000

The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data.

#### 9.5.1.18 #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000

The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.

#### 9.5.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008

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

#### 9.5.1.20 #define LOCEVENTMASKNMEA 0x00000004

The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.

#### 9.5.1.21 #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000

The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.

#### 9.5.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001

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

#### 9.5.1.23 #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.).

#### 9.5.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000

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

#### 9.5.1.25 #define LOCEVENTMASKTIMESYNCREQ 0x00000800

The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.

#### 9.5.1.26 #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).

#### 9.5.1.27 #define LOCEVENTMASKWIFIREQ 0x00000200

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

#### 9.5.1.28 #define MAX_LOC_NMEA_STR_LEN 201

#### 9.5.1.29 #define MAX_SENSOR_DATA_LEN 64

#### 9.5.1.30 #define MAX_TEMP_DATA_LEN 64

### 9.5.2 Enumeration Type Documentation

#### 9.5.2.1 anonymous enum

**Enumerator**

***eQMI_LOC_SESS_STATUS_SUCCESS***

***eQMI_LOC_SESS_STATUS_IN_PROGRESS***

***eQMI_LOC_SESS_STATUS_FAILURE***

***eQMI_LOC_SESS_STATUS_TIMEOUT***

### 9.5.3 Function Documentation

#### 9.5.3.1 int pack_loc_DeleteAssistData ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_Delete_Assist_Data_t * *reqArg* )

Delete Assistant Data pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.5.3.2 int pack_loc_EventRegister ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_EventRegister_t * *reqArg* )

Event Register pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.5.3.3 int pack_loc_SetExtPowerState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SetExtPowerState_t * *reqArg* )

Set Ext Power State pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.5.3.4 int pack_loc_SetOperationMode ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SetOperationMode_t * *reqArg* )

Set Operation Mode pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.5.3.5 int pack_loc_SLQSLOCGetBestAvailPos ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCGetBestAvailPos_t * *reqArg* )

Get Best Avail position pack

**Parameters**

| | | |
|---|---|---|
| in,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

#### 9.5.3.6 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

#### 9.5.3.7 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

#### 9.5.3.8 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

#### 9.5.3.9 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

#### 9.5.3.10 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

#### 9.5.3.11 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

#### 9.5.3.12 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

#### 9.5.3.13 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

#### 9.5.3.14 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

#### 9.5.3.15 int unpack_loc_CradleMountCallback_Ind ( uint8_t * pResp, uint16_t respLen, unpack_loc_CradleMount-Callback_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

#### 9.5.3.16 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

#### 9.5.3.17 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

#### 9.5.3.18 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

#### 9.5.3.19 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

#### 9.5.3.20 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

#### 9.5.3.21 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

#### 9.5.3.22 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

#### 9.5.3.23 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

#### 9.5.3.24 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

#### 9.5.3.25 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

#### 9.5.3.26 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

#### 9.5.3.27 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

#### 9.5.3.28 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

#### 9.5.3.29 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

#### 9.5.3.30 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

#### 9.5.3.31 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

#### 9.5.3.32 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

#### 9.5.3.33 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

#### 9.5.3.34 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

#### 9.5.3.35 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

#### 9.5.3.36 int unpack_loc_SLQSLOCGetServer ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack get 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

#### 9.5.3.37 int unpack_loc_SLQSLOCInjectPosition ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack inject position.

**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

#### 9.5.3.38 int unpack_loc_SLQSLOCInjectSensorData ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack inject sensor data.

**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

#### 9.5.3.39 int unpack_loc_SLQSLOCInjectUTCTime ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack inject UTC time.

**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

#### 9.5.3.40 int unpack_loc_SLQSLOCSetCradleMountConfig ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack set cradle mount configure.

**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

#### 9.5.3.41 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

#### 9.5.3.42 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

#### 9.5.3.43 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

## 9.6 nas.h File Reference

### Data Structures

- struct unpack_nas_GetSignalStrengths_t
- struct unpack_nas_SLQSGetSysSelectionPref_t
- struct nas_netSelectionPref
- struct nas_acqOrderPref
- struct nas_CSGID
- struct pack_nas_SLQSSetSysSelectionPref_t
- struct 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 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

## 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

## 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_SLQSGetSysSelectionPref_t *pOutput)
- int pack_nas_SLQSSetSysSelectionPref (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSysSelectionPref_t *pReqParam)
- int unpack_nas_SLQSSetSysSelectionPref (uint8_t *pResp, uint16_t respLen)
- int pack_nas_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)
- 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)
- int pack_nas_SLQSNasIndicationRegisterExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasIndicationRegisterExt_t *pReqParam)
- int unpack_nas_SLQSNasIndicationRegisterExt (uint8_t *pResp, uint16_t respLen)
- int pack_nas_GetRFInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetRFInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_GetRFInfo_t *pOutput)
- int pack_nas_SLQSNasGetSigInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetSigInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetSigInfo_t *pOutput)
- int unpack_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)
- int pack_nas_SetRFInfoCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetRFInfoCallback (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SetLURejectCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetLURejectCallback (uint8_t *pResp, uint16_t respLen)
- int unpack_nas_SetEventReportInd (uint8_t *pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t *pOutput)
- int pack_nas_GetCDMANetworkParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetCDMANetworkParameters (uint8_t *pResp, uint16_t respLen, unpack_nas_GetCDMANetworkParameters_t *pOutput)
- int 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, uint32_t *pAuthStatus)
- int pack_nas_GetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetACCOLC (uint8_t *pResp, uint16_t respLen, uint8_t *pAccolc)
- int pack_nas_SetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetACCOLC_t reqParam)
- int unpack_nas_SetACCOLC (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasConfigSigInfo2_t *pReqParam)
- int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t *pResp, uint16_t respLen)
- int unpack_nas_SetDataCapabilitiesCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetDataCapabilitiesCallback_ind_t *pOutput)
- int pack_nas_GetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)

- int unpack_nas_GetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_GetNetworkPreference_t *pOutput)
- int pack_nas_SetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetNetworkPreference_t *reqArg)
- int unpack_nas_SetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t *pOutput)
- int unpack_nas_SetRoamingIndicatorCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetRoamingIndicatorCallback_ind_t *pOutput)
- int unpack_nas_SetServingSystemCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t *pOutput)
- int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SlqsGetLTECphyCAInfo_t *pOutput)
- int unpack_nas_SLQSSetSysSelectionPrefCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t *pOutput)
- int unpack_nas_SLQSNasSwiOTAMessageCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t *pOutput)
- int pack_nas_SLQSInitiateNetworkRegistration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSInitiateNetworkRegistration_t *pReqParam)
- int unpack_nas_SLQSInitiateNetworkRegistration (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNasSwiIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasSwiIndicationRegister_t *pReqParam)
- int unpack_nas_SLQSNasSwiIndicationRegister (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSGetPLMNName (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSGetPLMNName_t *reqArg)
- int unpack_nas_SLQSGetPLMNName (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetPLMNName_t *pOutput)
- int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetCellLocationInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetCellLocationInfo_t *pOutput)
- int pack_nas_SLQSGetNetworkTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetNetworkTime (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetNetworkTime_t *pOutput)
- int unpack_nas_SLQSNasNetworkTimeCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t *pOutput)
- int unpack_nas_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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)

### 9.6.1 Macro Definition Documentation

#### 9.6.1.1 #define NAS_IMSI_M_S1_LENGTH 7

#### 9.6.1.2 #define NAS_IMSI_M_S2_LENGTH 3

#### 9.6.1.3 #define NAS_MAX_DESCRIPTION_LENGTH 255

#### 9.6.1.4 #define NAS_MAX_NUM_NETWORKS 30

#### 9.6.1.5 #define NAS_MAX_SCC_RX_INFO_INSTANCES 255

#### 9.6.1.6 #define NAS_NAM_NAME_LENGTH 12

#### 9.6.1.7 #define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048

#### 9.6.1.8 #define NAS_PLMN_LENGTH 3

#### 9.6.1.9 #define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255

### 9.6.2 Enumeration Type Documentation

#### 9.6.2.1 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*

#### 9.6.2.2 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*

#### 9.6.2.3 enum NAS_LTE_CPHY_CA_BW_NRB_LITE

Enumerator

*eNAS_LTE_CPHY_CA_BW_NRB_LITE_6*

*eNAS_LTE_CPHY_CA_BW_NRB_LITE_15*

*eNAS_LTE_CPHY_CA_BW_NRB_LITE_25*

*eNAS_LTE_CPHY_CA_BW_NRB_LITE_50*

*eNAS_LTE_CPHY_CA_BW_NRB_LITE_75*

*eNAS_LTE_CPHY_CA_BW_NRB_LITE_100*

#### 9.6.2.4 enum NAS_LTE_CPHY_SCELL_STATE_LITE

**Enumerator**

***eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE***

***eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE***

***eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE***

### 9.6.3 Function Documentation

#### 9.6.3.1 int pack_nas_GetACCOLC ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

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

#### 9.6.3.2 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

#### 9.6.3.3 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

#### 9.6.3.4 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

#### 9.6.3.5 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

#### 9.6.3.6 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 |

#### 9.6.3.7 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

#### 9.6.3.8 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

#### 9.6.3.9 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

#### 9.6.3.10 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

#### 9.6.3.11 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

#### 9.6.3.12 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

#### 9.6.3.13 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

#### 9.6.3.14 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

#### 9.6.3.15 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

#### 9.6.3.16 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_SLQSSetSysSelectionPref() 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 |

#### 9.6.3.17 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

#### 9.6.3.18 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

#### 9.6.3.19 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

#### 9.6.3.20 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 |

#### 9.6.3.21 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

#### 9.6.3.22 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

#### 9.6.3.23 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

#### 9.6.3.24 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

#### 9.6.3.25 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_SLQSNasGetSigInfo() 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

#### 9.6.3.26 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

#### 9.6.3.27 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

#### 9.6.3.28 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

#### 9.6.3.29 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

#### 9.6.3.30 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

#### 9.6.3.31 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

#### 9.6.3.32 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

#### 9.6.3.33 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

#### 9.6.3.34 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

#### 9.6.3.35 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

#### 9.6.3.36 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

#### 9.6.3.37 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

#### 9.6.3.38 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

#### 9.6.3.39 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

#### 9.6.3.40 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

#### 9.6.3.41 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

#### 9.6.3.42 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

#### 9.6.3.43 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

#### 9.6.3.44 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

#### 9.6.3.45 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

#### 9.6.3.46 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

#### 9.6.3.47 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

#### 9.6.3.48 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

#### 9.6.3.49 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

#### 9.6.3.50 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

#### 9.6.3.51 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

#### 9.6.3.52 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

#### 9.6.3.53 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

#### 9.6.3.54 int unpack_nas_GetACCOLC ( uint8_t * *pResp*, uint16_t *respLen*, uint8_t * *pAccolc* )

Retrieves information about the access overload class (ACCOLC) unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pAccolc* | An 8-bit integer representation of the ACCOLC.<br>• Range: 0 to 15 (0x00 to 0x0F).<br>• 0xFF invalid data. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.6.3.55 int unpack_nas_GetANAAAAuthenticationStatus ( uint8_t * *pResp*, uint16_t *respLen*, uint32_t * *pAuthStatus* )

AN-AAA authentication status of the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pAuthStatus* | AN-AAA Authentication Status. 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>• 0xffffffff - invalid data. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.6.3.56 int unpack_nas_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

#### 9.6.3.57 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

#### 9.6.3.58 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

#### 9.6.3.59 int unpack_nas_GetNetworkPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetNetworkPreference_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

#### 9.6.3.60 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

#### 9.6.3.61 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

#### 9.6.3.62 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

#### 9.6.3.63 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_SLQSNasGetSig-Info() 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

#### 9.6.3.64 int unpack_nas_InitiateDomainAttach ( uint8_t * *pResp*, uint16_t *respLen* )

Initiates a domain attach/detach of the device 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

#### 9.6.3.65 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

#### 9.6.3.66 int unpack_nas_SetACCOLC ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack Sets the access overload class (ACCOLC)

**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

#### 9.6.3.67 int unpack_nas_SetCDMANetworkParameters ( uint8_t * *pResp*, uint16_t *respLen* )

CDMA Network 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

#### 9.6.3.68 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

#### 9.6.3.69 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

#### 9.6.3.70 int unpack_nas_SetLURejectCallback ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack set registration reject Callback

**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

#### 9.6.3.71 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

#### 9.6.3.72 int unpack_nas_SetNetworkPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNetworkPreference_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

#### 9.6.3.73 int unpack_nas_SetRFInfoCallback ( uint8_t * *pResp*, uint16_t *respLen* )

Unack Set RF Band Information callback.

**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

#### 9.6.3.74 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

#### 9.6.3.75 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

#### 9.6.3.76 int unpack_nas_SLQSConfigSigInfo ( uint8_t * *pResp*, uint16_t *respLen* )

Config Sig Info 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

#### 9.6.3.77 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

#### 9.6.3.78 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

#### 9.6.3.79 int unpack_nas_SLQSGetNetworkTime ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetNetwork-Time_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

#### 9.6.3.80 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

#### 9.6.3.81 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

#### 9.6.3.82 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

#### 9.6.3.83 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

#### 9.6.3.84 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

#### 9.6.3.85 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

#### 9.6.3.86 int unpack_nas_SLQSInitiateNetworkRegistration ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack initiates a network registration.

**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

#### 9.6.3.87 int unpack_nas_SLQSNasConfigSigInfo2 ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack sets the signal strength reporting thresholds

**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

#### 9.6.3.88 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

#### 9.6.3.89 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

#### 9.6.3.90 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

#### 9.6.3.91 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

#### 9.6.3.92 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

#### 9.6.3.93 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

#### 9.6.3.94 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

#### 9.6.3.95 int unpack_nas_SLQSNasIndicationRegisterExt ( uint8_t * *pResp*, uint16_t *respLen* )

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 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.6.3.96 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

#### 9.6.3.97 int unpack_nas_SLQSNASSeteDRXParams ( uint8_t * *pResp*, uint16_t *respLen* )

Set eDRX 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

#### 9.6.3.98 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

#### 9.6.3.99 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

#### 9.6.3.100 int unpack_nas_SLQSNasSwiIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen* )

unpack sets the registration state for different QMI_NAS SWI indications.

**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

#### 9.6.3.101 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

#### 9.6.3.102 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

#### 9.6.3.103 int unpack_nas_SLQSNASSwiSetChannelLock ( uint8_t * *pResp*, uint16_t *respLen* )

Set Channel Lock 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

#### 9.6.3.104 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

#### 9.6.3.105 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

#### 9.6.3.106 int unpack_nas_SLQSSetBandPreference ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack sets the different system selection preferences of the device.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

#### 9.6.3.107 int unpack_nas_SLQSSetSignalStrengthsCallback ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack set signal strength thresholds callback.

**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

#### 9.6.3.108 int unpack_nas_SLQSSetSysSelectionPref ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the different system selection preferences of the device 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

#### 9.6.3.109 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

#### 9.6.3.110 int unpack_nas_SLQSSetSysSelectionPrefExt ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the different system selection preferences of the device 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

#### 9.6.3.111 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

#### 9.6.3.112 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

#### 9.6.3.113 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

#### 9.6.3.114 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

#### 9.6.3.115 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

#### 9.6.3.116 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

### 9.6.3.117 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

### 9.6.3.118 int unpack_nas_SLQSSwiPSDetach ( uint8_t * *pResp*, uint16_t *respLen* )

PS Connection Detach 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

### 9.6.3.119 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

#### 9.6.3.120 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

#### 9.6.3.121 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

#### 9.6.3.122 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

#### 9.6.3.123 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

## 9.7 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

## 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)
- 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)
- 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)
- 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)
- int pack_pds_StopPDSTrackingSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_StopPDSTrackingSession (uint8_t *pResp, uint16_t respLen)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- int pack_pds_SLQSPDSDeterminePosition (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_SLQSPDSDeterminePosition (uint8_t *pResp, uint16_t respLen)
- 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_SLQSGetGPSStateInfo_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)
- int pack_pds_SetEventReportCallback (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen, pack_pds_SetEventReportCallback_t *pReqParam)
- int unpack_pds_SetEventReportCallback (uint8_t *pResp, uint16_t respLen)
- 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 *pOutput)

### 9.7.1 Function Documentation

#### 9.7.1.1 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

#### 9.7.1.2 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

#### 9.7.1.3 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

#### 9.7.1.4 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

#### 9.7.1.5 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

#### 9.7.1.6 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

#### 9.7.1.7 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

#### 9.7.1.8 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

#### 9.7.1.9 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

#### 9.7.1.10 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

#### 9.7.1.11 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

#### 9.7.1.12 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

#### 9.7.1.13 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

### 9.7.1.14 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

### 9.7.1.15 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

### 9.7.1.16 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

#### 9.7.1.17 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

#### 9.7.1.18 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

#### 9.7.1.19 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

#### 9.7.1.20 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

#### 9.7.1.21 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

#### 9.7.1.22 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

#### 9.7.1.23 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

#### 9.7.1.24 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

### 9.7.1.25 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

### 9.7.1.26 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

### 9.7.1.27 int unpack_pds_ForceXTRADownload ( uint8_t * *pResp*, uint16_t *respLen* )

Forces the XTRA database to be downloaded to the device 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

#### 9.7.1.28 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

#### 9.7.1.29 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

#### 9.7.1.30 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

#### 9.7.1.31 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

#### 9.7.1.32 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

#### 9.7.1.33 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

#### 9.7.1.34 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

#### 9.7.1.35 int unpack_pds_PDSInjectTimeReference ( uint8_t * *pResp*, uint16_t *respLen* )

Inject system time 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

#### 9.7.1.36 int unpack_pds_ResetPDSData ( uint8_t * *pResp*, uint16_t *respLen* )

Resets the specified PDS data 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

#### 9.7.1.37 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

#### 9.7.1.38 int unpack_pds_SetEventReportCallback ( uint8_t * *pResp*, uint16_t *respLen* )

Enables/disables the PDS Event report callback function 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

#### 9.7.1.39 int unpack_pds_SetPDSDefaults ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the default tracking session 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

#### 9.7.1.40 int unpack_pds_SetPDSState ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the PDS 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

#### 9.7.1.41 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

#### 9.7.1.42 int unpack_pds_SetPortAutomaticTracking ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the automatic tracking configuration for the NMEA COM port 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

#### 9.7.1.43 int unpack_pds_SetServiceAutomaticTracking ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the automatic tracking state for the service 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

#### 9.7.1.44 int unpack_pds_SetXTRAAutomaticDownload ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the XTRA automatic database download 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

#### 9.7.1.45 int unpack_pds_SetXTRANetwork ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the XTRA WWAN network 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

#### 9.7.1.46 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

#### 9.7.1.47 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

#### 9.7.1.48 int unpack_pds_SLQSPDSDeterminePosition ( uint8_t * *pResp*, uint16_t *respLen* )

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 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.1.49 int unpack_pds_SLQSPDSInjectAbsoluteTimeReference ( uint8_t * *pResp*, uint16_t *respLen* )

Injects a absolute time reference into the PDS engine 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

#### 9.7.1.50 int unpack_pds_SLQSPDSInjectPositionData ( uint8_t * *pResp*, uint16_t *respLen* )

Injects position data into the PDS engine 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

#### 9.7.1.51 int unpack_pds_SLQSSetAGPSConfig ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the PDS AGPS (MS-based) 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

#### 9.7.1.52 int unpack_pds_SLQSSetPositionMethodState ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the state of positioning methods for the device 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

#### 9.7.1.53 int unpack_pds_StartPDSTrackingSessionExt ( uint8_t * *pResp*, uint16_t *respLen* )

Starts a PDS tracking 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

#### 9.7.1.54 int unpack_pds_StopPDSTrackingSession ( uint8_t * *pResp*, uint16_t *respLen* )

Stops a PDS tracking 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

## 9.8 qaCbkCatEventReportInd.h File Reference

### Data Structures

- struct CatEventIDDataTlv
- struct CatAlPhaIdentifierTlv
- struct CatEventListTlv
- struct CatRefreshTlv
- struct CatEndProactiveSessionTlv
- union currentCatEvent
- struct CatCommonEventTlv
- struct QmiCbkCatEventStatusReportInd

### Macros

- #define QMI_MAX_CAT_EVENT_DATA_LENGTH 255
- #define QMI_CAN_COMMON_EVENT_TLV_NUMBER 11

### Enumerations

- enum eQMI_CAT_EVENT_REPORT_IND_TLV {
  eTLV_CBK_DISPLAY_TEXT = 0x10,
  eTLV_CBK_GET_IN_KEY = 0x11,
  eTLV_CBK_GET_INPUT = 0x12,
  eTLV_CBK_SETUP_MENU = 0x13,
  eTLV_CBK_SELECT_ITEM = 0x14,
  eTLV_CBK_ALPHA_IDENTIFIER = 0x15,
  eTLV_CBK_SETUP_EVENT_LIST = 0x16,
  eTLV_CBK_SETUP_IDLE_MODE_TEXT = 0x17,
  eTLV_CBK_LANGUAGE_NOTIFICATION = 0x18,
  eTLV_CBK_REFRESH = 0x19,
  eTLV_CBK_END_PROACTIVE_SESSION = 0x1A }
- enum eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH {
  eTLV_SETUP_EVENT_LIST_LENGTH = 0x04,
  eTLV_REFRESH_LENGTH = 0x03,
  eTLV_END_PROACTIVE_SESSION_LENGTH = 0x01 }

### Functions

- enum eQCWWANError UpkQmiCbkCatEventReportInd (BYTE *pMdmResp, struct QmiCbkCatEventStatus-ReportInd *pAipResp)

### 9.8.1 Macro Definition Documentation

#### 9.8.1.1 #define QMI_CAN_COMMON_EVENT_TLV_NUMBER 11

#### 9.8.1.2 #define QMI_MAX_CAT_EVENT_DATA_LENGTH 255

### 9.8.2 Enumeration Type Documentation

#### 9.8.2.1 enum eQMI_CAT_EVENT_REPORT_IND_TLV

Enumerator

***eTLV_CBK_DISPLAY_TEXT***

***eTLV_CBK_GET_IN_KEY***

***eTLV_CBK_GET_INPUT***

***eTLV_CBK_SETUP_MENU***

***eTLV_CBK_SELECT_ITEM***

***eTLV_CBK_ALPHA_IDENTIFIER***

***eTLV_CBK_SETUP_EVENT_LIST***

***eTLV_CBK_SETUP_IDLE_MODE_TEXT***

***eTLV_CBK_LANGUAGE_NOTIFICATION***

***eTLV_CBK_REFRESH***

***eTLV_CBK_END_PROACTIVE_SESSION***

#### 9.8.2.2 enum eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH

Enumerator

***eTLV_SETUP_EVENT_LIST_LENGTH***

***eTLV_REFRESH_LENGTH***

***eTLV_END_PROACTIVE_SESSION_LENGTH***

### 9.8.3 Function Documentation

#### 9.8.3.1 enum eQCWWANError UpkQmiCbkCatEventReportInd ( BYTE * *pMdmResp*, struct QmiCbkCatEventStatusReportInd * *pAipResp* )

## 9.9 qaCbkSwiOmaDmEventReportInd.h File Reference

### Data Structures

- struct sessionInfoTlv
- struct sessionInfoTlvExt
- struct QmiCbkSwiOmaDmEventStatusReportInd
- struct QmiCbkSwiOmaDmEventStatusReportIndExt

### Macros

- #define QMI_SWIOMA_DM_FOTA 0x00
- #define QMI_SWIOMA_DM_CONFIG 0x01
- #define QMI_SWIOMA_DM_NOT 0x02

### Enumerations

- enum eQMI_SWIOMA_DM_EVENT_REPORT_IND {
  eTLV_IND_OMA_DM_FOTA = 0x10,
  eTLV_IND_OMA_DM_CONFIG = 0x11,
  eTLV_IND_OMA_DM_NOT = 0x12 }

### Functions

- enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportInd (BYTE *pMdmResp, struct QmiCbkSwiOmaDmEventStatusReportInd *pApiResp)
- package enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportIndExt (BYTE *pMdmResp, struct QmiCbkSwiOmaDmEventStatusReportInd *pApiResp)

#### 9.9.1 Macro Definition Documentation

##### 9.9.1.1 #define QMI_SWIOMA_DM_CONFIG 0x01

##### 9.9.1.2 #define QMI_SWIOMA_DM_FOTA 0x00

##### 9.9.1.3 #define QMI_SWIOMA_DM_NOT 0x02

#### 9.9.2 Enumeration Type Documentation

##### 9.9.2.1 enum eQMI_SWIOMA_DM_EVENT_REPORT_IND

Enumerator

***eTLV_IND_OMA_DM_FOTA***

***eTLV_IND_OMA_DM_CONFIG***

***eTLV_IND_OMA_DM_NOT***

#### 9.9.3 Function Documentation

##### 9.9.3.1 enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportInd ( BYTE * *pMdmResp,* struct QmiCbkSwiOmaDmEventStatusReportInd * *pApiResp* )

##### 9.9.3.2 package enum eQCWWANError UpkQmiCbkSwiOmaDmEventReportIndExt ( BYTE * *pMdmResp,* struct QmiCbkSwiOmaDmEventStatusReportInd * *pApiResp* )

## 9.10 qaGobiApiAudio.h File Reference

Audio Service API function prototypes.

### Data Structures

- struct GetAudioProfileReq
- struct GetAudioProfileResp
- struct SetAudioProfileReq
- struct GetAudioPathConfigReq
- struct TXPCMIIRFltr
- struct RXPCMIIRFltr
- struct RXAGCList

- struct RXAVCList
- struct TXAGCList
- struct GetAudioPathConfigResp
- struct SetAudioPathConfigReq
- struct GetAudioVolTLBConfigReq
- struct GetAudioVolTLBConfigResp
- struct SetAudioVolTLBConfigReq
- struct SetAudioVolTLBConfigResp

## Functions

- ULONG SLQSGetAudioProfile (GetAudioProfileReq *pGetAudioProfileReq, GetAudioProfileResp *pGetAudioProfileResp)
- ULONG SLQSSetAudioProfile (SetAudioProfileReq *pSetAudioProfileReq)
- ULONG SLQSGetAudioPathConfig (GetAudioPathConfigReq *pGetAudioPathConfigReq, GetAudioPathConfigResp *pGetAudioPathConfigResp)
- ULONG SLQSSetAudioPathConfig (SetAudioPathConfigReq *pSetAudioPathConfigReq)
- ULONG SLQSGetAudioVolTLBConfig (GetAudioVolTLBConfigReq *pGetAudioVolTLBCfgReq, GetAudioVolTLBConfigResp *pGetAudioVolTLBCfgResp)
- ULONG SLQSSetAudioVolTLBConfig (SetAudioVolTLBConfigReq *pSetAudioVolTLBCfgReq, SetAudioVolTLBConfigResp *pSetAudioVolTLBCfgResp)

### 9.10.1 Detailed Description

Audio Service API function prototypes.

### 9.10.2 Function Documentation

#### 9.10.2.1 ULONG SLQSGetAudioPathConfig ( GetAudioPathConfigReq * *pGetAudioPathConfigReq*, GetAudioPathConfigResp * *pGetAudioPathConfigResp* )

This API gets the audio path configuration parameters.

**Parameters**

| | |
|---|---|
| *pGetAudioPathConfigReq[IN]* | • See GetAudioPathConfigReq for more information |
| *pGetAudioPathConfigResp[OUT]* | • See GetAudioPathConfigResp 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**

Device Supported: SL9090
Timeout: 5 seconds

#### 9.10.2.2 ULONG SLQSGetAudioProfile ( GetAudioProfileReq * *pGetAudioProfileReq*, GetAudioProfileResp * *pGetAudioProfileResp* )

This API get the profile content of the requested audio generator.

**Parameters**

| | |
|---|---|
| *pGetAudioProfileReq[IN]* | • See GetAudioProfileReq for more information |
| *pGetAudioProfileResp[OUT]* | • See GetAudioProfileResp 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**

Device Supported: SL9090
Timeout: 5 seconds

#### 9.10.2.3 ULONG SLQSGetAudioVolTLBConfig ( GetAudioVolTLBConfigReq * *pGetAudioVolTLBCfgReq*, GetAudioVolTLBConfigResp * *pGetAudioVolTLBCfgResp* )

This API gets the audio path configuration parameters.

**Parameters**

| | |
|---|---|
| *pGetAudioVolTLBCfgReq[IN]* | • See GetAudioVolTLBConfigReq for more information |
| *pGetAudioVolTLBCfgResp[OUT]* | • See GetAudioVolTLBConfigResp 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**

Device Supported: SL9090
Timeout: 5 seconds

#### 9.10.2.4 ULONG SLQSSetAudioPathConfig ( SetAudioPathConfigReq * *pSetAudioPathConfigReq* )

This API sets the audio path configuration parameters.

**Parameters**

| | |
|---|---|
| *pSetAudioPath-ConfigReq[IN]* | • See SetAudioPathConfigReq 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**

Device Supported: SL9090
Timeout: 5 seconds

#### 9.10.2.5 ULONG SLQSSetAudioProfile ( SetAudioProfileReq * *pSetAudioProfileReq* )

This API sets an audio profile.

**Parameters**

| | |
|---|---|
| *pSetAudio-ProfileReq[IN]* | • See SetAudioProfileReq 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**

Device Supported: SL9090
Timeout: 5 seconds

#### 9.10.2.6 ULONG SLQSSetAudioVoITLBConfig ( SetAudioVoITLBConfigReq * *pSetAudioVoITLBCfgReq*, SetAudioVoITLBConfigResp * *pSetAudioVoITLBCfgResp* )

This API sets the audio path configuration parameters.

**Parameters**

| | |
|---|---|
| *pSetAudioVolTL-BCfgReq[IN]* | • See SetAudioVolTLBConfigReq for more information |
| *pSetAudioVolTL-BCfgResp[OUT]* | • See SetAudioVolTLBConfigResp 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**

Device Supported: SL9090
Timeout: 5 seconds

## 9.11 qaGobiApiCat.h File Reference

Card Application Toolkit API function headers.

### Functions

- ULONG CATSendEnvelopeCommand (ULONG cmdID, ULONG dataLen, BYTE *pData)
- ULONG CATSendTerminalResponse (ULONG refID, ULONG dataLen, BYTE *pData)

### 9.11.1 Detailed Description

Card Application Toolkit API function headers.

### 9.11.2 Function Documentation

#### 9.11.2.1 ULONG CATSendEnvelopeCommand ( ULONG *cmdID,* ULONG *dataLen,* BYTE * *pData* )

Sends the envelope command to the device.

**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[IN]* | • Encoded envelope data as defined in ETSI TS 102 223, section 7<br>[Smart Cards: Card Application Toolkit (CAT) – Release 4] |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 2 seconds

#### 9.11.2.2 ULONG CATSendTerminalResponse ( ULONG *refID,* ULONG *dataLen,* BYTE * *pData* )

Sends the terminal response to the device.

**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[IN]* | • 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] |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 2 seconds

## 9.12 qaGobiApiCbk.h File Reference

Callback Service API function prototypes.

### Data Structures

- struct qaQmiInterfaceInfo
- struct slqsSessionStateInfo
- struct slqsWdsEventInfo
- struct TransferStatsDataType
- struct SignalStrengthDataType
- struct SMSMTMessage
- struct SMSTransferRouteMTMessage
- struct SMSMessageMode
- struct SMSEtwsMessage
- struct SMSEtwsPlmn
- struct SMSCAddress
- struct SMSOnIMS
- struct SMSEventInfo_s
- struct CATEventDataType
- struct ServingSystemInfo
- struct RoamingInfo
- struct SLQSSignalStrengthsInformation
- struct SLQSSignalStrengthsIndReq
- struct ECTNum
- struct voiceSUPSNotification
- struct voiceSetAllCallStatusCbkInfo
- struct _transLayerInfoNotification
- struct _transNWRegInfoNotification
- struct UIMStatusChangeInfo
- struct voicePrivacyInfo
- struct voiceDTMFEventInfo
- struct voiceSUPSInfo
- struct nasSysInfo
- struct UniversalTime
- struct nasNetworkTime
- struct omaDmFotaTlv
- struct omaDmFotaTlvExt
- struct omaDmConfigTlvExt
- struct omaDmConfigTlv
- struct omaDmNotificationsTlv
- union sessionInfo
- union sessionInfoExt
- struct SMSMemoryInfo
- struct voiceOTASPStatusInfo
- struct signalInfo
- struct callerIDInfo
- struct callingPartyInfo
- struct calledPartyInfo
- struct redirNumInfo
- struct NSSAudioCtrl
- struct lineCtrlInfo
- struct extDispRecInfo
- struct voiceInfoRec
- struct msgWaitingInfo
- struct QosFlowInfoState
- struct QosFlowInfo
- struct QosEventInfo
- struct nasSigInfo

- struct _modemTempNotification
- struct _packetSrvStatus
- struct HDRPersonalityInd
- struct imsSIPConfigInfo
- struct imsRegMgrConfigInfo
- struct imsSMSConfigInfo
- struct imsUserConfigInfo
- struct imsVoIPConfigInfo
- struct USSDNoWaitIndicationInfo
- struct channelRate
- struct DUNCallInfoInd
- struct SMSAsyncRawSend_s
- struct LteNasReleaseInfo_s
- struct SwiOTAMsg_s
- struct DHCPOption
- struct DHCPOptionList
- struct WdsDHCPv4ClientLeaseInd
- struct QmiCbkLocCradleMountInd
- struct QmiCbkLocEventTimeSyncInd
- struct QmiCbkLocInjectTimeInd
- struct accelAcceptReady_s
- struct gyroAcceptReady_s
- struct accelTempAcceptReady_s
- struct gyroTempAcceptReady_s
- struct QmiCbkLocSensorStreamingInd
- struct QmiCbkLocInjectSensorDataInd
- struct precisionDilution_s
- struct gpsTime_s
- struct sensorDataUsage_s
- struct svUsedforFix_s
- struct QmiCbkLocPositionReportInd
- struct t_Sv
- struct t_sensor
- struct t_gpsTime
- struct QmiCbkLocBestAvailPosInd
- struct imsaRegStatusInfo
- struct imsaSvcStatusInfo
- struct imsaRatStatusInfo
- struct imsaPdpStatusInfo
- struct satelliteInfo
- struct gnssSvInfoNotification
- struct delAssistDataStatus
- struct QmiCbkNasLTECphyCaInfo
- struct RankIndicatorInd
- struct QmiCbkLocInjectUTCTimeInd
- struct QmiCbkLocInjectPositionInd
- struct UIMSlotStatusChangeInfo
- struct QmiCbkLocEngineStateInd
- struct _getResetInfoNotification
- struct _MitigationDevInfo
- struct QmiCbkTmdMitiLvlRptInd
- struct QmiCbkLocSetExtPowerConfigInd
- struct nasTimers
- struct CsgId
- struct PlmnID

- struct nasNetworkReject
- struct _SwiAvmsEventInd
- struct IPv4Info
- struct IPv6Info
- struct urlAddr
- struct QmiCbkLocGetServerInd
- struct dmsPsmCfgParamChange
- struct QmiCbkLocSetServerInd

## Macros

- #define SIGSTRENGTH_THRESHOLD_ARR_SZ 5
- #define QMI_WMS_MAX_PAYLOAD_LENGTH 256
- #define QMI_ETWS_MAX_PAYLOAD_LENGTH 1254 /∗ Qualcomm defined max ∗/
- #define QMI_MAX_VOICE_NUMBER_LENGTH 81
- #define MAX_NO_OF_UUSINFO 20
- #define MAXUSSDLENGTH 182
- #define MAX_NO_OF_CALLS 20
- #define CBK_ENABLE_EVENT 0x01
- #define CBK_DISABLE_EVENT 0x00
- #define CBK_NOCHANGE 0xFF
- #define MAX_NO_OF_APPLICATIONS 10
- #define MAX_NO_OF_SLOTS 5
- #define MAX_NO_OF_FILES 255
- #define MAX_PATH_LENGTH 255
- #define EVENT_MASK_CARD 0x00000001
- #define EVENT_MASK_PHY_SLOT_STATUS 0x00000010
- #define EVENT_MASK_DEREGISTER_ALL 0x00000000
- #define REGISTER_EVENT 0x01
- #define DEREGISTER_EVENT 0x00
- #define FIRST_INSTANCE 0x00
- #define SECOND_INSTANCE 0x01
- #define THIRD_INSTANCE 0x02
- #define INVALID_INSTACNE 0x08
- #define REGISTER_SRV 0x01
- #define DEREGISTER_SRV 0x00
- #define WDS_SRV 0x01
- #define NAS_SRV 0x02
- #define PDS_SRV 0x04
- #define VOICE_SRV 0x08
- #define NUM_OF_SET 0xFF
- #define IPV4 4
- #define IPV6 6
- #define IPV4V6 7
- #define LOC_EVENT_MASK_ENG_STATE 0x00000080
- #define LOC_EVENT_MASK_TIME_SYNC 0x00000800
- #define LOC_EVENT_MASK_INJECT_TIME 0x00000010
- #define LOC_EVENT_MASK_SENSOR_STREAM 0x00000400
- #define LOC_EVENT_POSITION_REPORT 0x00000001
- #define LOC_EVENT_MASK_GNSS_SV_INFO 0x00000002
- #define LOC_EVENT_MASK_NMEA 0x00000004
- #define MAX_NMEA_STR_LEN 201
- #define DHCP_MAX_NUM_OPTIONS 30

- #define DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */
- #define MAX_MITIGATION_DEV_ID_LEN 255
- #define MAX_URL_ADDR_STR_LEN 256
- #define MAX_RADIO_INTERFACE_LIST 255
- #define USSD_DCS_ASCII 0x01 /* ASCII coding scheme */
- #define USSD_DCS_8BIT 0x02 /* 8-bit coding scheme */
- #define USSD_DCS_UCS2 0x03 /* UCS2 coding scheme */

## Typedefs

- typedef void(* tFNSLQSSessionState )(slqsSessionStateInfo *pSessionStateInfo)
- typedef void(* tFNSLQSWDSEvent )(slqsWdsEventInfo *pWdsEventInfo)
- typedef void(* tFNPower )(ULONG operatingMode)
- typedef void(* tFNActivationStatus )(ULONG activationStatus)
- typedef void(* tFNMobileIPStatus )(ULONG mipStatus)
- typedef void(* tFNRoamingIndicator )(ULONG roaming)
- typedef void(* tFNDataCapabilities )(BYTE dataCapsSize, BYTE *pDataCaps)
- typedef void(* tFNSignalStrength )(INT8 signalStrength, ULONG radioInterface)
- typedef void(* tFNRFInfo )(ULONG radioInterface, ULONG activeBandClass, ULONG activeChannel)
- typedef void(* tFNLUReject )(ULONG serviceDomain, ULONG rejectCause)
- typedef void(* tFNNewSMS )(ULONG storageType, ULONG messageIndex)
- typedef enum SMSEventType eSMSEventType
- typedef struct SMSMTMessage SMSMTMessageInfo
- typedef struct SMSTransferRouteMTMessage SMSTransferRouteMTMessageInfo
- typedef struct SMSMessageMode SMSMessageModeInfo
- typedef struct SMSEtwsMessage SMSEtwsMessageInfo
- typedef struct SMSEtwsPlmn SMSEtwsPlmnInfo
- typedef struct SMSCAddress SMSCAddressInfo
- typedef struct SMSOnIMS SMSOnIMSInfo
- typedef struct SMSEventInfo_s SMSEventInfo
- typedef void(* tFNSMSEvents )(SMSEventInfo *pSMSEventInfo)
- typedef void(* tFNNewNMEA )(LPCSTR pNMEA)
- typedef void(* tFNPDSState )(ULONG enabledStatus, ULONG trackingStatus)
- typedef void(* tFNCATEvent )(ULONG eventID, ULONG eventLen, BYTE *pEventData)
- typedef enum device_state_enum eDevState
- typedef void(* tFNDeviceStateChange )(eDevState device_state)
- typedef void(* tFNNet )(ULONG q_depth, BYTE isThrottle, BYTE instanceId)
- typedef void(* tFNFwDldCompletion )(ULONG fwdld_completion_status)
- typedef void(* tFNSLQSOMADMAlert )(ULONG eventType, BYTE *pEventFields)
- typedef void(* tFNOMADMState )(ULONG sessionState, ULONG failureReason)
- typedef void(* tFNServingSystem )(struct ServingSystemInfo *pServingSystem, struct RoamingInfo *pRoamingInfo)
- typedef void(* tFNBandPreference )(ULONGLONG band_pref)
- typedef void(* tFNUSSDRelease )(void)
- typedef void(* tFNUSSDNotification )(ULONG type, BYTE *pNetworkInfo)
- typedef void(* tFNSLQSSignalStrengths )(struct SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo)
- typedef void(* tFNSUPSNotification )(voiceSUPSNotification *pVoiceSUPSNotification)
- typedef void(* tFNSDKTerminated )(BYTE *psReason)
- typedef void(* tFNAllCallStatus )(voiceSetAllCallStatusCbkInfo *pVoiceSetAllCallStatusCbkInfo)
- typedef struct _transLayerInfoNotification transLayerNotification

- typedef void(* tFNtransLayerInfo )(transLayerNotification *pTransLayerNotification)
- typedef struct _transNWRegInfoNotification transNWRegInfoNotification
- typedef void(* tFNtransNWRegInfo )(transNWRegInfoNotification *pTransNWRegInfoNotification)
- typedef void(* tFNSysSelectionPref )(sysSelectPrefInfo *pSysSelectPrefInfo)
- typedef void(* tFNUIMRefresh )(UIMRefreshEvent *pUIMRefreshEvent)
- typedef void(* tFNUIMStatusChangeInfo )(UIMStatusChangeInfo *pUIMStatusChangeInfo)
- typedef void(* tFNPrivacyChange )(voicePrivacyInfo *pVoicePrivacyInfo)
- typedef void(* tFNDTMFEvent )(voiceDTMFEventInfo *pVoiceDTMFEventInfo)
- typedef void(* tFNSUPSInfo )(voiceSUPSInfo *pVoiceSUPSInfo)
- typedef void(* tFNSysInfo )(nasSysInfo *pNasSysInfo)
- typedef void(* tFNNetworkTime )(nasNetworkTime *pNasNetworkTime)
- typedef union sessionInfo sessionInformation
- typedef union sessionInfoExt sessionInformationExt
- typedef void(* tFNMemoryFull )(SMSMemoryInfo *pSMSMemoryFullInfo)
- typedef void(* tFNOTASPStatus )(voiceOTASPStatusInfo *pVoiceOTASPStatusInfo)
- typedef void(* tFNInfoRec )(voiceInfoRec *pVoiceInfoRec)
- typedef void(* tFNMessageWaiting )(msgWaitingInfo *pSMSMessageWaitingInfo)
- typedef void(* tFNSLQSQOSEvent )(BYTE instance, QosFlowInfo *pFlowInfo)
- typedef void(* tFNQosStatus )(BYTE instance, ULONG id, BYTE status, BYTE event, BYTE reason)
- typedef void(* tFNQosNWStatus )(BYTE status)
- typedef void(* tFNQosPriEvent )(WORD event)
- typedef void(* tFNSigInfo )(nasSigInfo *pNasSigInfo)
- typedef struct _modemTempNotification modemTempNotification
- typedef void(* tFNModemTempInfo )(modemTempNotification *pModemTempNotification)
- typedef struct _packetSrvStatus packetSrvStatus
- typedef void(* tFNPacketSrvState )(packetSrvStatus *pPacketSrvStatus)
- typedef void(* tFNHDRPersonaity )(HDRPersonalityInd *pHDRPers)
- typedef void(* tFNImsSIPConfig )(imsSIPConfigInfo *pImsSIPConfigInfo)
- typedef void(* tFNImsRegMgrConfig )(imsRegMgrConfigInfo *pImsRegMgrConfigInfo)
- typedef void(* tFNImsSMSConfig )(imsSMSConfigInfo *pImsSMSConfigInfo)
- typedef void(* tFNImsUserConfig )(imsUserConfigInfo *pImsUserConfigInfo)
- typedef void(* tFNImsVoIPConfig )(imsVoIPConfigInfo *pImsVoIPConfigInfo)
- typedef void(* tFNUSSDNoWaitIndication )(USSDNoWaitIndicationInfo *pNetworkInfo)
- typedef void(* tFNDUNCallInfo )(DUNCallInfoInd *pDUNCallInfo)
- typedef void(* tFNDataSysStatus )(CurrDataSysStat *pCurrDataSysStat)
- typedef struct SMSAsyncRawSend_s SMSAsyncRawSend
- typedef void(* tFNAsyncRawSend )(SMSAsyncRawSend *pSMSAsyncRawSend)
- typedef struct LteNasReleaseInfo_s LteNasReleaseInfo
- typedef struct SwiOTAMsg_s SwiOTAMsg
- typedef void(* tFNASwiOTAMsg )(SwiOTAMsg *pSwiOTAMsg)
- typedef void(* tFNNewGPS )(double dLongitude, double dLatitude, BYTE session_status, ULONG pos_src)
- typedef void(* tFNNewRMTransferStatistics )(QmiCbkWdsStatisticsIndState *pMsg)
- typedef void(* tFNDHCPv4ClientLeaseStatus )(BYTE instance, WdsDHCPv4ClientLeaseInd *pMsg)
- typedef void(* tFNSetCradleMount )(QmiCbkLocCradleMountInd *pSetLocCradleMount)
- typedef void(* tFNSetEventTimeSync )(QmiCbkLocEventTimeSyncInd *pSetLocEventTimeSync)
- typedef void(* tFNInjectTimeStatus )(QmiCbkLocInjectTimeInd *pLocInjectTime)
- typedef struct accelAcceptReady_s accelAcceptReady
- typedef struct gyroAcceptReady_s gyroAcceptReady
- typedef struct accelTempAcceptReady_s accelTempAcceptReady
- typedef struct gyroTempAcceptReady_s gyroTempAcceptReady
- typedef void(* tFNSensorStreaming )(QmiCbkLocSensorStreamingInd *pLocSensorStream)

- typedef void(∗ tFNInjectSensorData )(QmiCbkLocInjectSensorDataInd ∗pLocInjectSensorData)
- typedef struct precisionDilution_s precisionDilution
- typedef struct gpsTime_s gpsTime
- typedef struct sensorDataUsage_s sensorDataUsage
- typedef struct svUsedforFix_s svUsedforFix
- typedef void(∗ tFNEventPosition )(QmiCbkLocPositionReportInd ∗pLocPositionReport)
- typedef void(∗ tFNBestAvailPos )(QmiCbkLocBestAvailPosInd ∗pBestAvailPos)
- typedef void(∗ tFNOpMode )(ULONG mode)
- typedef void(∗ tFNImsaRegStatus )(imsaRegStatusInfo ∗pImsaRegStatusInfo)
- typedef void(∗ tFNImsaSvcStatus )(imsaSvcStatusInfo ∗pImsaSvcStatusInfo)
- typedef void(∗ tFNImsaRatStatus )(imsaRatStatusInfo ∗pImsaRatStatusInfo)
- typedef void(∗ tFNImsaPdpStatus )(imsaPdpStatusInfo ∗pImsaPdpStatusInfo)
- typedef void(∗ tFNGnssSvInfo )(gnssSvInfoNotification ∗pGnssSvInfoNotification)
- typedef void(∗ tFNDelAssistData )(delAssistDataStatus ∗pAssistDataNotification)
- typedef void(∗ tFNASwiLTECphyCaInfo )(QmiCbkNasLTECphyCaInfo ∗pQmiCbkNasLTECphyCaInfo)
- typedef void(∗ tFNRankIndicator )(RankIndicatorInd ∗pRankIndicatorInd)
- typedef void(∗ tFNInjectUTCTime )(QmiCbkLocInjectUTCTimeInd ∗pInjectUTCTimeNotification)
- typedef void(∗ tFNInjectPosition )(QmiCbkLocInjectPositionInd ∗pInjectPositionNotification)
- typedef void(∗ tFNCbkUimSlotStatusChangeInd )(UIMSlotStatusChangeInfo ∗pQmiCbkUimSlotStatus-ChangeInd)
- typedef void(∗ tFNSetEngineState )(QmiCbkLocEngineStateInd ∗pSetLocEngineState)
- typedef struct _getResetInfoNotification ResetInfoNotification
- typedef void(∗ tFNResetInfo )(ResetInfoNotification ∗pResetInfoNotification)
- typedef void(∗ tFNMitiLvlRpt )(QmiCbkTmdMitiLvlRptInd ∗pSetLocCradleMount)
- typedef void(∗ tFNSetExtPowerConfig )(QmiCbkLocSetExtPowerConfigInd ∗pSetExtConfigIndStatus)
- typedef void(∗ tFNNasTimer )(nasTimers ∗timers)
- typedef void(∗ tFNNetworkReject )(nasNetworkReject ∗pNetworkReject)
- typedef struct _SwiAvmsEventInd SwiAvmsEventInd
- typedef void(∗ tFNSwiAvmEventReport )(SwiAvmsEventInd ∗pEvent)
- typedef void(∗ tFNEventNmea )(CHAR ∗pNMEAData)
- typedef void(∗ tFNGetServer )(QmiCbkLocGetServerInd ∗pGetLocServerInfo)
- typedef void(∗ tFNPSMCfgParamChange )(dmsPsmCfgParamChange ∗pPSMCfgParamInd)
- typedef void(∗ tFNSetServer )(QmiCbkLocSetServerInd ∗pSetLocServerInfo)

## Enumerations

- enum eQaQMIService {
  eQA_QMI_SVC_WDS = 0x01,
  eQA_QMI_SVC_NAS = 0x03,
  eQA_QMI_SVC_NA = 0xFF }
- enum SMSEventType {
  SMS_EVENT_MT_MESSAGE = 0x01,
  SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE = 0x02,
  SMS_EVENT_MESSAGE_MODE = 0x04,
  SMS_EVENT_ETWS = 0x08,
  SMS_EVENT_ETWS_PLMN = 0x10,
  SMS_EVENT_SMSC_ADDRESS = 0x20,
  SMS_EVENT_SMS_ON_IMS = 0x40 }
- enum device_state_enum {
  DEVICE_STATE_DISCONNECTED,
  DEVICE_STATE_READY,
  DEVICE_STATE_BOOT,
  DEVICE_STATE_SERIAL_ONLY }

- enum TYPE_OF_SERVICE_DOMAIN {
  SYS_SRV_DOMAIN_NO_SRV = 0x00,
  SYS_SRV_DOMAIN_CS_ONLY = 0x01,
  SYS_SRV_DOMAIN_PS_ONLY = 0x02,
  SYS_SRV_DOMAIN_CS_PS = 0x03,
  SYS_SRV_DOMAIN_CAMPED = 0x04 }

## Functions

- ULONG SLQSSetSessionStateCallback (tFNSLQSSessionState pCallback)
- ULONG SLQSSetWdsEventCallback (tFNSLQSWDSEvent pCallback, BYTE interval, BYTE instanceid, BYTE ipfamily)
- ULONG SLQSSetWdsTransferStatisticCallback (tFNSLQSWDSEvent pXferStatsCb, BYTE interval, BYTE instanceid, BYTE ipfamily)
- ULONG iSLQSSetWdsFirstInstEventCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsSecondInstEventCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsThirdInstEventCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsXferStatsFirstInstCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsXferStatsSecondInstCallback (tFNSLQSWDSEvent pCallback)
- ULONG SetPowerCallback (tFNPower pCallback)
- ULONG SetActivationStatusCallback (tFNActivationStatus pCallback)
- ULONG SetMobileIPStatusCallback (tFNMobileIPStatus pCallback)
- ULONG SetRoamingIndicatorCallback (tFNRoamingIndicator pCallback)
- ULONG SetDataCapabilitiesCallback (tFNDataCapabilities pCallback)
- ULONG SetSignalStrengthCallback (tFNSignalStrength pCallback, BYTE thresholdsSize, INT8 *pThresholds)
- ULONG iSetSignalStrengthCallback (tFNSignalStrength pCallback)
- ULONG SetRFInfoCallback (tFNRFInfo pCallback)
- ULONG SetLURejectCallback (tFNLUReject pCallback)
- ULONG SetNewSMSCallback (tFNNewSMS pCallback)
- ULONG SLQSSetSMSEventCallback (tFNSMSEvents pCallback)
- ULONG SetNMEACallback (tFNNewNMEA pCallback)
- ULONG SetPDSStateCallback (tFNPDSState pCallback)
- ULONG SetCATEventCallback (tFNCATEvent pCallback, ULONG eventMask, ULONG *pErrorMask)
- ULONG iSetCATEventCallback (tFNCATEvent pCallback)
- ULONG SetDeviceStateChangeCbk (tFNDeviceStateChange pCallback)
- ULONG SetNetChangeCbk (BYTE instance, tFNNet pCallback, ULONG loMark, ULONG hiMark, ULONG period)
- ULONG SetFwDldCompletionCbk (tFNFwDldCompletion pCallback)
- ULONG SetSLQSOMADMAlertCallback (tFNSLQSOMADMAlert pCallback)
- ULONG SetSLQSOMADMAlertCallbackExt (tFNSLQSOMADMAlert pCallback)
- ULONG SetOMADMStateCallback (tFNOMADMState pCallback)
- ULONG SLQSSetServingSystemCallback (tFNServingSystem pCallback)
- ULONG SLQSSetBandPreferenceCbk (tFNBandPreference pCallback)
- ULONG SetUSSDReleaseCallback (tFNUSSDRelease pCallback)
- ULONG SetUSSDNotificationCallback (tFNUSSDNotification pCallback)
- ULONG SLQSSetSignalStrengthsCallback (tFNSLQSSignalStrengths pCallback, struct SLQSSignalStrengthsIndReq *pSLQSSignalStrengthsIndReq)
- ULONG iSLQSSetSignalStrengthsCallback (tFNSLQSSignalStrengths pCallback)
- ULONG SLQSVoiceSetSUPSNotificationCallback (tFNSUPSNotification pCallback)
- ULONG SLQSSetSDKTerminatedCallback (tFNSDKTerminated pCallback)
- ULONG SLQSVoiceSetAllCallStatusCallBack (tFNAllCallStatus pCallback)
- ULONG SLQSSetTransLayerInfoCallback (tFNtransLayerInfo pCallback)
- ULONG SLQSSetTransNWRegInfoCallback (tFNtransNWRegInfo pCallback)
- ULONG SLQSSetSysSelectionPrefCallBack (tFNSysSelectionPref pCallback)

- ULONG SLQSUIMSetRefreshCallBack (tFNUIMRefresh pCallback)
- ULONG SLQSUIMSetStatusChangeCallBack (tFNUIMStatusChangeInfo pCallback)
- ULONG SLQSVoiceSetPrivacyChangeCallBack (tFNPrivacyChange pCallback)
- ULONG SLQSVoiceSetDTMFEventCallBack (tFNDTMFEvent pCallback)
- ULONG SLQSVoiceSetSUPSCallBack (tFNSUPSInfo pCallback)
- ULONG SLQSNasSysInfoCallBack (tFNSysInfo pCallback)
- ULONG SLQSNasNetworkTimeCallBack (tFNNetworkTime pCallback)
- ULONG SLQSWmsMemoryFullCallBack (tFNMemoryFull pCallback)
- ULONG SLQSVoiceSetOTASPStatusCallBack (tFNOTASPStatus pCallback)
- ULONG SLQSVoiceInfoRecCallback (tFNInfoRec pCallback)
- ULONG SLQSWmsMessageWaitingCallBack (tFNMessageWaiting pCallback)
- ULONG SLQSSetQosEventCallback (BYTE instance, tFNSLQSQOSEvent pCallback)
- ULONG SLQSSetQosStatusCallback (BYTE instance, tFNQosStatus pCallback)
- ULONG SLQSSetQosNWStatusCallback (tFNQosNWStatus pCallback)
- ULONG SLQSSetQosPriEventCallback (tFNQosPriEvent pCallback)
- ULONG SLQSNasSigInfoCallBack (tFNSigInfo pCallback, sigInfo *pSigInfo)
- ULONG SLQSSetModemTempCallback (tFNModemTempInfo pCallback)
- ULONG SLQSSetPacketSrvStatusCallback (tFNPacketSrvState pCallback)
- ULONG SLQSSetSwiHDRPersCallback (tFNHDRPersonaity pCallback)
- ULONG SLQSSetSIPConfigCallback (tFNImsSIPConfig pCallback)
- ULONG SLQSSetRegMgrConfigCallback (tFNImsRegMgrConfig pCallback)
- ULONG SLQSSetIMSSMSConfigCallback (tFNImsSMSConfig pCallback)
- ULONG SLQSSetIMSUserConfigCallback (tFNImsUserConfig pCallback)
- ULONG SLQSSetIMSVoIPConfigCallback (tFNImsVoIPConfig pCallback)
- ULONG SetUSSDNoWaitIndicationCallback (tFNUSSDNoWaitIndication pCallback)
- ULONG SLQSSetDUNCallInfoCallback (BYTE StatsPeriod, tFNDUNCallInfo pCallback)
- ULONG iSLQSSetDUNCallInfoCallback (tFNDUNCallInfo pCallback)
- ULONG SLQSSetDataSystemStatusCallback (tFNDataSysStatus pCallback)
- ULONG SLQSWmsAsyncRawSendCallBack (tFNAsyncRawSend pCallback)
- ULONG SLQSNasSwiOTAMessageCallback (NasSwiIndReg *req, tFNASwiOTAMsg pCallback)
- ULONG SetGPSCallback (tFNNewGPS pCallback)
- ULONG SetRMTransferStatisticsCallback (tFNNewRMTransferStatistics pCallback)
- ULONG SLQSSetDHCPv4ClientLeaseStatusCallback (BYTE instance, tFNDHCPv4ClientLeaseStatus pCallback)
- ULONG SetLocCradleMountCallback (tFNSetCradleMount pCallback)
- ULONG SetLocEventTimeSyncCallback (tFNSetEventTimeSync pCallback)
- ULONG SetLocInjectTimeCallback (tFNInjectTimeStatus pCallback)
- ULONG SetLocSensorStreamingCallback (tFNSensorStreaming pCallback)
- ULONG SetLocInjectSensorDataCallback (tFNInjectSensorData pCallback)
- ULONG SetLocEventPositionCallback (tFNEventPosition pCallback)
- ULONG SetLocOpModeCallback (tFNOpMode pCallback)
- ULONG SLQSSetIMSARegStatusCallback (tFNImsaRegStatus pCallback)
- ULONG SLQSSetIMSASvcStatusCallback (tFNImsaSvcStatus pCallback)
- ULONG SLQSSetIMSARatStatusCallback (tFNImsaRatStatus pCallback)
- ULONG SLQSSetIMSAPdpStatusCallback (tFNImsaPdpStatus pCallback)
- ULONG SLQSNasSigInfo2CallBack (tFNSigInfo pCallback, setSignalStrengthInfo *pSigInfo2)
- ULONG SetLocGnssSvInfoCallback (tFNGnssSvInfo pCallback)
- ULONG SetLocDeleteAssistDataCallback (tFNDelAssistData pCallback)
- ULONG SetNasLTECphyCaIndCallback (tFNASwiLTECphyCaInfo pCallback)
- ULONG SetRankIndicatorCallback (tFNRankIndicator pCallback)
- ULONG SLQSSetLocInjectUTCTimeCallback (tFNInjectUTCTime pCallback)
- ULONG SLQSSetLocInjectPositionCallback (tFNInjectPosition pCallback)
- ULONG SetUimSlotStatusChangeCallback (tFNCbkUimSlotStatusChangeInd pCallback)
- ULONG SetLocEngineStateCallback (tFNSetEngineState pCallback)
- ULONG SLQSSetSwiGetResetInfoCallback (tFNResetInfo pCallback)

- ULONG SLQSTmdMitigationLvlRptCallback (TmdMitigationLvlIndReq *req, tFNMitiLvlRpt pCallback)
- ULONG SetLocSetExtPowerConfigCallback (tFNSetExtPowerConfig pCallback)
- ULONG SetLocBestAvailPosCallback (tFNBestAvailPos pCallback)
- ULONG SLQSNasTimerCallback (tFNNasTimer pCallback)
- ULONG SLQSNasNetworkRejectCallback (tFNNetworkReject pCallback)
- ULONG SLQSSwiAVMSEventReportCallback (tFNSwiAvmEventReport pCallback)
- ULONG SetLocEventMaskNMEACallback (tFNEventNmea pCallback)
- ULONG SetLocGetServerCallback (LocGetServerReq *req, tFNGetServer pCallback)
- ULONG SetCfgParamChangeCallback (tFNPSMCfgParamChange pCallback)
- ULONG SetLocSetServerCallback (LocSetServerReq *req, tFNSetServer pCallback)

### 9.12.1 Detailed Description

Callback Service API function prototypes.

### 9.12.2 Macro Definition Documentation

#### 9.12.2.1 #define CBK_DISABLE_EVENT 0x00

#### 9.12.2.2 #define CBK_ENABLE_EVENT 0x01

#### 9.12.2.3 #define CBK_NOCHANGE 0xFF

#### 9.12.2.4 #define DEREGISTER_EVENT 0x00

#### 9.12.2.5 #define DEREGISTER_SRV 0x00

#### 9.12.2.6 #define DHCP_MAX_NUM_OPTIONS 30

#### 9.12.2.7 #define DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */

#### 9.12.2.8 #define EVENT_MASK_CARD 0x00000001

#### 9.12.2.9 #define EVENT_MASK_DEREGISTER_ALL 0x00000000

#### 9.12.2.10 #define EVENT_MASK_PHY_SLOT_STATUS 0x00000010

#### 9.12.2.11 #define FIRST_INSTANCE 0x00

#### 9.12.2.12 #define INVALID_INSTACNE 0x08

#### 9.12.2.13 #define IPV4 4

#### 9.12.2.14 #define IPV4V6 7

#### 9.12.2.15 #define IPV6 6

#### 9.12.2.16 #define LOC_EVENT_MASK_ENG_STATE 0x00000080

#### 9.12.2.17 #define LOC_EVENT_MASK_GNSS_SV_INFO 0x00000002

#### 9.12.2.18 #define LOC_EVENT_MASK_INJECT_TIME 0x00000010

#### 9.12.2.19 #define LOC_EVENT_MASK_NMEA 0x00000004

9.12.2.20 #define LOC_EVENT_MASK_SENSOR_STREAM 0x00000400

9.12.2.21 #define LOC_EVENT_MASK_TIME_SYNC 0x00000800

9.12.2.22 #define LOC_EVENT_POSITION_REPORT 0x00000001

9.12.2.23 #define MAX_MITIGATION_DEV_ID_LEN 255

9.12.2.24 #define MAX_NMEA_STR_LEN 201

9.12.2.25 #define MAX_NO_OF_APPLICATIONS 10

9.12.2.26 #define MAX_NO_OF_CALLS 20

9.12.2.27 #define MAX_NO_OF_FILES 255

9.12.2.28 #define MAX_NO_OF_SLOTS 5

9.12.2.29 #define MAX_NO_OF_UUSINFO 20

9.12.2.30 #define MAX_PATH_LENGTH 255

9.12.2.31 #define MAX_RADIO_INTERFACE_LIST 255

9.12.2.32 #define MAX_URL_ADDR_STR_LEN 256

9.12.2.33 #define MAXUSSDLENGTH 182

9.12.2.34 #define NAS_SRV 0x02

9.12.2.35 #define NUM_OF_SET 0xFF

9.12.2.36 #define PDS_SRV 0x04

9.12.2.37 #define QMI_ETWS_MAX_PAYLOAD_LENGTH 1254 /* Qualcomm defined max */

9.12.2.38 #define QMI_MAX_VOICE_NUMBER_LENGTH 81

9.12.2.39 #define QMI_WMS_MAX_PAYLOAD_LENGTH 256

9.12.2.40 #define REGISTER_EVENT 0x01

9.12.2.41 #define REGISTER_SRV 0x01

9.12.2.42 #define SECOND_INSTANCE 0x01

9.12.2.43 #define SIGSTRENGTH_THRESHOLD_ARR_SZ 5

9.12.2.44 #define THIRD_INSTANCE 0x02

9.12.2.45 #define USSD_DCS_8BIT 0x02 /* 8-bit coding scheme */

9.12.2.46 #define USSD_DCS_ASCII 0x01 /* ASCII coding scheme */

9.12.2.47 #define USSD_DCS_UCS2 0x03 /* UCS2 coding scheme */

#### 9.12.2.48 #define VOICE_SRV 0x08

#### 9.12.2.49 #define WDS_SRV 0x01

### 9.12.3 Typedef Documentation

#### 9.12.3.1 typedef struct accelAcceptReady_s accelAcceptReady

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. |

#### 9.12.3.2 typedef struct accelTempAcceptReady_s accelTempAcceptReady

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. |

#### 9.12.3.3 typedef enum device_state_enum eDevState

Device State enumeration

- See device_state_enum for more details

#### 9.12.3.4 typedef enum SMSEventType eSMSEventType

This enumeration defines the different type of SMS events that are received

- See SMSEventType for more details

#### 9.12.3.5 typedef struct gpsTime_s gpsTime

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 |

#### 9.12.3.6 typedef struct gyroAcceptReady_s gyroAcceptReady

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. |

#### 9.12.3.7 typedef struct gyroTempAcceptReady_s gyroTempAcceptReady

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. |

#### 9.12.3.8 typedef struct LteNasReleaseInfo_s LteNasReleaseInfo

This structure contains LTE Nas Release Infomation

**Parameters**

| | |
|---|---|
| *nas_release* | • LTE NAS release |
| *nas_major* | • LTE NAS version major |
| *nas_minor* | • LTE NAS version minor |

#### 9.12.3.9 typedef struct _modemTempNotification modemTempNotification

Contains the parameters passed for SLQSSetModemTempCallback by the device.

**Parameters**

| | |
|---|---|
| *ModemTemp-State* | • provides the temperature state of the modem<br>• Values:<br>– 0 - unknown<br>– 1 - normal<br>– 2 - high(warning)<br>– 3 - high(critical)<br>– 4 - low(critical) |

| | |
|---|---|
| *Modem-Temperature* | • provides the temperature of the modem |

**Note**

None

#### 9.12.3.10 typedef struct _packetSrvStatus packetSrvStatus

Contains the parameters passed for SLQSSetPacketSrvStatusCallback by the device.

**Parameters**

| | |
|---|---|
| *pQmiInterface-Info* | • See qaQmiInterfaceInfo for more information |
| *connStatus* | • Current Link Status<br>– 1 - Disconnected<br>– 2 - Connected<br>– 3 - Suspended<br>– 4 - Authenticating |
| *reconfigReqd* | • Indicates if the network interface on the host needs to be reconfigured<br>– 0 - No need to reconfigure<br>– 1 - Reconfiguration required |
| *sessionEnd-Reason* | • See qaGobiApiTableCallEndReasons.h for Call End Reason, 0xFFFF means invalid value |
| *verboseSessn-EndReasonType* | • Call End Reason Type<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 value |
| *verboseSessn-EndReason* | • See qaGobiApiTableCallEndReasons.h for verbose Call End Reason. The values depend on verboseSessnEndReasonType parameter 0xFFFF means invalid value |

| | |
|---|---|
| *ipFamily* | • IP Family of the packet data connection<br>– 4 - IPv4<br>– 6 - IPv6<br>– 0xFF - invalid value |
| *techName* | • Technology name of the packet data connection.<br>– 32767 - CDMA<br>– 32764 - UMTS<br>– 30592 - EPC<br>– 30590 - EMBMS<br>– 30584 - Modem Link Local<br>– 0xFFFF - invalid value EPC is a logical interface to support LTE/eHRPD handoff. Modem Link is an interface for transferring data between entities on the AP and modem. |
| *bearerID* | • Bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection 0xFF means invalid value |

**Note**

Any parameter not returned by the device is returned as its maximum unsigned value by the callback.

#### 9.12.3.11 typedef struct precisionDilution_s precisionDilution

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) |

#### 9.12.3.12 typedef struct _getResetInfoNotification ResetInfoNotification

Contains the parameters passed for SLQSSetSwiGetResetInfoCallback by the device.

**Parameters**

| | |
|---|---|
| *type* | • 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 |
| *source* | • entity which initiated the reset or power down, possible values listed below:<br>– 0 - unknown<br>– 1 - user requested<br>– 2 - hardware switch<br>– 3 - temperature critical<br>– 4 - voltage critical<br>– 5 - configuration update<br>– 6 - LWM2M<br>– 7 - OMA-DM<br>– 8 - FOTA |

**Note**

None

#### 9.12.3.13 typedef struct sensorDataUsage_s sensorDataUsage

This structure contains Sensor Data Usage info.

**Parameters**

| | |
|---|---|
| *usageMask* | • Specifies which sensors were used in calculating the position in the position report. |

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

**Parameters**

| | |
|---|---|
| *aidingIndicator-Mask* | |

  - Specifies which results were aided by sensors.

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

#### 9.12.3.14 typedef union sessionInfo sessionInformation

This union sessionInfo consist of omaDmFotaTlv, omaDmConfigTlv and omaDmNotificationsTlv, out of which one will be unpacked against pEventFields.

#### 9.12.3.15 typedef union sessionInfoExt sessionInformationExt

This union sessionInfo consist of omaDmFotaTlv and omaDmConfigTlv, out of which one will be unpacked against pEventFields.

#### 9.12.3.16 typedef struct SMSAsyncRawSend_s SMSAsyncRawSend

This structure contains SMS parameters

**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 |
| *messageID* | • Unique ID assigned by WMS for non-retry messages. |
| *causeCode* | • WMS cause code |
| *errorClass* | • Error Class<br>• Values:<br>– 0x00 - ERROR_CLASS_TEMPORARY<br>– 0x01 - ERROR_CLASS_PERMANENT |
| *RPCause* | • GW RP cause |
| *TPCause* | • GW TP Cause |
| *msgDelFailure-Type* | • Message delivery failure type<br>• Values:<br>– 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY<br>– 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT |
| *msgDelFailure-Cause* | • Message delivery failure cause<br>• Values:<br>– 0x00 - WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL |

| | |
|---|---|
| *alphaIDLen* | • Number of sets of the pAlphaID |
| *pAlphaID* | • Alpha ID |
| *userData* | • Identifies the request associated with this indication. |

#### 9.12.3.17 typedef struct SMSCAddress SMSCAddressInfo

This structure holds SMSC information

**Parameters**

| | |
|---|---|
| *length* | • Number of sets of following element |
| *data* | • SMSC address |

#### 9.12.3.18 typedef struct SMSEtwsMessage SMSEtwsMessageInfo

This structure holds information related earthquake and Tsunami warning system

**Parameters**

| | |
|---|---|
| *notificationType* | • Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS |
| *length* | • Number of sets of following elements |
| *data* | • Raw message data |

#### 9.12.3.19 typedef struct SMSEtwsPlmn SMSEtwsPlmnInfo

This structure holds information related ETWS PLMN

**Parameters**

| | |
|---|---|
| *mobileCountry-Code* | • 16 bit representation of MCC value range : 0 -999 |
| *mobileNetwork-Code* | • 16 bit representation of MNC value range : 0 -999 |

#### 9.12.3.20 typedef struct SMSEventInfo_s SMSEventInfo

This structure will hold the information related to received SMS events

**Parameters**

| | |
|---|---|
| *smsEventType* | • Type of the SMS events that are received. This is a bit map of SMSEventType. Only the parameters (which follows) related to the events received would be filled, and the rest of the parameters would be NULL |
| *pMTMessage-Info* | • pointer to the SMSMTMessageInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pTransferRoute-MTMessageInfo* | • pointer to the SMSTransferRouteMTMessageInfo structure<br>. NULL, if this event is not present in the smsEventType parameter |
| *pMessageMode-Info* | • pointer to the SMSMessageModeInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pEtwsMessage-Info* | • pointer to the SMSEtwsMessageInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pEtwsPlmnInfo* | • pointer to the SMSEtwsPlmnInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pSMSCAddress-Info* | • pointer to the SMSCAddressInfo structure<br>NULL, if this event is not present in the smsEventType parameter |
| *pSMSOnIMSInfo* | • pointer to the SMSOnIMSInfo structure<br>NULL, if this event is not present in the smsEventType parameter Note: None |

#### 9.12.3.21 typedef struct SMSMessageMode SMSMessageModeInfo

This structure holds information related to message mode

**Parameters**

| | |
|---|---|
| *messageMode* | • Message mode 0x00 - CDMA 0x01 - GW |

#### 9.12.3.22 typedef struct SMSMTMessage SMSMTMessageInfo

This structure holds information related to MT SMS

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type for the new message 0 - UIM 1 - NV |
| *messageIndex* | • Index of the new message |

#### 9.12.3.23 typedef struct SMSOnIMS SMSOnIMSInfo

This structure holds information related to message mode

**Parameters**

| | |
|---|---|
| *smsOnIMS* | • Indicates whether the message is received from IMS 0x00 - Message is not received from IMS 0x01 - Message is received from IMS 0x02-0xFF - Reserved Note: In multiple modem solutions, this TLV may be used to help the client determine with which modem to communicate. This TLV may not be supported on all implementations. |

#### 9.12.3.24 typedef struct SMSTransferRouteMTMessage SMSTransferRouteMTMessageInfo

This structure holds information related to transfer route MT SMS

**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 |

#### 9.12.3.25 typedef struct svUsedforFix_s svUsedforFix

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 |

#### 9.12.3.26 typedef struct _SwiAvmsEventInd SwiAvmsEventInd

#### 9.12.3.27 typedef struct SwiOTAMsg_s SwiOTAMsg

This structure contains OTA message

**Parameters**

| | |
|---|---|
| *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* | • OTA Message Content |
| *pLteNasRelInfo* | • LTE NAS Release Info<br>• see LteNasReleaseInfo for details |
| *pTime* | • Seconds in local time since Jan. 6th 1980 00:00:00 UTC |

#### 9.12.3.28 typedef void(∗ tFNActivationStatus)(ULONG activationStatus)

Activation status callback function.

**Parameters**

| | |
|---|---|
| *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 |

#### 9.12.3.29 typedef void(∗ tFNAllCallStatus)(voiceSetAllCallStatusCbkInfo ∗pVoiceSetAllCallStatusCbkInfo)

Voice Call Status Callback function. This function pointer will be executed to process received Indication.

**Parameters**

| | |
|---|---|
| *pVoiceSetAllCallStatusCbkInfo* | • Call back will populated memory pointed by this parameter when a call is originated, connected, or ended.<br>See voiceSetAllCallStatusCbkInfo for more information. |

#### 9.12.3.30 typedef void(∗ tFNASwiLTECphyCaInfo)(QmiCbkNasLTECphyCaInfo ∗pQmiCbkNasLTECphyCaInfo)

LTE CPHY CA message callback function.

**Parameters**

| | |
|---|---|
| *pQmiCbkNasLTECphyCaInfo[OUT]* | • Events related to NAS, see QmiCbkNasLTECphyCaInfo for details. |

#### 9.12.3.31 typedef void(∗ tFNASwiOTAMsg)(SwiOTAMsg ∗pSwiOTAMsg)

OTA message callback function.

**Parameters**

| | |
|---|---|
| *pSwiOTAMsg[OUT]* | • Events related to NAS, see SwiOTAMsg for details |

#### 9.12.3.32 typedef void(∗ tFNAsyncRawSend)(SMSAsyncRawSend ∗pSMSAsyncRawSend)

SMS event related callback function.

**Parameters**

| | |
|---|---|
| *pSMSEventInfo[-OUT]* | • Events related to SMS, see SMSEventInfo for details |

#### 9.12.3.33 typedef void(∗ tFNBandPreference)(ULONGLONG band_pref)

Band Preference Callback function

**Parameters**

| | |
|---|---|
| *pBandPref* | - Bit mask representing the current band preference 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 |

**Note**

Timeout: NA To set the band preference the API SLQSSetBandPreference() should be used

#### 9.12.3.34 typedef void(∗ tFNBestAvailPos)(QmiCbkLocBestAvailPosInd ∗pBestAvailPos)

Best available position callback function.

**Parameters**

| | |
|---|---|
| *pBestAvailPos[-OUT]* | • Events related to LOC, see QmiCbkLocBestAvailPosInd for details |

#### 9.12.3.35 typedef void(∗ tFNCATEvent)(ULONG eventID, ULONG eventLen, BYTE ∗pEventData)

CAT event callback function.

**Parameters**

| | |
|---|---|
| *eventID* | • Event ID<br>– 16 - Display Text<br>– 17 - Get In-Key<br>– 18 - Get Input<br>– 19 - Setup Menu<br>– 20 - Select Item<br>– 21 - Send SMS - Alpha Identifier<br>– 22 - Setup Event List<br>– 23 - Setup Idle Mode Text<br>– 24 - Language Notification<br>– 25 - Refresh<br>– 26 - End Proactive Session |
| *eventLen* | • Length of pData (in bytes) |
| *pEventData* | • Data specific to the CAT event ID See currentCatEvent for details |

**Note**

Technology Supported: UMTS

#### 9.12.3.36 typedef void(∗ tFNCbkUimSlotStatusChangeInd)(UIMSlotStatusChangeInfo ∗pQmiCbkUimSlotStatusChangeInd)

Slot Status Change Notification callback.

**Parameters**

| | |
|---|---|
| *pQmiCbkUimSlotStatusChangeInd* | • See UIMSlotStatusChangeInfo for more information. |

#### 9.12.3.37 typedef void(∗ tFNDataCapabilities)(BYTE dataCapsSize, BYTE ∗pDataCaps)

Serving system data capabilities callback function.

**Parameters**

| | |
|---|---|
| *dataCapsSize* | • Number of elements the data capability array contains |
| *pDataCaps* | • Data Capabilities Array.<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 |

#### 9.12.3.38 typedef void( ∗ tFNDataSysStatus)(CurrDataSysStat ∗pCurrDataSysStat)

Data System Status callback.

**Parameters**

| | |
|---|---|
| *pCurrDataSys-Stat* | • See CurrDataSysStat for more information. |

#### 9.12.3.39 typedef void(∗ tFNDelAssistData)(delAssistDataStatus ∗pAssistDataNotification)

Delete Assist Data Notification callback.

**Parameters**

| | |
|---|---|
| *pAssistData-Notification* | • See delAssistDataStatus for more information. |

#### 9.12.3.40 typedef void(∗ tFNDeviceStateChange)(eDevState device_state)

Device State Change callback function prototype

**Parameters**

| | |
|---|---|
| *device_state* | • the current state of the device |

**Note**

Does not require communication with the device

#### 9.12.3.41 typedef void(∗ tFNDHCPv4ClientLeaseStatus)(BYTE instance, WdsDHCPv4ClientLeaseInd ∗pMsg)

DHCPv4 client lease status message callback function.

**Parameters**

| | |
|---|---|
| *pMsg[OUT]* | • Events related to DHCPv4 client lease, see WdsDHCPv4ClientLeaseInd for details |

#### 9.12.3.42 typedef void( ∗ tFNDTMFEvent)(voiceDTMFEventInfo ∗pVoiceDTMFEventInfo)

Preferred DTMF event indication callback.

**Parameters**

| | |
|---|---|
| *pVoiceDTMF-EventInfo* | • See voiceDTMFEventInfo for more information. |

#### 9.12.3.43 typedef void( ∗ tFNDUNCallInfo)(DUNCallInfoInd ∗pDUNCallInfo)

DUN Call Info indication callback.

**Parameters**

| | |
|---|---|
| *pDUNCallInfo* | • See DUNCallInfoInd for more information. |

#### 9.12.3.44 typedef void(∗ tFNEventNmea)(CHAR ∗pNMEAData)

NMEA Event Notification callback.

**Parameters**

| | |
|---|---|
| *pNMEAData* | • NMEA string<br>• Type - NULLterminated string<br>• Maximum string length (including NULL terminator) - 201 |

#### 9.12.3.45 typedef void(∗ tFNEventPosition)(QmiCbkLocPositionReportInd ∗pLocPositionReport)

Position report callback function.

**Parameters**

| | |
|---|---|
| *pLocPosition-Report[OUT]* | • Events related to LOC, see QmiCbkLocPositionReportInd for details |

#### 9.12.3.46 typedef void(∗ tFNFwDldCompletion)(ULONG fwdld_completion_status)

Firmware Download Completion callback function prototype

**Parameters**

| | |
|---|---|
| *error_code* | • error code returned from firmware download operation, the possible return values are listed below:<br>– eQCWWAN_ERR_NONE - indicates firmware download/switching is successful<br>– eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED - indicates no actual download takes place, this is the case of image switching stored on device<br>– eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE - indicates modem enters firmware download mode, firmware flashing is going to be started.<br>– eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE - indicates firmware flashing was complete, SDK is waiting for modem to reboot (can be more than one time), when modem is ready, SDK will send eQCWWAN_ERR_NONE to the host application. |

**Note**

Does not require communication with the device

#### 9.12.3.47 typedef void(∗ tFNGetServer)(QmiCbkLocGetServerInd ∗pGetLocServerInfo)

Get LOC Server Indication callback.

**Parameters**

| | |
|---|---|
| *pGetLocServer-Info* | • See QmiCbkLocGetServerInd for more information. |

#### 9.12.3.48 typedef void(∗ tFNGnssSvInfo)(gnssSvInfoNotification ∗pGnssSvInfoNotification)

GNSS SVN Information Notification callback.

**Parameters**

| | |
|---|---|
| *pGnssSvInfo-Notification* | • See gnssSvInfoNotification for more information. |

#### 9.12.3.49 typedef void( ∗ tFNHDRPersonaity)(HDRPersonalityInd ∗pHDRPers)

HDR Personality indication callback.

**Parameters**

| | |
|---|---|
| *pHDRPers* | • See HDRPersonalityInd for more information. |

**Note**

Technology Supported: CDMA

#### 9.12.3.50 typedef void( ∗ tFNImsaPdpStatus)(imsaPdpStatusInfo ∗pImsaPdpStatusInfo)

IMSA PDP status indication callback.

**Parameters**

| | |
|---|---|
| *pImsaPdp-StatusInfo* | • See imsaPdpStatusInfo for more information. |

#### 9.12.3.51 typedef void( ∗ tFNImsaRatStatus)(imsaRatStatusInfo ∗pImsaRatStatusInfo)

IMSA RAT handover status indication callback.

**Parameters**

| | |
|---|---|
| *pImsaRatStatus-Info* | • See imsaRatStatusInfo for more information. |

#### 9.12.3.52 typedef void( ∗ tFNImsaRegStatus)(imsaRegStatusInfo ∗pImsaRegStatusInfo)

IMSA Registration Status indication callback.

**Parameters**

| | |
|---|---|
| *pImsaReg-StatusInfo* | • See imsaRegStatusInfo for more information. |

#### 9.12.3.53 typedef void( ∗ tFNImsaSvcStatus)(imsaSvcStatusInfo ∗pImsaSvcStatusInfo)

IMSA Service Status indication callback.

**Parameters**

| | |
|---|---|
| *pImsaSvcStatus-Info* | • See imsaSvcStatusInfo for more information. |

#### 9.12.3.54 typedef void( * tFNImsRegMgrConfig)(imsRegMgrConfigInfo *pImsRegMgrConfigInfo)

IMS Reg Mgr Config indication callback.

**Parameters**

| | |
|---|---|
| *pImsRegMgrConfigInfo* | • See imsRegMgrConfigInfo for more information. |

#### 9.12.3.55 typedef void( * tFNImsSIPConfig)(imsSIPConfigInfo *pImsSIPConfigInfo)

IMS SIP Config indication callback.

**Parameters**

| | |
|---|---|
| *pImsSIPConfigInfo* | • See imsSIPConfigInfo for more information. |

#### 9.12.3.56 typedef void( * tFNImsSMSConfig)(imsSMSConfigInfo *pImsSMSConfigInfo)

IMS SMS Config indication callback.

**Parameters**

| | |
|---|---|
| *pImsSMSConfigInfo* | • See imsSMSConfigInfo for more information. |

#### 9.12.3.57 typedef void( * tFNImsUserConfig)(imsUserConfigInfo *pImsUserConfigInfo)

IMS User Config indication callback.

**Parameters**

| | |
|---|---|
| *pImsUserConfigInfo* | • See imsUserConfigInfo for more information. |

#### 9.12.3.58 typedef void( * tFNImsVoIPConfig)(imsVoIPConfigInfo *pImsVoIPConfigInfo)

IMS VoIP Config indication callback.

**Parameters**

| | |
|---|---|
| *pImsVoIPConfigInfo* | • See imsVoIPConfigInfo for more information. |

#### 9.12.3.59 typedef void( ∗ tFNInfoRec)(voiceInfoRec ∗pVoiceInfoRec)

Voice Information Record callback.

**Parameters**

| | |
|---|---|
| *pVoiceInfoRec* | • See voiceInfoRec for more information. |

**Note**

Technology Supported: CDMA
Device Supported: MC7750

#### 9.12.3.60 typedef void(∗ tFNInjectPosition)(QmiCbkLocInjectPositionInd ∗pInjectPositionNotification)

Inject Position Notification callback.

**Parameters**

| | |
|---|---|
| *pInjectPosition-Notification* | • See QmiCbkLocInjectPositionInd for more information. |

#### 9.12.3.61 typedef void(∗ tFNInjectSensorData)(QmiCbkLocInjectSensorDataInd ∗pLocInjectSensorData)

LOC Inject Sensor Data indication callback.

**Parameters**

| | |
|---|---|
| *pLocInject-SensorData* | • See QmiCbkLocInjectSensorDataInd for more information. |

#### 9.12.3.62 typedef void(∗ tFNInjectTimeStatus)(QmiCbkLocInjectTimeInd ∗pLocInjectTime)

LOC Inject time status indication callback.

**Parameters**

| | |
|---|---|
| *pLocInjectTime* | • See QmiCbkLocInjectTimeInd for more information. |

#### 9.12.3.63 typedef void(∗ tFNInjectUTCTime)(QmiCbkLocInjectUTCTimeInd ∗pInjectUTCTimeNotification)

Inject UTC Time Notification callback.

**Parameters**

| | |
|---|---|
| *pInjectUTCTime-Notification* | • See QmiCbkLocInjectUTCTimeInd for more information. |

#### 9.12.3.64 typedef void(∗ tFNLUReject)(ULONG serviceDomain, ULONG rejectCause)

LU reject callback function.

**Parameters**

| | |
|---|---|
| *serviceDomain* | • Service domain<br>– 1 - Circuit Switched<br>– 2 - Packet Switched<br>– 3 - Circuit and Packet Switched |

| | |
|---|---|
| *rejectCause* | • Reject cause<br>• Valid Values<br>– 2 - IMSI unknown in HLR<br>– 3 - Illegal MS<br>– 4 - IMSI unknown in VLR<br>– 5 - IMEI not accepted<br>– 6 - Illegal ME<br>– 7 - EPS services not allowed<br>– 8 - EPS services and non-EPS services not allowed<br>– 9 - UE identity cannot be derived by the network<br>– 10 - Implicitly detached<br>– 11 - PLMN not allowed<br>– 12 - Location Area not allowed<br>– 13 - Roaming not allowed in this location area<br>– 14 - EPS services not allowed in this PLMN<br>– 15 - No Suitable Cells In Location Area<br>– 16 - MSC temporarily not reachable<br>– 17 - Network failure<br>– 18 - CS domain not available<br>– 19 - ESM failure<br>– 20 - MAC failure<br>– 21 - Synch failure<br>– 22 - Congestion<br>– 23 - GSM authentication unacceptable<br>– 24 - Security mode rejected, unspecified<br>– 25 - Not authorized for this CSG<br>– 26 - Non-EPS authentication unacceptable<br>– 32 - Service option not supported<br>– 33 - Requested service option not subscribed<br>– 34 - Service option temporarily out of order<br>– 38 - Call cannot be identified<br>– 48 to 63 - retry upon entry into a new cell<br>– 95 - Semantically incorrect message<br>– 96 - Invalid mandatory information<br>– 97 - Message type non-existent or not implemented<br>– 98 - Message type not compatible with the protocol state<br>– 99 - Information element non-existent or not implemented<br>– 100 - Conditional IE error<br>– 101 - Message not compatible with the protocol state<br>– 111 - Protocol error, unspecified<br>– Note - Any other value received by the mobile station shall be treated as 34, 'Service option temporarily out of order'.<br>* Any other value received by the network shall be treated as 111, 'Protocol error, unspecified'.<br>See 3GPP TS 24.008, Section 4.4.4.7 and Section 10.5.3.6 See qaGobiApiTableCallEndReasons.h for Call End reasons |

**Note**

Technology Supported: UMTS

#### 9.12.3.65 typedef void(∗ tFNMemoryFull)(SMSMemoryInfo ∗pSMSMemoryFullInfo)

SMS Memory related callback function.

**Parameters**

| | |
|---|---|
| *pSMSMemoryFullInfo[OUT]* | • pointer to SMSMemoryInfo.<br>• see SMSMemoryInfo for details. |

#### 9.12.3.66 typedef void(∗ tFNMessageWaiting)(msgWaitingInfo ∗pSMSMessageWaitingInfo)

SMS Memory related callback function.

**Parameters**

| | |
|---|---|
| *pSMSMessageWaitingInfo[OUT]* | • pointer to msgWaitingInfo.<br>• see msgWaitingInfo for details. |

#### 9.12.3.67 typedef void(∗ tFNMitiLvlRpt)(QmiCbkTmdMitiLvlRptInd ∗pSetLocCradleMount)

Set Loc cradle mount Indication callback.

**Parameters**

| | |
|---|---|
| *pSetLocCradleMount* | • See QmiCbkTmdMitiLvlRptInd for more information. |

#### 9.12.3.68 typedef void(∗ tFNMobileIPStatus)(ULONG mipStatus)

Mobile IP status callback function.

**Parameters**

| | |
|---|---|
| *mipStatus* | • Mobile IP Status<br>– 0 - success<br>– All others error codes as defined in RFC 2002<br>See qaGobiApiTableCallEndReasons.h for mobile IP error codes |

#### 9.12.3.69 typedef void(∗ tFNModemTempInfo)(modemTempNotification ∗pModemTempNotification)

Modem Temperature Information callback.

**Parameters**

| | |
|---|---|
| *pModemTemp-Notification* | • See modemTempNotification for more information. |

#### 9.12.3.70 typedef void(∗ tFNNasTimer)(nasTimers ∗timers)

Network timer Indication callback.

**Parameters**

| | |
|---|---|
| *timers* | • See nasTimers for more information. |

#### 9.12.3.71 typedef void(∗ tFNNet)(ULONG q_depth, BYTE isThrottle, BYTE instanceId)

Transmit Queue Length Change callback function prototype

**Parameters**

| | |
|---|---|
| *q_depth* | • transmit queue length |
| *isThrottle* | • 0: unthrottle<br>• 1: throttle |
| *instanceId* | • qmi instance id |

**Note**

Does not require communication with the device

#### 9.12.3.72 typedef void( ∗ tFNNetworkReject)(nasNetworkReject ∗pNetworkReject)

Network Reject Cause callback.

**Parameters**

| | |
|---|---|
| *pNasNetwork-Time* | • See nasNetworkTime for more information. |

#### 9.12.3.73 typedef void( ∗ tFNNetworkTime)(nasNetworkTime ∗pNasNetworkTime)

Network Time indication callback.

**Parameters**

| | |
|---|---|
| *pNasNetwork-Time* | • See nasNetworkTime for more information. |

#### 9.12.3.74 typedef void(∗ tFNNewGPS)(double dLongitude, double dLatitude, BYTE session_status, ULONG pos_src)

Set Current Location Data.

**Parameters**

| | |
|---|---|
| *dLongitude[IN]* | • Current Longitude Value |
| *dLatitude[IN]* | • Current Latitude Value |
| *session_status[I-N]* | • Session Status<br>– 0 - Success<br>– 1 - In progress<br>– 2 - General failure<br>– 3 - Timeout<br>– 4 - User ended the session<br>– 5 - Bad parameter<br>– 6 - Phone is offline<br>– 7 - Engine is locked<br>– 8 - E911 session in progress |
| *pos_src[IN]* | • position source<br>• Bitmasks<br>– 0x01 - GPS<br>– 0x02 - Cell ID<br>– 0x04 - GLONASS<br>– 0x08 - Network<br>– 0x10 - External positino injection<br>– Others - unknown |

#### 9.12.3.75 typedef void(∗ tFNNewNMEA)(LPCSTR pNMEA)

New NMEA sentence callback function.

**Parameters**

| | |
|---|---|
| *pNMEA* | • NULL-terminated string containing the position data in NMEA sentence format |

#### 9.12.3.76 typedef void(∗ tFNNewRMTransferStatistics)(QmiCbkWdsStatisticsIndState ∗pMsg)

PDS session state callback function.

**Parameters**

| | |
|---|---|
| *enabledStatus* | • GPS enabled status<br>– 0 - Disable<br>– 1 - Enable |
| *trackingStatus* | • GPS tracking status<br>– 0 - Unknown<br>– 1 - Inactive<br>– 2 - Active RM Transfer Statistics message callback function. |
| *pMsg[OUT]* | • Events related to NAS, see QmiCbkWdsStatisticsIndState for details |

#### 9.12.3.77 typedef void(∗ tFNNewSMS)(ULONG storageType, ULONG messageIndex)

New SMS message callback function.

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type for the new message<br>0 - UIM 1 - NV |
| *messageIndex* | • Index of the new message |

#### 9.12.3.78 typedef void(∗ tFNOMADMState)(ULONG sessionState, ULONG failureReason)

OMA-DM state callback function

**Parameters**

| | |
|---|---|
| *sessionState* | • Session state<br>– 0x00 - Complete, information was updated<br>– 0x01 - Complete, update information is unavailable<br>– 0x02 - Failed<br>– 0x03 - Retrying<br>– 0x04 - Connecting<br>– 0x05 - Connected<br>– 0x06 - Authenticated<br>– 0x07 - Mobile Directory Number (MDN) downloaded<br>– 0x08 - Mobile Station Identifier (MSID) downloaded<br>– 0x09 - PRL downloaded<br>– 0x0A - Mobile IP (MIP) profile downloaded |
| *failureReason* | • Session failure reason (when state indicates failure)<br>– 0x00 - Unknown<br>– 0x01 - Network is unavailable<br>– 0x02 - Server is unavailable<br>– 0x03 - Authentication failed<br>– 0x04 - Maximum retry exceeded<br>– 0x05 - Session is cancelled |

**Note**

Technology Supported: CDMA

#### 9.12.3.79 typedef void(∗ tFNOpMode)(ULONG mode)

Operation mode callback function.

**Parameters**

| | |
|---|---|
| *mode[OUT]* | • Status of the Set Operation Mode 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 |

#### 9.12.3.80 typedef void( ∗ tFNOTASPStatus)(voiceOTASPStatusInfo ∗pVoiceOTASPStatusInfo)

OTASP or OTAPA event Indication Callback function

**Parameters**

| | |
|---|---|
| *pVoiceOTASP-StatusInfo* | • OTASP Status Information.<br>• See voiceOTASPStatusInfo for more information |

**Note**

Technology Supported: CDMA

#### 9.12.3.81 typedef void(∗ tFNPacketSrvState)(packetSrvStatus ∗pPacketSrvStatus)

Packet Service state callback function.

**Parameters**

| | |
|---|---|
| *pPacketSrv-Status* | • See packetSrvStatus for more details |

#### 9.12.3.82 typedef void(∗ tFNPDSState)(ULONG enabledStatus, ULONG trackingStatus)

PDS session state callback function.

**Parameters**

| | |
|---|---|
| *enabledStatus* | • GPS enabled status<br>– 0 - Disable<br>– 1 - Enable |
| *trackingStatus* | • GPS tracking status<br>– 0 - Unknown<br>– 1 - Inactive<br>– 2 - Active |

#### 9.12.3.83 typedef void(∗ tFNPower)(ULONG operatingMode)

Power operating mode callback function.

**Parameters**

| | |
|---|---|
| *operatingMode* | • Service Operating mode<br>See Tables for Operating Modes |

**Note**

Technology Supported: UMTS/CDMA
Device Supported: MC83x5, MC7700/50

#### 9.12.3.84 typedef void( ∗ tFNPrivacyChange)(voicePrivacyInfo ∗pVoicePrivacyInfo)

Preferred voice privacy indication callback.

**Parameters**

| | |
|---|---|
| *pVoicePrivacy-Info* | • See voicePrivacyInfo for more information. |

**Note**

Technology Supported: CDMA

#### 9.12.3.85 typedef void(∗ tFNPSMCfgParamChange)(dmsPsmCfgParamChange ∗pPSMCfgParamInd)

Get PSM status Indication callback.

**Parameters**

| | |
|---|---|
| *pPSMCgfParam-Ind* | • See dmsPsmCfgParamChange for more information. |

#### 9.12.3.86 typedef void(∗ tFNQosNWStatus)(BYTE status)

QOS Network status callback function.

**Parameters**

| | |
|---|---|
| *status* | Network QoS support status<br>• 0x00 – Current network does not support QoS<br>• 0x01 – Current network supports QoS |

#### 9.12.3.87 typedef void(∗ tFNQosPriEvent)(WORD event)

QOS primary flow callback function.

**Parameters**

| | |
|---|---|
| *event* | Event which causes this indication:<br>• 0x0001 – Primary flow QoS modify operation success<br>• 0x0002 – Primary flow QoS modify operation failure |

#### 9.12.3.88 typedef void(∗ tFNQosStatus)(BYTE instance, ULONG id, BYTE status, BYTE event, BYTE reason)

QOS Status callback function.

**Parameters**

| | |
|---|---|
| *instance* | • QMI instance |
| *id* | • Index identifying the QoS flow whose status is being reported |
| *status* | Current QoS flow status:<br>• 0x01 – QMI_QOS_STATUS_ACTIVATED<br>• 0x02 – QMI_QOS_STATUS_SUSPENDED<br>• 0x03 – QMI_QOS_STATUS_GONE |
| *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 |

| | |
|---|---|
| *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 |

#### 9.12.3.89 typedef void( ∗ tFNRankIndicator)(RankIndicatorInd ∗pRankIndicatorInd)

Rank Indicator callback function.

**Parameters**

| | |
|---|---|
| *pRankIndicator-Ind[OUT]* | • Events related to NAS, see RankIndicatorInd for details. |

#### 9.12.3.90 typedef void(∗ tFNResetInfo)(ResetInfoNotification ∗pResetInfoNotification)

Get Reset Info Indication callback.

**Parameters**

| | |
|---|---|
| *pResetInfo-Notification* | • See ResetInfoNotification for more information. |

#### 9.12.3.91 typedef void(∗ tFNRFInfo)(ULONG radioInterface, ULONG activeBandClass, ULONG activeChannel)

RF information callback function.

**Parameters**

| | |
|---|---|
| *radioInterface* | • Radio interface technology of the signal being measured<br>See Tables for Radio Interface |
| *activeBandClass* | • Active band class<br>See Tables for Active Band Class |
| *activeChannel* | • Active channel<br>– 0 - Channel is not relevant to the reported technology |

#### 9.12.3.92 typedef void(∗ tFNRoamingIndicator)(ULONG roaming)

Roaming indicator callback function.

**Parameters**

| | |
|---|---|
| *roaming* | • Roaming Indication<br>– 0 - Roaming<br>– 1 - Home<br>– 2 - Roaming partner<br>– >2 - Operator defined values |

#### 9.12.3.93 typedef void(∗ tFNSDKTerminated)(BYTE ∗psReason)

SDK terminated callback function prototype

**Parameters**

| | |
|---|---|
| *psReason* | • sdk termination reason string |

**Note**

Timeout: None
Does not require communication with the device

#### 9.12.3.94 typedef void(∗ tFNSensorStreaming)(QmiCbkLocSensorStreamingInd ∗pLocSensorStream)

LOC Sensor Streaming indication callback.

**Parameters**

| | |
|---|---|
| *pLocSensor-Stream* | • See QmiCbkLocSensorStreamingInd for more information. |

#### 9.12.3.95 typedef void(∗ tFNServingSystem)(struct ServingSystemInfo ∗pServingSystem,struct RoamingInfo ∗pRoamingInfo)

Serving System callback function

**Parameters**

| | |
|---|---|
| *pServingSystem* | • ServingSystemInfo structure |

#### 9.12.3.96 typedef void(∗ tFNSetCradleMount)(QmiCbkLocCradleMountInd ∗pSetLocCradleMount)

Set Cradle Mount indication callback.

**Parameters**

| | |
|---|---|
| *pSetLocCradle-Mount* | • See QmiCbkLocCradleMountInd for more information. |

#### 9.12.3.97 typedef void(∗ tFNSetEngineState)(QmiCbkLocEngineStateInd ∗pSetLocEngineState)

Set Engine State Indication callback.

**Parameters**

| | |
|---|---|
| *pSetLocEngine-State* | • See QmiCbkLocEngineStateInd for more information. |

#### 9.12.3.98 typedef void(∗ tFNSetEventTimeSync)(QmiCbkLocEventTimeSyncInd ∗pSetLocEventTimeSync)

LOC event time sync indication callback.

**Parameters**

| | |
|---|---|
| *pSetLocEvent-TimeSync* | • See QmiCbkLocEventTimeSyncInd for more information. |

#### 9.12.3.99 typedef void(∗ tFNSetExtPowerConfig)(QmiCbkLocSetExtPowerConfigInd ∗pSetExtConfigIndStatus)

Set Ext Power Config Indication callback.

**Parameters**

| | |
|---|---|
| *pSetExtConfig-IndStatus* | • See QmiCbkLocSetExtPowerConfigInd for more information. |

#### 9.12.3.100 typedef void(∗ tFNSetServer)(QmiCbkLocSetServerInd ∗pSetLocServerInfo)

Set LOC Server Indication callback.

**Parameters**

| | |
|---|---|
| *pSetLocServer-Info* | • See QmiCbkLocSetServerInd for more information. |

#### 9.12.3.101 typedef void(∗ tFNSigInfo)(nasSigInfo ∗pNasSigInfo)

Signal Strength Information indication callback.

**Parameters**

| | |
|---|---|
| *pNasSigInfo* | • See nasSigInfo for more information. |

#### 9.12.3.102 typedef void(∗ tFNSignalStrength)(INT8 signalStrength, ULONG radioInterface)

Signal strength callback function.

**Parameters**

| | |
|---|---|
| *signalStrength* | • Received signal strength (in dBm) |
| *radioInterface* | • Radio interface technology of the signal being measured<br>See Tables for Radio Interface |

#### 9.12.3.103 typedef void(∗ tFNSLQSOMADMAlert)(ULONG eventType, BYTE ∗pEventFields)

SWIOMA-DM network-initiated alert callback function

**Parameters**

| | |
|---|---|
| *eventType* | • 0x00 - SWIOMA-DM FOTA<br>• 0x01 - SWIOMA-DM Config<br>• 0x02 - SWIOMA-DM Notification |
| *pEventFields* | • Pointer to structure containing info for that session type<br>• See sessionInfo for more details |

#### 9.12.3.104 typedef void(* tFNSLQSQOSEvent)(BYTE instance, QosFlowInfo *pFlowInfo)

QOS Event callback function.

**Parameters**

| | |
|---|---|
| *instance* | • QMI instance |
| *pFlowInfo* | • See QosFlowInfo for more information |

#### 9.12.3.105 typedef void(* tFNSLQSSessionState)(slqsSessionStateInfo *pSessionStateInfo)

Session state callback function.

**Parameters**

| | |
|---|---|
| *pSessionStateInfo* | • See slqsSessionStateInfo for more details |

#### 9.12.3.106 typedef void(* tFNSLQSSignalStrengths)(struct SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo)

Received Signal Strength Information callback function.

**Parameters**

| | |
|---|---|
| *sSLQSSignalStrengthsInfo* | • See SLQSSignalStrengthsInformation for more information. |

#### 9.12.3.107 typedef void(* tFNSLQSWDSEvent)(slqsWdsEventInfo *pWdsEventInfo)

WDS Event callback function.

**Parameters**

| | |
|---|---|
| *pWdsEventInfo* | • See slqsWdsEventInfo for more details |

#### 9.12.3.108 typedef void(* tFNSMSEvents)(SMSEventInfo *pSMSEventInfo)

SMS event related callback function.

**Parameters**

| | |
|---|---|
| *pSMSEventInfo[-OUT]* | • Events related to SMS, see SMSEventInfo for details |

#### 9.12.3.109 typedef void( ∗ tFNSUPSInfo)(voiceSUPSInfo ∗pVoiceSUPSInfo)

Preferred SUPS indication callback.

**Parameters**

| | |
|---|---|
| *pVoiceSUPSInfo* | • See voiceSUPSInfo for more information. |

**Note**

Technology Supported: GSM

#### 9.12.3.110 typedef void(∗ tFNSUPSNotification)(voiceSUPSNotification ∗pVoiceSUPSNotification)

Supplementary service notification callback.

**Parameters**

| | |
|---|---|
| *pVoiceSUPS-Notification* | • See voiceSUPSNotification for more information. |

#### 9.12.3.111 typedef void( ∗ tFNSwiAvmEventReport)(SwiAvmsEventInd ∗pEvent)

AVMS Event Report indication callback.

**Parameters**

| | |
|---|---|
| *pEvent* | • See SwiAvmsEventInd for more information. |

#### 9.12.3.112 typedef void( ∗ tFNSysInfo)(nasSysInfo ∗pNasSysInfo)

System Information indication callback.

**Parameters**

| | |
|---|---|
| *pNasSysInfo* | • See nasSysInfo for more information. |

#### 9.12.3.113 typedef void(∗ tFNSysSelectionPref)(sysSelectPrefInfo ∗pSysSelectPrefInfo)

System Selection Preference Callback function

**Parameters**

| | |
|---|---|
| *pSysSelectPref-Info* | • Current System Selection preferences for the device.<br>• See sysSelectPrefInfo for more information |

#### 9.12.3.114 typedef void(* tFNtransLayerInfo)(transLayerNotification *pTransLayerNotification)

Transport Layer Information callback.

**Parameters**

| | |
|---|---|
| *transLayer-Notification* | • See transLayerNotification for more information. |

#### 9.12.3.115 typedef void(* tFNtransNWRegInfo)(transNWRegInfoNotification *pTransNWRegInfoNotification)

Transport Network Registration Information callback.

**Parameters**

| | |
|---|---|
| *pTransNWReg-InfoNotification* | • See transNWRegInfoNotification for more information. |

#### 9.12.3.116 typedef void(* tFNUIMRefresh)(UIMRefreshEvent *pUIMRefreshEvent)

UIM Refresh Callback function

**Parameters**

| | |
|---|---|
| *pUIMRefresh-Event* | • Pointer to Refresh Event structure.<br>• See UIMRefreshEvent for more information |

#### 9.12.3.117 typedef void( * tFNUIMStatusChangeInfo)(UIMStatusChangeInfo *pUIMStatusChangeInfo)

UIM Status Change Callback function

**Parameters**

| | |
|---|---|
| *pUIMStatus-ChangeInfo* | • Pointer to UIM status change structure.<br>• See UIMStatusChangeInfo for more information |

#### 9.12.3.118 typedef void(* tFNUSSDNotification)(ULONG type, BYTE *pNetworkInfo)

SetUSSDNotificationCallback function prototype

**Parameters**

| | |
|---|---|
| *type* | - Notification type<br>• 0x01 - No action required<br>• 0x02 - Action required |

| | |
|---|---|
| *pNetworkInfo* | • USS information from the network (0 indicates that no info was received)<br>– See USSInfo for more details |

**Note**

Technology Supported: UMTS

#### 9.12.3.119 typedef void(∗ tFNUSSDNoWaitIndication)(USSDNoWaitIndicationInfo ∗pNetworkInfo)

USSD No Wait indication callback.

**Parameters**

| | |
|---|---|
| *pNetworkInfo* | • See USSDNoWaitIndicationInfo for more information. |

#### 9.12.3.120 typedef void(∗ tFNUSSDRelease)(void)

USSD releasecallback function prototype

**Note**

Technology Supported: UMTS

#### 9.12.3.121 typedef struct _transLayerInfoNotification transLayerNotification

Contains the parameters passed for SLQSSetTransLayerInfoCallback by the device.

**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 |
| *pTransLayerInfo* | • Optional parameter<br>• See transLayerInfo for more information |

**Note**

None

#### 9.12.3.122 typedef struct _transNWRegInfoNotification transNWRegInfoNotification

Contains the parameters passed for SLQSSetTransNWRegInfoCallback 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 |

**Note**

None

### 9.12.4 Enumeration Type Documentation

#### 9.12.4.1 enum device_state_enum

Device State enumeration

- See device_state_enum for more details

**Enumerator**

***DEVICE_STATE_DISCONNECTED***

***DEVICE_STATE_READY***

***DEVICE_STATE_BOOT***

***DEVICE_STATE_SERIAL_ONLY***

#### 9.12.4.2 enum eQaQMIService

The QMI service information which is exposed to the application, only the services which are relevant to multiple PDP are listed in this enumeration as these are the only required services to be exposed.

**Enumerator**

***eQA_QMI_SVC_WDS***

***eQA_QMI_SVC_NAS***

***eQA_QMI_SVC_NA***

#### 9.12.4.3 enum SMSEventType

This enumeration defines the different type of SMS events that are received

- See SMSEventType for more details

**Enumerator**

***SMS_EVENT_MT_MESSAGE***

***SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE***

***SMS_EVENT_MESSAGE_MODE***

***SMS_EVENT_ETWS***

***SMS_EVENT_ETWS_PLMN***

***SMS_EVENT_SMSC_ADDRESS***

***SMS_EVENT_SMS_ON_IMS***

#### 9.12.4.4 enum TYPE_OF_SERVICE_DOMAIN

**Enumerator**

***SYS_SRV_DOMAIN_NO_SRV***

***SYS_SRV_DOMAIN_CS_ONLY***

***SYS_SRV_DOMAIN_PS_ONLY***

***SYS_SRV_DOMAIN_CS_PS***

***SYS_SRV_DOMAIN_CAMPED***

### 9.12.5 Function Documentation

#### 9.12.5.1 ULONG iSetCATEventCallback ( tFNCATEvent *pCallback* )

#### 9.12.5.2 ULONG iSetSignalStrengthCallback ( tFNSignalStrength *pCallback* )

#### 9.12.5.3 ULONG iSLQSSetDUNCallInfoCallback ( tFNDUNCallInfo *pCallback* )

#### 9.12.5.4 ULONG iSLQSSetSignalStrengthsCallback ( tFNSLQSSignalStrengths *pCallback* )

#### 9.12.5.5 ULONG iSLQSSetWdsFirstInstEventCallback ( tFNSLQSWDSEvent *pCallback* )

#### 9.12.5.6 ULONG iSLQSSetWdsSecondInstEventCallback ( tFNSLQSWDSEvent *pCallback* )

#### 9.12.5.7 ULONG iSLQSSetWdsThirdInstEventCallback ( tFNSLQSWDSEvent *pCallback* )

#### 9.12.5.8 ULONG iSLQSSetWdsXferStatsFirstInstCallback ( tFNSLQSWDSEvent *pCallback* )

#### 9.12.5.9 ULONG iSLQSSetWdsXferStatsSecondInstCallback ( tFNSLQSWDSEvent *pCallback* )

#### 9.12.5.10 ULONG SetActivationStatusCallback ( tFNActivationStatus *pCallback* )

Enables/disables the Activation Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.5.11 ULONG SetCATEventCallback ( tFNCATEvent *pCallback*, ULONG *eventMask*, ULONG * *pErrorMask* )

Enables/disables the CAT event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |
| *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 |
| *pErrorMask[OUT]* | • 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 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Technology Supported: UMTS
Timeout: 2 seconds

#### 9.12.5.12 ULONG SetCfgParamChangeCallback ( tFNPSMCfgParamChange *pCallback* )

Enables/disables PSM Configuration Change callback function.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable)<br>• See tFNPSMCfgParamChange for more information. |

#### 9.12.5.13 ULONG SetDataCapabilitiesCallback ( tFNDataCapabilities *pCallback* )

Enables/disables the data capabilities callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Does not require communication with the device

#### 9.12.5.14 ULONG SetDeviceStateChangeCbk ( tFNDeviceStateChange *pCallback* )

Used by the client application to register a Callback function for Device State Change (DSC) event notifications. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to be notified of DSC events<br>• NULL to disable DSC event notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.5.15 ULONG SetFwDldCompletionCbk ( tFNFwDldCompletion *pCallback* )

Used by the client application to register a Callback function for a Firmware Download Completion (FDC) event notification. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable FDC event notification<br>• NULL to disable FDC event notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: N/A

#### 9.12.5.16 ULONG SetGPSCallback ( tFNNewGPS *pCallback* )

Enables/disables the NMEA sentence callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.17 ULONG SetLocBestAvailPosCallback ( tFNBestAvailPos *pCallback* )

Enables/disables Best Available Location callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.18 ULONG SetLocCradleMountCallback ( tFNSetCradleMount *pCallback* )

Enables/disables the Cradle Mount callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.19 ULONG SetLocDeleteAssistDataCallback ( tFNDelAssistData *pCallback* )

Enables/disables Delete Assist Data callback function. This API is used to receive the SUCCESS/FAILURE status of API SLQSLOCDelAssData().

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

#### 9.12.5.20 ULONG SetLocEngineStateCallback ( tFNSetEngineState *pCallback* )

Sends the GPS State Information event to the control point.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.21 ULONG SetLocEventMaskNMEACallback ( tFNEventNmea *pCallback* )

Enables/disables NMEA Event callback function

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable)<br>• See tFNEventNmea for more information. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.22 ULONG SetLocEventPositionCallback ( tFNEventPosition *pCallback* )

Enables/disables the Event Position Report callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.23 ULONG SetLocEventTimeSyncCallback ( tFNSetEventTimeSync *pCallback* )

Enables/disables the Event Time Sync callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.24 ULONG SetLocGetServerCallback ( LocGetServerReq * *req,* tFNGetServer *pCallback* )

Enables/disables Get location server callback function

**Parameters**

| | |
|---|---|
| *req[IN]* | • See LocGetServerReq for more information. |
| *pCallback[IN]* | • Callback function pointer (0 - Disable)<br>• See tFNGetServer for more information. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.25 ULONG SetLocGnssSvInfoCallback ( tFNGnssSvInfo *pCallback* )

Enables/disables the GNSS SV Info callback function. This API is used to send the satellite report to the application. The satellite reports are sent only to the application that invoked API SLQSLOCStart() that generated the satellite report.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.26 ULONG SetLocInjectSensorDataCallback ( tFNInjectSensorData *pCallback* )

Enables/disables the Inject Sensor Data callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.27 ULONG SetLocInjectTimeCallback ( tFNInjectTimeStatus *pCallback* )

Enables/disables the Inject Time Sync Data callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.28 ULONG SetLocOpModeCallback ( tFNOpMode *pCallback* )

Enables/disables Set Operating Mode callback function. This API is used to receive the SUCCESS/FAILURE status of API SLQSLOCSetOpMode().

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

#### 9.12.5.29 ULONG SetLocSensorStreamingCallback ( tFNSensorStreaming *pCallback* )

Enables/disables the Event Sensor Streaming Ready Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.30 ULONG SetLocSetExtPowerConfigCallback ( tFNSetExtPowerConfig *pCallback* )

Enables/disables the Set External Power Config Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.31 ULONG SetLocSetServerCallback ( LocSetServerReq * *req*, tFNSetServer *pCallback* )

Set location server callback function

**Parameters**

| | |
|---|---|
| *req[IN]* | • See LocSetServerReq for more information. |
| *pCallback[IN]* | • Callback function pointer (0 - Disable)<br>• See tFNSetServer for more information. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.32 ULONG SetLURejectCallback ( tFNLUReject *pCallback* )

Enables/disables the LU reject callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**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

#### 9.12.5.33 ULONG SetMobileIPStatusCallback ( tFNMobileIPStatus *pCallback* )

Enables/disables the Mobile IP Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**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
Timeout: 2 seconds

#### 9.12.5.34 ULONG SetNasLTECphyCaIndCallback ( tFNASwiLTECphyCaInfo *pCallback* )

Enables/disables the LTE NAS CA Info callback function.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.35 ULONG SetNetChangeCbk ( BYTE *instance,* tFNNet *pCallback,* ULONG *loMark,* ULONG *hiMark,* ULONG *period* )

Used by the client application to register a Callback function for USB Transmit Queue Length Change event notifications. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *instance[IN]* | • PDP instance |
| *pCallback[IN]* | • a valid function pointer to be notified of the event<br>• NULL to disable the event notification |
| *loMark[IN]* | • Transmit queue length smaller will trigger unthrottle event notification |
| *hiMark[IN]* | • Transmit queue length larger will trigger throttle event notification |
| *period[IN]* | • monitoring period in seconds, minimum 1 second |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.5.36 ULONG SetNewSMSCallback ( tFNNewSMS *pCallback* )

Enables/disables the new SMS callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Timeout: 2 seconds

#### 9.12.5.37 ULONG SetNMEACallback ( tFNNewNMEA *pCallback* )

Enables/disables the NMEA sentence callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use SetLocEventPositionCallback()

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.38 ULONG SetOMADMStateCallback ( tFNOMADMState *pCallback* )

Enables/disables the OMADM state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use SetSLQSOMADMAlertCallback()

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable OMADMState notification<br>• NULL to disable OMADMState notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Technology Supported: CDMA
Timeout: 2 seconds

#### 9.12.5.39 ULONG SetPDSStateCallback ( tFNPDSState *pCallback* )

Enables/disables the PDS service state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.40 ULONG SetPowerCallback ( tFNPower *pCallback* )

Enables/disables the Operating Mode callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**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

#### 9.12.5.41 ULONG SetRankIndicatorCallback ( tFNRankIndicator *pCallback* )

#### 9.12.5.42 ULONG SetRFInfoCallback ( tFNRFInfo *pCallback* )

Enables/disables the radio frequency information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**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

#### 9.12.5.43 ULONG SetRMTransferStatisticsCallback ( tFNNewRMTransferStatistics *pCallback* )

Enables/disables the RM Transfer Statistics callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.44 ULONG SetRoamingIndicatorCallback ( tFNRoamingIndicator *pCallback* )

Enables/disables the Roaming Indicator callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use SLQSNasSysInfoCallBack() instead

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**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

#### 9.12.5.45 ULONG SetSignalStrengthCallback ( tFNSignalStrength *pCallback,* BYTE *thresholdsSize,* INT8 * *pThresholds* )

Enables/disables the Signal Strength callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This API is deprecated on MC73xx/-EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSNasIndicationRegisterExt() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |
| *thresholdsSize* | • Number of elements threshold array contains; a maximum of five thresholds is supported;<br>• This parameter is not used when disabling the callback. |
| *pThresholds[IN]* | • Signal threshold array for each entry (in dBm).<br>• This parameter is not used when disabling the callback. |

**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
The signal strength callback function is called when a threshold in the threshold array is crossed.

#### 9.12.5.46 ULONG SetSLQSOMADMAlertCallback ( tFNSLQSOMADMAlert *pCallback* )

Enables/disables the SWIOMADM network-initiated alert callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable SLQSOMADMAlert notification<br>• NULL to disable SLQSOMADMAlert notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.47 ULONG SetSLQSOMADMAlertCallbackExt ( tFNSLQSOMADMAlert *pCallback* )

Enables/disables the SWIOMADM network-initiated alert callback function for SL9090 module. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable SLQSOMADMAlert notification<br>• NULL to disable SLQSOMADMAlert notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Technology Supported: UMTS/CDMA
Device Supported: SL9090
Timeout: 2 seconds

#### 9.12.5.48 ULONG SetUimSlotStatusChangeCallback ( tFNCbkUimSlotStatusChangeInd *pCallback* )

Enables/disables Slot Status Change callback function.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.49 ULONG SetUSSDNotificationCallback ( tFNUSSDNotification *pCallback* )

Enables/disables the USSDNotification callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable ServingSystem notification<br>• NULL to disable ServingSystem notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Technology Supported: UMTS
Timeout: Does not require communication with device

#### 9.12.5.50 ULONG SetUSSDNoWaitIndicationCallback ( tFNUSSDNoWaitIndication *pCallback* )

SetUSSDNoWaitIndicationCallback

**Parameters**

| | |
|---|---|
| *pNetworkInfo* | • Data from the network.<br>• See USSDNoWaitIndicationInfo for more details. |

**Note**

Technology Supported: UMTS
Device Supported: MC83x5

#### 9.12.5.51 ULONG SetUSSDReleaseCallback ( tFNUSSDRelease *pCallback* )

Enables/disables the USSD release callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable ServingSystem notification<br>• NULL to disable ServingSystem notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Technology Supported: UMTS
Timeout: Does not require communication with the device

#### 9.12.5.52 ULONG SLQSNasNetworkRejectCallback ( tFNNetworkReject *pCallback* )

#### 9.12.5.53 ULONG SLQSNasNetworkTimeCallBack ( tFNNetworkTime *pCallback* )

Enables/disables the Network Time callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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
This callback is sent when the 3GPP or 3GPP2 network sends time information to the User Equipment.

#### 9.12.5.54 ULONG SLQSNasSigInfo2CallBack ( tFNSigInfo *pCallback,* setSignalStrengthInfo * *pSigInfo2* )

Enables/disables the Signal Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |
| *pSigInfo2[IN]* | • Structure containing the threshold values beyond which signal information is to be reported<br>• See setSignalStrengthInfo for more details |

**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
This callback is sent when the signal strength change occurs

#### 9.12.5.55 ULONG SLQSNasSigInfoCallBack ( tFNSigInfo *pCallback*, sigInfo * *pSigInfo* )

Enables/disables the Signal Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback is deprecated on MC73xx/-EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use callback SLQSNasSigInfo2CallBack() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |
| *pSigInfo[IN]* | • Structure containing the threshold values beyond which signal information is to be reported<br>• See sigInfo for more details |

**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
This callback is sent when the signal strength change occurs

#### 9.12.5.56 ULONG SLQSNasSwiOTAMessageCallback ( NasSwiIndReg * *req*, tFNASwiOTAMsg *pCallback* )

Enables/disables the SLQSNasSwiOTAMessageCallback callback function. To disable the callback, provide both req and pCallback as NULL pointer to the API

**Parameters**

| | |
|---|---|
| *req[IN]* | • the request to which kind of message type should be enabled, see NasSwiIndReg for details |
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.57 ULONG SLQSNasSysInfoCallBack ( tFNSysInfo *pCallback* )

Enables/disables the Sys Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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
This callback provides current serving system information, including registration information and system property. The serving system information of the radio interfaces specified in mode_pref are included in the response message. When any value in the sys_info message changes, an indication message is sent. Indications contain all the values for all active RATs.

#### 9.12.5.58 ULONG SLQSNasTimerCallback ( tFNNasTimer *pCallback* )

Enables/disables Network Timer callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.59 ULONG SLQSSetBandPreferenceCbk ( tFNBandPreference *pCallback* )

Enables/disables the Band Preference callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable Band Preference Indication notification<br>• NULL to disable Band Preference notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Timeout: NA To set the band preference the API SLQSSetBandPreference() should be used

#### 9.12.5.60 ULONG SLQSSetDataSystemStatusCallback ( tFNDataSysStatus *pCallback* )

Enables/disables the Data System Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.12.5.61 ULONG SLQSSetDHCPv4ClientLeaseStatusCallback ( BYTE *instance,* tFNDHCPv4ClientLeaseStatus *pCallback* )

Enables/disables the DHCP Client V4 Lease Status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *instance[IN]* | • QMI instance |
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.62 ULONG SLQSSetDUNCallInfoCallback ( BYTE *StatsPeriod,* tFNDUNCallInfo *pCallback* )

Enables/disables the DUN Call Info callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *StatsPeriod[IN]* | • Period between reports(seconds)<br>• 0 - Do not report<br>• Only applicable to pTXOKBytesCount and pRXOKBytesCount parameters |
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.12.5.63 ULONG SLQSSetIMSAPdpStatusCallback ( tFNImsaPdpStatus *pCallback* )

SLQSSetIMSAPdpStatusCallback

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.12.5.64 ULONG SLQSSetIMSARatStatusCallback ( tFNImsaRatStatus *pCallback* )

SLQSSetIMSARatStatusCallback

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.12.5.65 ULONG SLQSSetIMSARegStatusCallback ( tFNImsaRegStatus *pCallback* )

SLQSSetIMSARegStatusCallback

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.12.5.66 ULONG SLQSSetIMSASvcStatusCallback ( tFNImsaSvcStatus *pCallback* )

SLQSSetIMSASvcStatusCallback

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.12.5.67 ULONG SLQSSetIMSSMSConfigCallback ( tFNImsSMSConfig *pCallback* )

Enables/disables the SMS Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.68 ULONG SLQSSetIMSUserConfigCallback ( tFNImsUserConfig *pCallback* )

Enables/disables the User Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.69 ULONG SLQSSetIMSVoIPConfigCallback ( tFNImsVoIPConfig *pCallback* )

Enables/disables the VoIP Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.70 ULONG SLQSSetLocInjectPositionCallback ( tFNInjectPosition *pCallback* )

Enables/disables Inject Position callback function. This API is used to receive the SUCCESS/FAILURE status of API SLQSLOCInjectPosition().

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

#### 9.12.5.71 ULONG SLQSSetLocInjectUTCTimeCallback ( tFNInjectUTCTime *pCallback* )

Enables/disables Inject UTC Time callback function. This API is used to receive the SUCCESS/FAILURE status of API SLQSLOCInjectUTCTime().

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

#### 9.12.5.72 ULONG SLQSSetModemTempCallback ( tFNModemTempInfo *pCallback* )

Enables/disables the Modem Temperature information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.12.5.73 ULONG SLQSSetPacketSrvStatusCallback ( tFNPacketSrvState *pCallback* )

Enables/disables the session state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: none; does not require communication with the device

#### 9.12.5.74 ULONG SLQSSetQosEventCallback ( BYTE *instance,* tFNSLQSQOSEvent *pCallback* )

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS flow state

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| in | *pCallback* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

maximum number of tx/rx filters supported is 25 (pTxQFilter/pRxQFilter)

#### 9.12.5.75 ULONG SLQSSetQosNWStatusCallback ( tFNQosNWStatus *pCallback* )

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS Network supports status

**Parameters**

| | | |
|---|---|---|
| in | *pCallback[IN]* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.5.76 ULONG SLQSSetQosPriEventCallback ( tFNQosPriEvent *pCallback* )

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS Primary flow event

**Parameters**

| | | |
|---|---|---|
| in | *pCallback* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS

#### 9.12.5.77 ULONG SLQSSetQosStatusCallback ( BYTE *instance,* tFNQosStatus *pCallback* )

Enables/disables the QoS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the QoS status

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |

| | | |
|---|---|---|
| in | *pCallback[IN]* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.5.78 ULONG SLQSSetRegMgrConfigCallback ( tFNImsRegMgrConfig *pCallback* )

Enables/disables the Reg Mgr Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.79 ULONG SLQSSetSDKTerminatedCallback ( tFNSDKTerminated *pCallback* )

Used by the client application to register a Callback function for SDK terminated event notifications. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to be notified of SWI events<br>• NULL to disable SWI event notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: N/A
Device Supported: N/A
Timeout: N/A
The following signals will trigger this callback:

```
2 INT      4 ILL      5 TRAP     6 ABRT     7 BUS
8 FPE     11 SEGV    13 PIPE    15 TERM    31 SYS
```

#### 9.12.5.80 ULONG SLQSSetServingSystemCallback ( tFNServingSystem *pCallback* )

Enables/disables the Serving System callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

This API is deprecated on all MC/EM74xx firmware versions. Please use SLQSNasSysInfoCallBack()

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable ServingSystem notification<br>• NULL to disable ServingSystem notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.81 ULONG SLQSSetSessionStateCallback ( tFNSLQSSessionState *pCallback* )

Enables/disables the session state callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the multiple PDP interface

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: none; does not require communication with the device

#### 9.12.5.82 ULONG SLQSSetSignalStrengthsCallback ( tFNSLQSSignalStrengths *pCallback,* struct SLQSSignalStrengthsIndReq * *pSLQSSignalStrengthsIndReq* )

Enables/disables the Received Signal Strength Information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This API is same as API SetSignalStrengthsCallback() except providing more information of signal such as ECIO, SNR etc. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all E-M74xx firmware versions. Please use API SLQSNasIndicationRegisterExt() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |
| *pSLQSSignal-StrengthsIndReq* | • See SLQSSignalStrengthsIndReq for more information<br>• This parameter is not used when disabling the callback. |

**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
The signal strength callback function is called when a threshold in the threshold array is crossed.

#### 9.12.5.83 ULONG SLQSSetSIPConfigCallback ( tFNImsSIPConfig *pCallback* )

Enables/disables the SIP Config callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.84 ULONG SLQSSetSMSEventCallback ( tFNSMSEvents *pCallback* )

Enables/disables the events related to SMS callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### 9.12.5.85 ULONG SLQSSetSwiGetResetInfoCallback ( tFNResetInfo *pCallback* )

Reset Info callback.

**Parameters**

| | |
|---|---|
| *pCallback* | • See tFNResetInfo for more information. |

#### 9.12.5.86 ULONG SLQSSetSwiHDRPersCallback ( tFNHDRPersonaity *pCallback* )

Enables/disables the HDR Personality callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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
Timeout: 5 seconds

#### 9.12.5.87 ULONG SLQSSetSysSelectionPrefCallBack ( tFNSysSelectionPref *pCallback* )

Enables/disables the System Selection Preference callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable System Selection Preference Indication notification<br>• NULL to disable Band Preference notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Timeout: 2 seconds
To set the system selection preferences the API SLQSSetSysSelectionPref() should be used

#### 9.12.5.88 ULONG SLQSSetTransLayerInfoCallback ( tFNtransLayerInfo *pCallback* )

Enables/disables the Transport Layer information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.89 ULONG SLQSSetTransNWRegInfoCallback ( tFNtransNWRegInfo *pCallback* )

Enables/disables the Transport Network Registration information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.90 ULONG SLQSSetWdsEventCallback ( tFNSLQSWDSEvent *pCallback,* BYTE *interval,* BYTE *instanceid,* BYTE *ipfamily* )

Enables/disables the WDS event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the multiple PDP interface. Transfer statistic are reported only when changed.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |
| *interval* | • Interval in seconds.<br>• ignored when disabling, should be non-zero when enabling<br>• period only affect transfer statistic attributes |
| *instanceid* | • PDP instance id 0 - First PDP instance 1 - Second PDP instance 2 - Third PDP instance |
| *ipfamily* | • 4 for an IPv4 data session<br>• 6 for an IPv6 data session<br>• 7 for an IPv4v6 data session |

**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 Currently 3 PDP instances are supported in device. user of this callback can subscribe by passing instanceid of particular instance. All PDP instance can be subscribed by passing instanceid sequentially.

#### 9.12.5.91 ULONG SLQSSetWdsTransferStatisticCallback ( tFNSLQSWDSEvent *pXferStatsCb,* BYTE *interval,* BYTE *instanceid,* BYTE *ipfamily* )

Enables/disables the WDS transfer statistic callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. This callback provide extra information regarding the multiple PDP interface. Transfer statistic are reported only when changed.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - disable) |
| *interval* | • Interval in seconds.<br>• ignored when disabling, should be non-zero when enabling<br>• period only affect transfer statistic attributes |
| *instanceid* | • PDP instance id 0 - First PDP instance 1 - Second PDP instance 2 - Third PDP instance |
| *ipfamily* | • 4 for an IPv4 data session<br>• 6 for an IPv6 data session<br>• 7 for an IPv4v6 data session |

**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 Currently 3 PDP instances are supported in device. User of this callback can subscribe by passing instance id of particular instance. All PDP instance can be subscribed by passing instance id sequentially.

#### 9.12.5.92 ULONG SLQSSwiAVMSEventReportCallback ( tFNSwiAvmEventReport *pCallback* )

#### 9.12.5.93 ULONG SLQSTmdMitigationLvlRptCallback ( TmdMitigationLvlIndReq * *req,* tFNMitiLvlRpt *pCallback* )

Thermal Mitigation callback.

**Parameters**

| | |
|---|---|
| *req* | • See TmdMitigationLvlIndReq for more information. |
| *pCallback* | • See tFNMitiLvlRpt for more information. |

#### 9.12.5.94 ULONG SLQSUIMSetRefreshCallBack ( tFNUIMRefresh *pCallback* )

Enables/disables the UIM refresh callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable UIM Refresh Indication notification<br>• NULL to disable Band Preference notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Timeout: 2 seconds
SLQSUIMRefreshRegister() API should be invoked prior to the invocation of the callback for the events to be registered.

#### 9.12.5.95 ULONG SLQSUIMSetStatusChangeCallBack ( tFNUIMStatusChangeInfo *pCallback* )

Enables/disables the UIM Status Change Callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable UIM Status Change Indication notification<br>• NULL to disable Band Preference notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Timeout: 2 seconds

#### 9.12.5.96 ULONG SLQSVoiceInfoRecCallback ( tFNInfoRec *pCallback* )

Enables/disables the Voice information Record callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs. (Applicable only for 3GPP2)

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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
Timeout: 2 seconds

#### 9.12.5.97 ULONG SLQSVoiceSetAllCallStatusCallBack ( tFNAllCallStatus *pCallback* )

Enables/disables Voice Call Status Callback function. User can subscribe this callback get the call state change notifications. eg:- Call originated, connected, or ended. Whenever there is a change in the call information, there will be a indication with the information.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0 - Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.5.98 ULONG SLQSVoiceSetDTMFEventCallBack ( tFNDTMFEvent *pCallback* )

Enables/disables the DTMF Event callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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
This callback communicates that a DTMF event has been received.

#### 9.12.5.99 ULONG SLQSVoiceSetOTASPStatusCallBack ( tFNOTASPStatus *pCallback* )

Enables/disables OTASP(Over-The-Air Service Provisioning) or OTAPA(Over-The-Air Parameter Administration) event CallBack Function (applicable only for 3GPP2). The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • a valid function pointer to enable OTASP or OTAPA event Indication notification<br>• NULL to disable OTASP or OTAPA event, Indication notification |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

Technology Supported: CDMA
Timeout: 10 seconds
This indication communicates the occurrence of an OTASP or OTAPA event. This indication is only applicable for 3GPP2 devices.

#### 9.12.5.100 ULONG SLQSVoiceSetPrivacyChangeCallBack ( tFNPrivacyChange *pCallback* )

Enables/disables the voice privacy change callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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
Timeout: 2 seconds
This callback communicates a change in the voice privacy of a call. This is applicable only in 3GPP2 devices.

#### 9.12.5.101 ULONG SLQSVoiceSetSUPSCallBack ( tFNSUPSInfo *pCallback* )

Enables/disables the SUPS callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: GSM
Timeout: 2 seconds
This callback notifies clients about the modem-originated supplementary service requests and the responses received from the network.

#### 9.12.5.102 ULONG SLQSVoiceSetSUPSNotificationCallback ( tFNSUPSNotification *pCallback* )

Enables/disables the supplementary service notification callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.103 ULONG SLQSWmsAsyncRawSendCallBack ( tFNAsyncRawSend *pCallback* )

Enables/disables the SLQSWmsAsyncRawSendCallBack callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**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

#### 9.12.5.104 ULONG SLQSWmsMemoryFullCallBack ( tFNMemoryFull *pCallback* )

Enables/disables the event related to memory full status callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.5.105 ULONG SLQSWmsMessageWaitingCallBack ( tFNMessageWaiting *pCallback* )

Enables/disables the event related to message waiting information callback function. The most recent successfully subscribed callback function will be the only function that is invoked when the corresponding event occurs.

**Parameters**

| | |
|---|---|
| *pCallback[IN]* | • Callback function pointer (0-Disable) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.13 qaGobiApiDcs.h File Reference

Device Connectivity Service API function prototypes.

### Data Structures

- struct DcsUsbPortNames
- struct QosMap
- struct NetStats

### Macros

- #define LEN 10
- #define PORTNAM_LEN 32

### Functions

- ULONG QCWWAN2kEnumerateDevices (BYTE *pDevicesSize, BYTE *pDevices)
- ULONG QCWWAN2kConnect (CHAR *pDeviceID, CHAR *pDeviceKey)
- ULONG QCWWANDisconnect ()
- ULONG QCWWAN2kGetConnectedDeviceID (ULONG deviceIDSize, CHAR *pDeviceID, ULONG deviceKeySize, CHAR *pDeviceKey)
- ULONG QCWWANEnumerateDevices (BYTE *pDevicesSize, BYTE *pDevices)
- ULONG QCWWANConnect (CHAR *pDeviceID, CHAR *pDeviceKey)
- ULONG SetSDKImagePath (LPCSTR pPath)
- ULONG SLQSGetUsbPortNames (struct DcsUsbPortNames *pUsbPortNames)
- ULONG SLQSStart_AVAgent (BYTE modem_index)
- ULONG SLQSStart (BYTE modem_index, CHAR *sn)
- ULONG SLQSKillSDKProcess ()
- ULONG SLQSGetDeviceMode (BYTE *pDeviceMode)
- ULONG SLQSStartSrv (BYTE action, BYTE mask)
- ULONG SLQSSetLoggingMask (BYTE mask)
- ULONG SLQSGetNetStatistic (struct NetStats *pNetStatistic, BYTE instance)
- ULONG GetQmapIsSupported (BYTE *pQmapsupport)
- ULONG SetQmapIPTable (struct QmuxIPTable *pQTable)
- ULONG SetQmapIPv6Table (QmuxIPv6Table *pQTable)

## 9.13.1 Detailed Description

Device Connectivity Service API function prototypes.

## 9.13.2 Macro Definition Documentation

### 9.13.2.1 #define LEN 10

### 9.13.2.2 #define PORTNAM_LEN 32

This structure contains the SLQSGetUsbPortNames Information

**Parameters**

| | |
|---|---|
| *AtCmdPort* | [OUT]<br>• Name of AT command port |
| *NmeaPort* | [OUT]<br>• Name of NMEA port |
| *DmPort* | [OUT]<br>• Name of DM port |

**Note**

Technology Supported: UMTS/CDMA
Device Supported: MC83x5, MC7700/10/50
Timeout: 2 seconds
Port names are limited to 32 characters.

### 9.13.3 Function Documentation

#### 9.13.3.1 ULONG GetQmapIsSupported ( BYTE * *pQmapsupport* )

Returns if the module supports QMAP feature.

**Parameters**

| | |
|---|---|
| *pQmapsupport[-OUT]* | • value<br>– 0 means module does not support QMAP, otherwise QMAP is supported |

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.13.3.2 ULONG QCWWAN2kConnect ( CHAR * *pDeviceID,* CHAR * *pDeviceKey* )

Connects the Connection Manager API to the first detected QC WWAN device. This function MUST be called after QCWWAN2kEnumerateDevices has been called.

**Parameters**

| | |
|---|---|
| *pDeviceID[IN]* | • Device path pertaining to the device for which the API is being invoked e.g. /dev/qcqmi0. |
| *pDeviceKey[IN]* | • Device key pertaining to the device for which the API is being invoked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_ERR_NO_DEVICE otherwise

**Note**

Timeout: 2 seconds

#### 9.13.3.3 ULONG QCWWAN2kEnumerateDevices ( BYTE * *pDevicesSize,* BYTE * *pDevices* )

Enumerates the QC WWAN devices currently attached to the host. This API MUST be called before any other API.

**Parameters**

| | |
|---|---|
| *pDeviceSize[IN/-OUT]* | • Upon input, maximum number of elements that the device array can contain.<br>• Upon successful output, actual number of elements in the device array. |
| *pDevices[IN/OUT]* | • Device array; array elements are structures with the following elements:<br>CHAR deviceID[256] - Device path (e.g. /dev/qcqmi0)<br>CHAR deviceKey[16] - Device key stored in the device (e.g. A1000004B01051) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_ERR_NO_DEVICE on otherwise

**Note**

Timeout: 2 seconds

#### 9.13.3.4 ULONG QCWWAN2kGetConnectedDeviceID ( ULONG *deviceIDSize,* CHAR * *pDeviceID,* ULONG *deviceKeySize,* CHAR * *pDeviceKey* )

Returns the device ID and device key of the currently connected QC WWAN device.

**Parameters**

| | |
|---|---|
| *deviceIDSize* | • Maximum number of characters (including NULL terminator) that the device ID array can contain. |
| *pDeviceID[OUT]* | • Device path string |
| *deviceKeySize* | • Maximum number of characters (including NULL terminator) that the device key array can contain. |
| *pDeviceKey[OUT]* | • Device key string |

**Returns**

eQCWWAN_ERR_NONE if device found, eQCWWAN_ERR_NO_DEVICE otherwise

**Note**

Timeout: 2 seconds

#### 9.13.3.5 ULONG QCWWANConnect ( CHAR * *pDeviceID,* CHAR * *pDeviceKey* )

Enumerates the QC WWAN devices currently attached to the host. This API MUST be called before any other API.

**Parameters**

| | |
|---|---|
| *pDeviceID[IN]* | • Device path pertaining to the device for which the API is being invoked e.g. /dev/qcqmi0. |
| *pDeviceKey[IN]* | • Device key pertaining to the device for which the API is being invoked |

**Returns**

eQCWWAN_ERR_NONE if device found, eQCWWAN_ERR_NO_DEVICE otherwise

**Note**

Timeout: 2 seconds
This API is deprecated; use QCWWAN2kConnect() instead

#### 9.13.3.6 ULONG QCWWANDisconnect ( )

Disconnects the Connection Manager API from a previously connected QC device. This function de-registers all the callback functions that have been registered.

**Parameters**

| | |
|---|---|
| *none* | |

**Returns**

eQCWWAN_ERR_NONE

**Note**

Timeout: 2 seconds

#### 9.13.3.7 ULONG QCWWANEnumerateDevices ( BYTE * *pDevicesSize,* BYTE * *pDevices* )

Enumerates the QC WWAN devices currently attached to the host. This API is deprecated; use QCWWAN2kEnumerateDevices() instead.

**Parameters**

| | |
|---|---|
| *pDeviceSize[IN/OUT]* | • Upon input, maximum number of elements that the device array can contain.<br>• Upon successful output, actual number of elements in the device array. |
| *pDevices[IN/OUT]* | • Device array; array elements are structures with the following elements:<br>CHAR deviceID[256] - Device path (e.g. /dev/qcqmi0)<br>CHAR deviceKey[16] - Device key stored in the device |

**Returns**

eQCWWAN_ERR_NONE

**Note**

Timeout: 2 seconds
This API must be called prior to any other APIs.

#### 9.13.3.8 ULONG SetQmapIPTable ( struct QmuxIPTable * *pQTable* )

Set the mux id associated with ip address to the SDK process, it will be propagated to driver by IOCTL

**Parameters**

| | |
|---|---|
| *pQTable[IN]* | • See QmuxIPTable for more details |

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.13.3.9 ULONG SetQmapIPv6Table ( QmuxIPv6Table * *pQTable* )

Set the mux id associated with ipv6 address to the SDK process, it will be propagated to driver by IOCTL

**Parameters**

| | |
|---|---|
| *pQTable[IN]* | • See QmuxIPv6Table for more details |

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.13.3.10 ULONG SetSDKImagePath ( LPCSTR *pPath* )

Set the location of the SLQS executable

**Parameters**

| | |
|---|---|
| *pPath[IN]* | - Pointer to fully qualified path of SLQS executable (includes the executable file's name) |

**Returns**

eQCWWAN_ERR_NONE

**Note**

Timeout: None

#### 9.13.3.11 ULONG SLQSGetDeviceMode ( BYTE * *pDeviceMode* )

Returns the Device Mode

**Parameters**

| | |
|---|---|
| *pDeviceMode[OUT]* | • Pointer to SLQS Device Mode of type eDevState |

**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

#### 9.13.3.12 ULONG SLQSGetNetStatistic ( struct NetStats * *pNetStatistic,* BYTE *instance* )

Returns the usbnet statistics for a particular PDN.

**Parameters**

| | | |
|---|---|---|
| | *pNetStatistic[OUT]* | • Pointer to the structure NetStats which the value of every member is to be retrieved |
| in | *instance* | • PDP Instance id |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.13.3.13 ULONG SLQSGetUsbPortNames ( struct DcsUsbPortNames * *pUsbPortNames* )

Returns the Usb Port Names currently in use.

**Parameters**

| | |
|---|---|
| *pUsbPort-Names[OUT]* | • Pointer to SLQS USB Port Names 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

#### 9.13.3.14 ULONG SLQSKillSDKProcess ( )

Kill the SDK process

**Parameters**

| | |
|---|---|
| *none* | |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: N/A
This API useful if the application was started with non-root privileges as subsequent attempt to start any application will fail because the SDK requires root permission to access /dev/qcqmi device special files.

#### 9.13.3.15 ULONG SLQSSetLoggingMask ( BYTE *mask* )

Limit Syslog messages according to the Mask provided by user

**Parameters**

| | |
|---|---|
| *mask* | • Mask 0x01: disable all log<br>• Mask 0xFF: enable all log |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.13.3.16 ULONG SLQSStart ( BYTE *modem_index,* CHAR * *sn* )

Create the SDK process and IPC sockets for the Application and SDK processes to communicate over.

**Parameters**

| | | |
|---|---|---|
| in | *modem_index* | • 0: first modem detected<br>• 1: second modem detected<br>• 2: third modem detected<br>• ...<br>• 7: seventh modem detected |
| in | *sn* | This field is optional, it can be serial number or usb path for multiple modem feature, it can be retrieved from sytem command "dmesg" when specified, the modem_index will be mapping to sn or usb path Please set to NULL when not used |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: N/A

#### 9.13.3.17 ULONG SLQSStart_AVAgent ( BYTE *modem_index* )

Create IPC sockets for AirVantage Agent and SDK processes to communicate over

**Parameters**

| | |
|---|---|
| *none* | |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: N/A

#### 9.13.3.18 ULONG SLQSStartSrv ( BYTE *action*, BYTE *mask* )

Registers/deregisters for service with unsolicited notifications

**Parameters**

| | |
|---|---|
| *action,1* | for register, 0 for deregister |
| *mask* | • Bit mask for unsolicited notifications<br>– Bit0 - WDS<br>– Bit1 - NAS<br>– Bit2 - PDS<br>– Bit3 - VOICE |

**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 API is useful to register for the services which supports unsolicited notifications. Registration/deregistration can be done by using parameter action if action is set then the mask (set bits) will be used for registering service and if action is "0" mask(set bits) will be used to deregister services. For example : bit mask 0x03 - Registers for services WDS and NAS if action is "1" and deregisters WDS and NAS if action is "0".

## 9.14 qaGobiApiDms.h File Reference

Device Management Service API function prototypes.

### Data Structures

- struct serialNumbersInfo
- struct ERIFileparams
- struct custFeaturesInfo
- struct custFeaturesSetting
- struct dmsCurrentPRLInfo
- struct FactorySequenceNumber
- struct FirmwareUpdatStat
- struct USBCompParams

- struct USBCompConfig
- struct CrashInfo
- struct CrashInfoParams
- struct _SLQSSwiGetHostDevInfoParams
- struct _SLQSSwiSetHostDevInfoParams
- struct _SLQSSwiGetOSInfoParams
- struct _SLQSSwiSetOSInfoParams
- struct _SLQSSwiGetSerialNoExtParams
- struct setCustomSettingV2
- struct getCustomInput
- struct custSettingInfo
- struct custSettingList
- struct getCustomFeatureV2
- struct getDyingGaspCfg
- struct setDyingGaspCfg
- struct getDyingGaspStatistics
- struct dmsIndicationRegisterReq
- struct dmsSwiGetResetInfo
- struct BandCapabilityResp
- struct dmsSwiGetPCInfo
- struct PSMEnableStateTlv
- struct PSMDurationThresholdTlv
- struct PSMDurationDueToOOSTlv
- struct PSMRandomizationWindowTlv
- struct PSMActiveTimerTlv
- struct PSMPeriodicUpdateTimerTlv
- struct PSMEarlyWakeupTimeTlv
- struct PowerSaveModeConfigResp
- struct PowerSaveModeConfigReq

## Macros

- #define MAX_FSN_LENGTH 255
- #define MAX_CUST_ID_LEN 64
- #define MAX_CUST_VALUE_LEN 8
- #define MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160
- #define MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20

## Typedefs

- typedef struct serialNumbersInfo serialNumbersInfo
- typedef struct ERIFileparams ERIFileparams
- typedef struct custFeaturesInfo custFeaturesInfo
- typedef struct custFeaturesSetting custFeaturesSetting
- typedef struct dmsCurrentPRLInfo dmsCurrentPRLInfo
- typedef struct _SLQSSwiGetHostDevInfoParams SLQSSwiGetHostDevInfoParams
- typedef struct _SLQSSwiSetHostDevInfoParams SLQSSwiSetHostDevInfoParams
- typedef struct _SLQSSwiGetOSInfoParams SLQSSwiGetOSInfoParams
- typedef struct _SLQSSwiSetOSInfoParams SLQSSwiSetOSInfoParams
- typedef struct _SLQSSwiGetSerialNoExtParams SLQSSwiGetSerialNoExtParams

## Functions

- ULONG GetManufacturer (BYTE stringSize, CHAR *pString)
- ULONG GetModelID (BYTE stringSize, CHAR *pString)
- ULONG GetFirmwareRevision (BYTE stringSize, CHAR *pString)
- ULONG GetFirmwareRevisions (BYTE amssSize, CHAR *pAMSSString, BYTE bootSize, CHAR *pBootString, BYTE priSize, CHAR *pPRIString)
- ULONG GetPRLVersion (WORD *pPRLVersion)
- ULONG GetIMSI (BYTE stringSize, CHAR *pString)
- ULONG GetSerialNumbers (BYTE esnSize, CHAR *pESNString, BYTE imeiSize, CHAR *pIMEIString, BYTE meidSize, CHAR *pMEIDString)
- ULONG SLQSGetSerialNumbers (serialNumbersInfo *pSerialNumbersInfo)
- ULONG GetHardwareRevision (BYTE stringSize, CHAR *pString)
- ULONG GetNetworkTime (ULONGLONG *pTimeStamp, ULONG *pTimeSource)
- ULONG UIMSetPINProtection (ULONG id, ULONG bEnable, CHAR *pValue, ULONG *pVerifyRetriesLeft, ULONG *pUnblockRetriesLeft)
- ULONG UIMUnblockPIN (ULONG id, CHAR *pPUKValue, CHAR *pNewValue, ULONG *pVerifyRetriesLeft, ULONG *pUnblockRetriesLeft)
- ULONG UIMVerifyPIN (ULONG id, CHAR *pValue, ULONG *pVerifyRetriesLeft, ULONG *pUnblockRetriesLeft)
- ULONG UIMChangePIN (ULONG id, CHAR *pOldValue, CHAR *pNewValue, ULONG *pVerifyRetriesLeft, ULONG *pUnblockRetriesLeft)
- ULONG GetVoiceNumber (BYTE voiceNumberSize, CHAR *pVoiceNumber, BYTE minSize, CHAR *pMIN)
- ULONG SetPower (ULONG powerMode)
- ULONG GetPower (ULONG *pPowerMode)
- ULONG UIMGetControlKeyStatus (ULONG id, ULONG *pStatus, ULONG *pVerifyRetriesLeft, ULONG *pUnblockRetriesLeft)
- ULONG UIMGetICCID (BYTE stringSize, CHAR *pString)
- ULONG UIMGetPINStatus (ULONG id, ULONG *pStatus, ULONG *pVerifyRetriesLeft, ULONG *pUnblockRetriesLeft)
- ULONG GetOfflineReason (ULONG *pReasonMask, ULONG *pbPlatform)
- ULONG UIMSetControlKeyProtection (ULONG id, ULONG status, CHAR *pValue, ULONG *pVerifyRetriesLeft)
- ULONG UIMUnblockControlKey (ULONG id, CHAR *pValue, ULONG *pUnblockRetriesLeft)
- ULONG GetDeviceCapabilities (ULONG *pMaxTXChannelRate, ULONG *pMaxRXChannelRate, ULONG *pDataServiceCapability, ULONG *pSimCapability, ULONG *pRadioIfacesSize, BYTE *pRadioIfaces)
- ULONG ResetToFactoryDefaults (CHAR *pSPC)
- ULONG ValidateSPC (CHAR *pSPC)
- ULONG ActivateAutomatic (CHAR *pActivationCode)
- ULONG SLQSGetERIFile (ERIFileparams *pERIFileparams)
- ULONG GetActivationState (ULONG *pActivationState)
- ULONG SLQSUIMGetState (ULONG *pUIMState)
- ULONG SLQSGetBandCapability (ULONGLONG *pBandCapability)
- ULONG SLQSGetCustFeatures (custFeaturesInfo *pCustFeaturesInfo)
- ULONG SLQSSetCustFeatures (custFeaturesSetting *pCustFeaturesSetting)
- ULONG SLQSGetCurrentPRLInfo (dmsCurrentPRLInfo *pCurrentPRLInfo)
- ULONG SLQSSwiGetFSN (FactorySequenceNumber *pFSNumber)
- ULONG SLQSSwiGetFwUpdateStatus (FirmwareUpdatStat *pFirmwareUpdatStat)
- ULONG SLQSSwiGetUSBComp (USBCompParams *pUSBCompParams)
- ULONG SLQSSwiSetUSBComp (USBCompConfig *pUSBCompConfig)
- ULONG SLQSSwiGetCrashInfo (BYTE *pClear, CrashInfoParams *pCrashInfoParams)
- ULONG SLQSSwiGetCrashAction (BYTE *pDevCrashState)
- ULONG SLQSSwiSetCrashAction (BYTE crashActionParams)
- ULONG SLQSSwiGetHostDevInfo (SLQSSwiGetHostDevInfoParams *pGetHostDevInfoParams)
- ULONG SLQSSwiSetHostDevInfo (SLQSSwiSetHostDevInfoParams *pSetHostDevInfoParams)

- ULONG SLQSSwiGetOSInfo (SLQSSwiGetOSInfoParams *pParams)
- ULONG SLQSSwiSetOSInfo (SLQSSwiSetOSInfoParams *pParams)
- ULONG SLQSSwiGetSerialNoExt (SLQSSwiGetSerialNoExtParams *pParams)
- ULONG SLQSSetCustFeaturesV2 (setCustomSettingV2 *pSetCustSetting)
- ULONG SLQSGetCustFeaturesV2 (getCustomFeatureV2 *pGetCustomFeatureV2)
- ULONG SLQSSwiGetDyingGaspCfg (getDyingGaspCfg *pConfig)
- ULONG SLQSSwiSetDyingGaspCfg (setDyingGaspCfg *pConfig)
- ULONG SLQSSwiGetDyingGaspStatistics (getDyingGaspStatistics *pStatistics)
- ULONG SLQSSwiClearDyingGaspStatistics ()
- ULONG SLQSDmsSwiIndicationRegister (dmsIndicationRegisterReq *pIndicationRegisterReq)
- ULONG SLQSDmsSwiGetResetInfo (dmsSwiGetResetInfo *pGetResetInfoResp)
- ULONG SLQSGetBandCapabilities (BandCapabilityResp *pBandCapability)
- ULONG SLQSDmsSwiGetPCInfo (dmsSwiGetPCInfo *pGetPCInfoResp)
- ULONG SLQSGetPowerSaveModeConfig (PowerSaveModeConfigResp *pPSMConfigParam)
- ULONG SLQSSetPowerSaveModeConfig (PowerSaveModeConfigReq *pPSMConfigParamReq)

### 9.14.1 Detailed Description

Device Management Service API function prototypes.

### 9.14.2 Macro Definition Documentation

#### 9.14.2.1 #define MAX_CUST_ID_LEN 64

#### 9.14.2.2 #define MAX_CUST_VALUE_LEN 8

#### 9.14.2.3 #define MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160

#### 9.14.2.4 #define MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20

#### 9.14.2.5 #define MAX_FSN_LENGTH 255

### 9.14.3 Typedef Documentation

#### 9.14.3.1 typedef struct custFeaturesInfo custFeaturesInfo

This structure contains current settings of custom features

**Parameters**

| | |
|---|---|
| *GpsEnable[OUT]* | • describes if GPS is enabled or disabled<br>• values:<br>– 0x00 - GPS is disabled<br>– 0x01 - GPS is enabled<br>• function SLQSGetCustFeatures() returns a default value FFFFFFFF if no value is returned by the modem |

| | |
|---|---|
| *pDisableIMSI[OUT]* | • 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>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem |
| *pIPFamSupport[OUT]* | • optional 2 byte BitMask<br>• bitmask representing the IP families supported<br>• values:<br>– 0x01 - IPv4<br>– 0x02 - IPv6<br>– 0x04 - IPv4v6<br>• function SLQSGetCustFeatures() returns a default value FFFF if no value is returned by the modem |
| *pRMAutoConnect[OUT]* | • optional 1 byte parameter<br>• QMI Mode RM Net Auto Connect Support<br>• values:<br>– 0x00 - Not Supported<br>– 0x01 - Supported<br>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem |
| *pGPSSel[OUT]* | • 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>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem |
| *pSMSSupport[OUT]* | • 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<br>• function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem. In this case assume, SMS is supported. |

| | |
|---|---|
| *pIsVoice-Enabled[OUT]* | • 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 |
| *pDHCPRelay-Enabled[OUT]* | • optional 1 byte parameter<br>• DHCP Relay support<br>• values:<br>– 0x00 - Disable DHCP relay<br>– 0x01 - Enable DHCP relay |
| *pGPSLPM[OUT]* | • 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 |

#### 9.14.3.2 typedef struct custFeaturesSetting custFeaturesSetting

This structure contains settings to be used for custom features

**Parameters**

| | |
|---|---|
| *pGPSSel* | • 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 |
| *pGPSEnable* | • optional 4 byte parameter<br>• GPS Enable/Disable<br>• values: The value of 7 least significant bits:<br>– 0 - Disabled<br>– 1 - MT & MO enabled<br>– 2 - MO enabled<br>– 3 - MT enabled<br>– 4 - MT & MO enabled if GPS_DISABLE pin is not asserted<br>– 5 - MO GPS enabled if GPS_DISABLE pin is not asserted<br>– 6 - MT GPS enabled if GPS_DISABLE pin is not asserted |

**Note**

Only MC7750 3.5.x firmware supports above 0x04, 0x05 and 0x06 settings. To disable GLONASS, set the most significant bit - 0x80. This setting is only applicable if GPS is not Disabled.

**Parameters**

| | |
|---|---|
| *pIsVoiceEnabled* | • optional 1 byte parameter<br>• voice enabled/disabled<br>• values:<br>– 0 - Enable voice on both AT and QMI interface (default)<br>– 1 - Reserved<br>– 2 - Disable voice on both AT and QMI interface |
| *pDHCPRelay-Enabled* | • optional 1 byte parameter<br>• DHCPRELAYENABLE support<br>• values:<br>– 0 - Disable DHCP relay<br>– 1 - Enable DHCP relay |
| *pGPSLPM* | • optional 1 byte parameter<br>• GPSLPM support<br>• values:<br>– 0 - Enable GPS in Low Power Mode<br>– 1 - Disable GPS in Low Power Mode |

#### 9.14.3.3 typedef struct dmsCurrentPRLInfo dmsCurrentPRLInfo

This structure contains GetCurrentPRLInfo response parameter

**Parameters**

| | |
|---|---|
| *pPRLVersion[O-UT]* | - Optional<br>• PRL version of device. |
| *pPRLPreference* | [OUT]- Optional<br>• PRL Preference<br>– 0 - Unset<br>– 1 - Set |

#### 9.14.3.4 typedef struct ERIFileparams ERIFileparams

This structure contains Extended Roaming Indicator(ERI) file parameters

**Parameters**

| | |
|---|---|
| *pFileSize[IN/OUT]* | • 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 |
| *pFile[OUT]* | • ERI data read from persistent storage( Max size is 1024 ) |

#### 9.14.3.5 typedef struct serialNumbersInfo serialNumbersInfo

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity) and MEID (Mobile Equipment Identifier).

**Parameters**

| | |
|---|---|
| *esnSize* | • The maximum number of characters (including NULL terminator) that the ESN string array can contain |
| *pESNString[OUT]* | • NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed |
| *imeiSize* | • The maximum number of characters (including NULL terminator) that the IMEI string array can contain |
| *pIMEIString[OUT]* | • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed |
| *meidSize* | • The maximum number of characters (including NULL terminator) that the MEID string array can contain |
| *pMEIDString[OUT]* | • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed |
| *imeiSvnSize* | • The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain. |
| *pImeiSvnString[OUT]* | • NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed. |

#### 9.14.3.6 typedef struct _SLQSSwiGetHostDevInfoParams SLQSSwiGetHostDevInfoParams

This structure is used to Get Host Device Information

**Parameters**

| | |
|---|---|
| *bManSize[IN/OUT]* | • Host Device Manufacturer String Size |
| *pManString[OUT]* | • Host Device Manufacturer Name(Optional parameter)<br>• Null terminated ASCII String |
| *bModelSize[IN/OUT]* | • Host Device Model String Size |
| *pModelString[OUT]* | • Host Device Model String(Optional parameter)<br>• Null terminated ASCII string. |
| *bSWVerSize[IN/OUT]* | • Host Device Software Version String Size |
| *pSWVerString[OUT]* | • Host Device Software Version String(Optional parameter)<br>• Null terminated ASCII string |
| *bPlasmaIDSize[IN/OUT]* | • Host Device Plasma ID String Size |
| *pPlasmaIDString[OUT]* | • Host Device Plasma ID String(Optional parameter)<br>• Null terminated alphanumeric ASCII String. |

#### 9.14.3.7 typedef struct _SLQSSwiGetOSInfoParams SLQSSwiGetOSInfoParams

This structure is used to Get OS Information

**Parameters**

| | |
|---|---|
| *bNameSize[IN/OUT]* | • Size of Operating System Name |
| *pNameString[OUT]* | • Operating System Name(Optional parameter)<br>• Null terminated ASCII string |
| *bVersionSize[IN/OUT]* | • Operating System Version Size |
| *pVersionString[OUT]* | • Operating System Version String(Optional parameter)<br>• Null terminated ASCII string. |

#### 9.14.3.8 typedef struct _SLQSSwiGetSerialNoExtParams SLQSSwiGetSerialNoExtParams

This structure is used to store MEID Information

**Parameters**

| | |
|---|---|
| *meidLength[OUT]* | • String length of the of MEID received |
| *pMeidString[OUT]* | • Optional parameter<br>• Pointer to receive String containing the Mobile Equipment Identifier(MEID) of the device. |

#### 9.14.3.9 typedef struct _SLQSSwiSetHostDevInfoParams SLQSSwiSetHostDevInfoParams

This structure is used to Set Host Device Information

**Parameters**

| | |
|---|---|
| *bManSize[IN]* | • Host Device Manufacturer String Size |
| *pManString[IN]* | • Host Device Manufacturer Name(Optional parameter)<br>• Null terminated ASCII String |
| *bModelSize[IN]* | • Host Device Model String Size |
| *pModelString[IN]* | • Host Device Model String(Optional parameter)<br>• Null terminated ASCII string. |
| *bSWVerSize[IN]* | • Host Device Software Version String Size |
| *pSWVerString[IN]* | • Host Device Software Version String(Optional parameter)<br>• Null terminated ASCII string |
| *bPlasmaIDSize[IN]* | • Host Device Plasma ID String Size |
| *pPlasmaIDString[IN]* | • Host Device Plasma ID String(Optional parameter)<br>• Null terminated alphanumeric ASCII String. |

#### 9.14.3.10 typedef struct _SLQSSwiSetOSInfoParams SLQSSwiSetOSInfoParams

This structure is used to Set OS Information

**Parameters**

| | |
|---|---|
| *bNameSize[IN]* | • Size of Operating System Name |
| *pNameString[IN]* | • Operating System Name(Optional parameter)<br>• Null terminated ASCII string |
| *bVersionSize[IN]* | • Operating System Version Size |
| *pVersionString[IN]* | • Operating System Version String(Optional parameter)<br>• Null terminated ASCII string. |

### 9.14.4 Function Documentation

#### 9.14.4.1 ULONG ActivateAutomatic ( CHAR * *pActivationCode* )

Requests the device to perform automatic service activation

**Parameters**

| | |
|---|---|
| *pActivation-Code[IN]* | • NULL-terminated string representing activation code (maximum string length of 12); specific carrier requirements may dictate actual activation code that is applicable, e.g., "*22899" |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 Minutes

#### 9.14.4.2 ULONG GetActivationState ( ULONG * *pActivationState* )

Returns the device activation state. This API is only supported for 3GPP2 devices.

**Parameters**

| | |
|---|---|
| *pActivation-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 |

**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
Timeout: 2 Seconds

#### 9.14.4.3 ULONG GetDeviceCapabilities ( ULONG * *pMaxTXChannelRate,* ULONG * *pMaxRXChannelRate,* ULONG * *pDataServiceCapability,* ULONG * *pSimCapability,* ULONG * *pRadioIfacesSize,* BYTE * *pRadioIfaces* )

Gets the device capabilities

**Parameters**

| | |
|---|---|
| *pMaxTX-ChannelRate[OUT]* | • 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 |
| *pMaxRX-ChannelRate[OUT]* | • 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 |
| *pDataService-Capability[OUT]* | • 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 |

| | |
|---|---|
| *pSimCapability[-OUT]* | • Device SIM capability<br>– 0 - SIM not supported<br>– 1 - SIM supported |
| *pRadioIfaces-Size[IN/OUT]* | • 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 |
| *pRadioIfaces[O-UT]* | • Radio interface array. This is a structure of array containing the elements below. ULONG radioInterface<br>– See qaGobiApiTableRadioInterfaces.h for Radio Interfaces |

**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

#### 9.14.4.4 ULONG GetFirmwareRevision ( BYTE *stringSize,* CHAR * *pString* )

Returns the device firmware revision

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain |
| *pString[OUT]* | • NULL terminated string |

**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

#### 9.14.4.5 ULONG GetFirmwareRevisions ( BYTE *amssSize,* CHAR * *pAMSSString,* BYTE *bootSize,* CHAR * *pBootString,* BYTE *priSize,* CHAR * *pPRIString* )

Returns the device firmware revisions (AMSS, boot, and PRI)

**Parameters**

| | |
|---|---|
| *amssSize* | • Maximum number of characters (including NULL terminator) that the AMSS string array can contain |
| *pAMSSString[OUT]* | • NULL-terminated AMSS revision string |
| *bootSize* | • Maximum number of characters (including NULL terminator) that the boot string array can contain |
| *pBootString[OUT]* | • NULL-terminated boot code revision string |
| *priSize* | • Maximum number of characters (including NULL terminator) that the PRI string array can contain |
| *pPRIString[OUT]* | • NULL-terminated PRI revision string |

**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

#### 9.14.4.6 ULONG GetHardwareRevision ( BYTE *stringSize,* CHAR * *pString* )

Returns the hardware revision of the device

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain |
| *pString[OUT]* | • NULL terminated string |

**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

#### 9.14.4.7 ULONG GetIMSI ( BYTE *stringSize,* CHAR * *pString* )

Returns the device IMSI. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use SLQSUIMReadTransparent()(EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain |
| *pString[OUT]* | • NULL terminated string |

**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

#### 9.14.4.8 ULONG GetManufacturer ( BYTE *stringSize,* CHAR * *pString* )

Returns the device manufacturer name

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain. |
| *pString[OUT]* | • NULL terminated string |

**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

#### 9.14.4.9 ULONG GetModelID ( BYTE *stringSize,* CHAR * *pString* )

Returns the device model ID

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain |
| *pString[OUT]* | • NULL terminated string |

**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

#### 9.14.4.10 ULONG GetNetworkTime ( ULONGLONG * *pTimeStamp,* ULONG * *pTimeSource* )

Returns the current time of the device based on the value supported by the network.

**Parameters**

| | |
|---|---|
| *pTimeStamp[OUT]* | • Count of 1.25 ms that have elapsed from the start of GPS time (Jan 6, 1980) |
| *pTimeSource[OUT]* | • Source of timestamp<br>0 - 32 kHz device clock<br>1 - CDMA network<br>2 - cdma2000 1xEV-DO network |

**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
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.

#### 9.14.4.11 ULONG GetOfflineReason ( ULONG * *pReasonMask,* ULONG * *pbPlatform* )

Returns 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 |
| *pbPlatform[OUT]* | • Optional parameter<br>• Is the device offline due to a platform restriction?<br>– 0 - No<br>– 1 - Yes |

**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

#### 9.14.4.12 ULONG GetPower ( ULONG * *pPowerMode* )

Returns the operating mode of the device.

**Parameters**

| | |
|---|---|
| *pPowerMode[OUT]* | • Selected operating mode<br>– See qaGobiApiTablePowerModes.h for power modes |

**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

#### 9.14.4.13 ULONG GetPRLVersion ( WORD * *pPRLVersion* )

Returns the version of the active Preferred Roaming List (PRL) in use by the device.

**Parameters**

| | |
|---|---|
| *pPRLVersion[OUT]* | • PRL version number |

**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

#### 9.14.4.14 ULONG GetSerialNumbers ( BYTE *esnSize*, CHAR * *pESNString*, BYTE *imeiSize*, CHAR * *pIMEIString*, BYTE *meidSize*, CHAR * *pMEIDString* )

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity) and MEID (Mobile Equipment Identifier).

**Parameters**

| | |
|---|---|
| *esnSize* | • The maximum number of characters (including NULL terminator) that the ESN string array can contain |

| | |
|---|---|
| *pESNString[OUT]* | • NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed |
| *imeiSize* | • The maximum number of characters (including NULL terminator) that the IMEI string array can contain |
| *pIMEIString[OUT]* | • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed |
| *meidSize* | • The maximum number of characters (including NULL terminator) that the MEID string array can contain |
| *pMEIDString[OUT]* | • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

For CDMA devices that use a RUIM, the MEID of the RUIM (if present) will be returned. Use SLQSSwiGetSerialNoExt() to get MEID of CDMA modems. Timeout: 2 seconds

#### 9.14.4.15 ULONG GetVoiceNumber ( BYTE *voiceNumberSize,* CHAR * *pVoiceNumber,* BYTE *minSize,* CHAR * *pMIN* )

Returns the voice number in use by the device

**Parameters**

| | |
|---|---|
| *voiceNumberSize* | • Maximum number of characters (including NULL terminator) that the voice number array can contain. |
| *pVoiceNumber[OUT]* | • Voice number string: MDN or MS ISDN |
| *minSize* | • Maximum number of characters (including NULL terminator) that the MIN array can contain. |
| *pMIN[OUT]* | • Optional Parameter<br>• MIN string: Empty string returned when MIN is not supported/ programmed. |

**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

#### 9.14.4.16 ULONG ResetToFactoryDefaults ( CHAR * *pSPC* )

Resets to default factory settings of the device

**Parameters**

| | |
|---|---|
| *pSPC[IN]* | • NULL-terminated string representing a six-digit service programming code |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 minutes

#### 9.14.4.17 ULONG SetPower ( ULONG *powerMode* )

Sets the operating mode of the device.

**Parameters**

| | |
|---|---|
| *powerMode[IN]* | • Selected operating mode<br>– See qaGobiApiTablePowerModes.h for power modes |

**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

#### 9.14.4.18 ULONG SLQSDmsSwiGetPCInfo ( dmsSwiGetPCInfo * *pGetPCInfoResp* )

This function is used to get the device power control status information

**Parameters**

| | |
|---|---|
| *pGetPCInfoResp* | • See dmsSwiGetPCInfo for more information of the input structure |

#### 9.14.4.19 ULONG SLQSDmsSwiGetResetInfo ( dmsSwiGetResetInfo * *pGetResetInfoResp* )

This function is used to get reset info

**Parameters**

| | |
|---|---|
| *pGetResetInfo-Resp* | • See dmsSwiGetResetInfo for more information of the input structure |

#### 9.14.4.20 ULONG SLQSDmsSwiIndicationRegister ( dmsIndicationRegisterReq * *pIndicationRegisterReq* )

This function used to set Swi Indication Register

**Parameters**

| | |
|---|---|
| *pConfig* | • See dmsIndicationRegisterReq for more information of the inpurt structure |

#### 9.14.4.21 ULONG SLQSGetBandCapabilities ( BandCapabilityResp * *pBandCapability* )

Returns the band capability of the device.

**Parameters**

| | |
|---|---|
| *pBand-Capability[OUT]* | See BandCapabilityResp for more information of the inpurt structure |

**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

#### 9.14.4.22 ULONG SLQSGetBandCapability ( ULONGLONG * *pBandCapability* )

Returns the band capability of the device.

**Parameters**

| | |
|---|---|
| *pBand-Capability[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 |

**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

#### 9.14.4.23 ULONG SLQSGetCurrentPRLInfo ( dmsCurrentPRLInfo * *pCurrentPRLInfo* )

This API get the currently active PRL information of the device.

**Parameters**

| | |
|---|---|
| *pCurrentPRLInfo* | • Pointer to structure dmsCurrentPRLInfo<br>• See dmsCurrentPRLInfo 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
Timeout: 2 Secs

#### 9.14.4.24 ULONG SLQSGetCustFeatures ( custFeaturesInfo * *pCustFeaturesInfo* )

This API fetches the current settings of custom features. This API is deprecated for EM74xx/MC74xx, please use SLQSGetCustFeaturesV2() instead for EM74xx/MC74xx.

**Parameters**

| | |
|---|---|
| *pCustFeatures-Info* | • Structure containing settings of custom features.<br>• See custFeaturesInfo 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 Secs

#### 9.14.4.25 ULONG SLQSGetCustFeaturesV2 ( getCustomFeatureV2 * *pGetCustomFeatureV2* )

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**

| | |
|---|---|
| *pGetCustom-FeatureV2* | • See getCustomFeatureV2 for more information of the inpurt structure |

#### 9.14.4.26 ULONG SLQSGetERIFile ( ERIFileparams * *pERIFileparams* )

Returns the Extended Roaming Indicator (ERI) file that is stored in EFS on the device at a predetermined location. See the carrier requirements for specific details.

**Parameters**

| | |
|---|---|
| *pERIFileparams* | • Pointer to structure ERIFileparams<br>• See ERIFileparams 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
Timeout: 5 Seconds

#### 9.14.4.27 ULONG SLQSGetPowerSaveModeConfig ( PowerSaveModeConfigResp * *pPSMConfigParam* )

Returns the Power Save Mode (PSM) configuration parameters..

**Parameters**

| | |
|---|---|
| *pPSMConfig-Param[OUT]* | See PowerSaveModeConfigResp for more information of the input structure |

**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

#### 9.14.4.28 ULONG SLQSGetSerialNumbers ( serialNumbersInfo * *pSerialNumbersInfo* )

Returns all the serial numbers assigned to the device. These serial numbers include the ESN (Electronic serial number of the device), the IMEI (International Mobile Equipment Identity), MEID (Mobile Equipment Identifier) and IMEI SVN (IMEI software version number).

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

For CDMA devices that use a RUIM, the MEID of the RUIM (if present) will be returned. Use SLQSSwiGet-SerialNoExt() to get MEID of CDMA modems. Timeout: 2 seconds

#### 9.14.4.29 ULONG SLQSSetCustFeatures ( custFeaturesSetting * *pCustFeaturesSetting* )

This API changes the settings of custom features, a reset is required for any settings that are changed to take effect. This API is deprecated for EM74xx/MC74xx, please use SLQSSetCustFeaturesV2() for EM74xx/MC74xx.

**Parameters**

| | |
|---|---|
| *pCustFeatures-Setting[IN]* | • Structure containing settings of custom features.<br>• See custFeaturesSetting 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 Secs

#### 9.14.4.30 ULONG SLQSSetCustFeaturesV2 ( setCustomSettingV2 * *pSetCustSetting* )

This function sets the modem for a list of supported features. This function is for firmware version 2.0 and newer. Currently supported customization features:

- GPIOSARENABLE
- GPSENABLE
- GPSSEL
- ISVOICEEN
- DHCPRELAYENABLE
- GPSLPM
- SUBNETMSKENABLE
- IMSWITCHHIDE
- IPV6ENABLE
- WAKEHOSTEN
- IPCHANNELRATEEN
- Other Custom Settings

**Parameters**

| | |
|---|---|
| *pSetCustSetting* | • Optional parameter<br>• See setCustomSettingV2 for more information |

#### 9.14.4.31 ULONG SLQSSetPowerSaveModeConfig ( PowerSaveModeConfigReq * *pPSMConfigParamReq* )

Sets the Power Save Mode (PSM) configuration parameters.

**Parameters**

| | |
|---|---|
| *pPSMConfig-Param[OUT]* | See PowerSaveModeConfigResp for more information of the input structure |

**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

#### 9.14.4.32 ULONG SLQSSwiClearDyingGaspStatistics ( )

This function Clear Dying GASP Statistics.

#### 9.14.4.33 ULONG SLQSSwiGetCrashAction ( BYTE * *pDevCrashState* )

This API queries the Crash State from the device.

**Parameters**

| | |
|---|---|
| *pDevCrash-State[OUT]* | • 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 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: NA
Please free two buffers after get crash report successfully

1. pCrashInfoParams->pCrashInfo->pCrashString
2. pCrashInfoParams->pCrashInfo->pGCDumpString Timeout: 5 Secs

#### 9.14.4.34 ULONG SLQSSwiGetCrashInfo ( BYTE * *pClear,* CrashInfoParams * *pCrashInfoParams* )

This API queries the Crash Information from the device.

**Parameters**

| | |
|---|---|
| *pClear[IN]* | • request parameter Clear(Optional parameter)<br>• Values: 0 - Do not clear crash data after response 1 - Clear crash data after response |
| *pCrashInfo-Params[Out]* | • Pointer to structure CrashInfoParams<br>• See CrashInfoParams 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: NA
Timeout: 5 Secs

#### 9.14.4.35 ULONG SLQSSwiGetDyingGaspCfg ( getDyingGaspCfg * *pConfig* )

This function queries Dying GASP Config.

**Parameters**

| | |
|---|---|
| *pGetCustom-FeatureV2* | • See getDyingGaspCfg for more information of the inpurt structure * |

#### 9.14.4.36 ULONG SLQSSwiGetDyingGaspStatistics ( getDyingGaspStatistics * *pStatistics* )

This function queries Dying GASP Statistics.

**Parameters**

| | |
|---|---|
| *pStatistics* | • See getDyingGaspStatistics for more information of the inpurt structure * |

#### 9.14.4.37 ULONG SLQSSwiGetFSN ( FactorySequenceNumber * *pFSNumber* )

This API get the Factory Sequence Number of the device.

**Parameters**

| | |
|---|---|
| *pFSNumber* | • Pointer to structure FactorySequenceNumber<br>• See FactorySequenceNumber 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
Timeout: 5 Secs

#### 9.14.4.38 ULONG SLQSSwiGetFwUpdateStatus ( FirmwareUpdatStat * *pFirmwareUpdatStat* )

This API will be used to query last firmware update status. The firmware status is stored in RAM and can be retained over warm resets but not power off resets.

**Parameters**

| | |
|---|---|
| *pFirmware-UpdatStat* | • Pointer to structure FirmwareUpdatStat<br>• See FirmwareUpdatStat 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: NA
Device Supported: MC73xx
Timeout: 5 Secs

#### 9.14.4.39 ULONG SLQSSwiGetHostDevInfo ( SLQSSwiGetHostDevInfoParams * *pGetHostDevInfoParams* )

This API Get Host Information from the device.

**Parameters**

| | |
|---|---|
| *pGetHostDev-InfoParams* | • See SLQSSwiGetHostDevInfoParams 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: NA
Timeout: 2 Secs

#### 9.14.4.40 ULONG SLQSSwiGetOSInfo ( SLQSSwiGetOSInfoParams * *pParams* )

This API queries the device operating system info configured on the modem for OMA-DM reporting

**Parameters**

| | |
|---|---|
| *pParams* | • - See SLQSSwiGetOSInfoParams 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: NA
Timeout: 2 Secs

#### 9.14.4.41 ULONG SLQSSwiGetSerialNoExt ( SLQSSwiGetSerialNoExtParams * *pParams* )

This API is used to get the MEID of the modem. For CDMA devices that use a RUIM, the MEID of the modem will always be returned.

**Parameters**

| | |
|---|---|
| *SLQSSwiGet-SerialNoExt-Params* | |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: NA
Timeout: 5 Secs

#### 9.14.4.42 ULONG SLQSSwiGetUSBComp ( USBCompParams * *pUSBCompParams* )

This API queries the modem's USB interface configuration and supported configuration parameters.

**Parameters**

| | |
|---|---|
| *pUSBComp-Params* | • Pointer to structure USBCompParams<br>• See USBCompParams 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: NA
Timeout: 5 Secs

#### 9.14.4.43 ULONG SLQSSwiSetCrashAction ( BYTE *crashActionParams* )

This API set the Crash Action to the device.

**Parameters**

| | |
|---|---|
| *crashAction-Params[IN]* | • Crash Action<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 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: NA
Timeout: 5 Secs

#### 9.14.4.44 ULONG SLQSSwiSetDyingGaspCfg ( setDyingGaspCfg * *pConfig* )

This function set Dying GASP Config.

**Parameters**

| | |
|---|---|
| *pConfig* | • See setDyingGaspCfg for more information of the inpurt structure |

#### 9.14.4.45 ULONG SLQSSwiSetHostDevInfo ( SLQSSwiSetHostDevInfoParams * *pSetHostDevInfoParams* )

This API Sets the host device info configured on the modem for OMA-DM reporting

**Parameters**

| | |
|---|---|
| *pSetHostDev-InfoParams* | • See SLQSSwiSetHostDevInfoParams 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: NA
Timeout: 2 Secs

#### 9.14.4.46 ULONG SLQSSwiSetOSInfo ( SLQSSwiSetOSInfoParams * *pParams* )

This API Set OS Information to the device.

**Parameters**

| | |
|---|---|
| *pParams* | • See SLQSSwiSetOSInfoParams 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: NA
Timeout: 2 Secs

#### 9.14.4.47 ULONG SLQSSwiSetUSBComp ( USBCompConfig * *pUSBCompConfig* )

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**

| | |
|---|---|
| *pUSBComp-Config* | • Pointer to structure USBCompConfig<br>• See USBCompConfig 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: NA
Timeout: 5 Secs

#### 9.14.4.48 ULONG SLQSUIMGetState ( ULONG * *pUIMState* )

Returns the UIM state. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMGetCardStatus() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pUIMState[OUT]* | • UIM state:<br>– 0x00 - UIM initialization completed<br>– 0x01 - UIM locked or failed<br>– 0x02 - UIM not present<br>– 0x03 - 0xFE - Reserved<br>– 0xFF - UIM state currently unavailable |

**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

#### 9.14.4.49 ULONG UIMChangePIN ( ULONG *id,* CHAR * *pOldValue,* CHAR * *pNewValue,* ULONG * *pVerifyRetriesLeft,* ULONG * *pUnblockRetriesLeft* )

Changes the PIN value for a given PIN. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMChangePin() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *id[IN]* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *pOldValue[IN]* | • Old PIN value of PIN to change |
| *pNewValue[IN]* | • New PIN value of PIN to change |
| *pVerifyRetries-Left[OUT]* | • Optional Parameter<br>• Upon operational failure, this will indicate number of retries left, after which PIN will be blocked.<br>– 0xFFFFFFFF - Unknown |
| *pUnblock-RetriesLeft[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>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.50 ULONG UIMGetControlKeyStatus ( ULONG *id,* ULONG * *pStatus,* ULONG * *pVerifyRetriesLeft,* ULONG * *pUnblockRetriesLeft* )

Returns the status of the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Pleaes use SLQSUIMGetConfiguration instead for new firmware versions and new modules.

**Parameters**

| | |
|---|---|
| *id[IN]* | • Facility ID<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) |
| *pStatus[OUT]* | • Control key status<br>– 0 - Deactivated<br>– 1 - Activated<br>– 2 - Blocked |
| *pVerifyRetries-Left[OUT]* | • The number of retries left, after which the control key will be blocked<br>– 0xFFFFFFFF - Unknown |
| *pUnblock-RetriesLeft[OUT]* | • The number of unblock retries left, after which the control key will be permanently blocked<br>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.51 ULONG UIMGetICCID ( BYTE *stringSize*, CHAR * *pString* )

Returns the UIM ICCID. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use SLQSUIMReadTransparent()(EF ID: 3F00 2FE2) instead for new firmware versions and new modules.

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain. |
| *pString[OUT]* | • NULL terminated string |

**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

#### 9.14.4.52 ULONG UIMGetPINStatus ( ULONG *id*, ULONG * *pStatus*, ULONG * *pVerifyRetriesLeft*, ULONG * *pUnblockRetriesLeft* )

Gets the status of the SIM PINs. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMGetCardStatus() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *id* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *pStatus[OUT]* | • PIN 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 |
| *pVerifyRetriesLeft[OUT]* | • Upon operational failure, this will indicate number of retries left, after which PIN will be blocked.<br>– 0xFFFFFFFF - Unknown |
| *pUnblockRetriesLeft[OUT]* | • 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>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.53 ULONG UIMSetControlKeyProtection ( ULONG *id*, ULONG *status*, CHAR * *pValue*, ULONG * *pVerifyRetriesLeft* )

Changes the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMDepersonalization() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *id[IN]* | • Facility ID<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) |
| *status[IN]* | • Control key status<br>– 0 - Deactivated<br>– 1 - Activated<br>– 2 - Blocked |
| *pValue[IN]* | • Control key de-personalization string (maximum length of 8 characters) |
| *pVerifyRetries-Left[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of retries left, after which the control key will be blocked<br>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.54 ULONG UIMSetPINProtection ( ULONG *id*, ULONG *bEnable*, CHAR * *pValue*, ULONG * *pVerifyRetriesLeft*, ULONG * *pUnblockRetriesLeft* )

Enables or disables protection of SIM contents for a given PIN, This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMSetPinProtection() for new firmware versions and new modules

**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 |
| *pVerifyRetries-Left[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of retries left, after which PIN will be blocked.<br>– 0xFFFFFFFF - Unknown |
| *pUnblock-RetriesLeft[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>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.55 ULONG UIMUnblockControlKey ( ULONG *id*, CHAR * *pValue*, ULONG * *pUnblockRetriesLeft* )

Unblocks the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMDepersonalization() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *id[IN]* | • Facility ID<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) |

| | |
|---|---|
| *pValue[IN]* | • Control key de-personalization string (maximum length of 8 characters) |
| *pUnblock-RetriesLeft[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of unblock retries left, after which the control key will be blocked<br>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.56 ULONG UIMUnblockPIN ( ULONG *id*, CHAR * *pPUKValue*, CHAR * *pNewValue*, ULONG * *pVerifyRetriesLeft*, ULONG * *pUnblockRetriesLeft* )

Unblocks a blocked SIM. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMUnblockPin() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *id[IN]* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *pPUKValue[IN]* | • PUK value of PIN to unblock |
| *pNewValue[IN]* | • New PIN value of PIN to unblock |
| *pVerifyRetries-Left[OUT]* | • Optional Parameter<br>• Upon operational failure, this will indicate number of retries left, after which the PIN will be blocked.<br>– 0xFFFFFFFF - Unknown |
| *pUnblock-RetriesLeft[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>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.57 ULONG UIMVerifyPIN ( ULONG *id*, CHAR * *pValue*, ULONG * *pVerifyRetriesLeft*, ULONG * *pUnblockRetriesLeft* )

Verifies a SIM PIN. This API is deprecated on MC73xx/EM73xx/WP76xx modules since firmware version SWI9X15C_05_xx_xx_xx, all EM74xx and WP76xx firmware versions. Please use API SLQSUIMVerifyPin()() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *id[IN]* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *pValue[IN]* | • Value of PIN to verify |
| *pVerifyRetriesLeft[OUT]* | • Optional Parameter<br>• Upon operational failure, this will indicate number of retries left, after which the PIN will be blocked.<br>– 0xFFFFFFFF - Unknown |
| *pUnblockRetriesLeft[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>– 0xFFFFFFFF - Unknown |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.14.4.58 ULONG ValidateSPC ( CHAR * *pSPC* )

This function Validates Service Programming code of the device

**Parameters**

| | |
|---|---|
| *pSPC[IN]* | • NULL-terminated string representing a six-digit service programming code |

**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
Device Supported: MC83x5, MC7750
Timeout: 2 seconds

## 9.15 qaGobiApiFms.h File Reference

Firmware Management Service API function prototypes.

### Data Structures

- struct fwinfo_s
- struct slqsfwinfo_s
- struct qmifwinfo_s
- struct ImageElement
- struct PrefImageList
- struct ImageIdElement
- struct ImageIDEntries
- struct ImageList
- struct sGetDeviceSeriesResult
- struct SWI_STRUCT_CarrierImage
- struct CurrImageInfo
- struct CurrentImgList

### Macros

- #define SLQSFWINFO_MODELID_SZ 20
- #define SLQSFWINFO_BOOTVERSION_SZ 85
- #define SLQSFWINFO_APPVERSION_SZ 85
- #define SLQSFWINFO_SKU_SZ 15
- #define SLQSFWINFO_PACKAGEID_SZ 85
- #define SLQSFWINFO_CARRIER_SZ 20
- #define SLQSFWINFO_PRIVERSION_SZ 16
- #define SLQSFWINFO_CUR_CARR_NAME 17

- #define SLQSFWINFO_CUR_CARR_REV 13
- #define GOBI_MBN_IMG_ID_STR_LEN 16
- #define GOBI_MBN_BUILD_ID_STR_LEN 100
- #define GOBI_LISTENTRIES_MAX 2
- #define GOBI_SET_IMG_PREF_RSPLEN 40
- #define DEVICE_SHUTDOWN 5
- #define DEVICE_RESET 4
- #define DEVICE_OFFLINE 3
- #define FIRMWARE_UPDATE_SUCCESS 0x01
- #define FIRMWARE_UPDATE_FAIL 0x01
- #define PRI_UPDATE_FAIL 0x02
- #define FIRMWARE_UPGRADE_SUCCESS 0x00
- #define IMG_ID_LEN 16
- #define BUILD_ID_LEN 100
- #define G3K_FIRMWARE_DOWNLOAD 1
- #define SPKG_FIRMWARE_DOWNLOAD 2
- #define UNIQUE_ID_LEN 16
- #define BUILD_ID_MAX_LEN 255
- #define IMGDETAILS_LEN 16
- #define MAX_IMAGE_IDE_ELEMENTS 50

## Enumerations

- enum eGobiImageTech {
  eGOBI_IMG_TECH_CDMA = 0,
  eGOBI_IMG_TECH_UMTS }
- enum eGobiImageCarrier {

eGOBI_IMG_CAR_GENERIC = 1,
eGOBI_IMG_CAR_FACTORY,
eGOBI_IMG_CAR_NORF,
eGOBI_IMG_CAR_VERIZON = 101,
eGOBI_IMG_CAR_SPRINT,
eGOBI_IMG_CAR_ALLTEL,
eGOBI_IMG_CAR_BELL,
eGOBI_IMG_CAR_TELUS,
eGOBI_IMG_CAR_US,
eGOBI_IMG_CAR_TELSTRA1,
eGOBI_IMG_CAR_CHINA_UNICOM,
eGOBI_IMG_CAR_TELCOM_NZ,
eGOBI_IMG_CAR_SK_TELCOM1,
eGOBI_IMG_CAR_RELIANCE1,
eGOBI_IMG_CAR_TATA,
eGOBI_IMG_CAR_METROPCS,
eGOBI_IMG_CAR_LEAP,
eGOBI_IMG_CAR_KDDI,
eGOBI_IMG_CAR_IUSACELL,
eGOBI_IMG_CAR_CHINA_TELECOM,
eGOBI_IMG_CAR_OMH,
eGOBI_IMG_CAR_GENERIC_CDMA,
eGOBI_IMG_CAR_ATT = 201,
eGOBI_IMG_CAR_VODAFONE,
eGOBI_IMG_CAR_TMOBILE,
eGOBI_IMG_CAR_ORANGE,
eGOBI_IMG_CAR_TELEFONICA,
eGOBI_IMG_CAR_TELCOM_ITALIA,
eGOBI_IMG_CAR_3,
eGOBI_IMG_CAR_O2,
eGOBI_IMG_CAR_SFR,
eGOBI_IMG_CAR_SWISSCOM,
eGOBI_IMG_CAR_CHINA_MOBILE,
eGOBI_IMG_CAR_TELSTRA2,
eGOBI_IMG_CAR_SINGTEL_OPTUS,
eGOBI_IMG_CAR_RELIANCE2,
eGOBI_IMG_CAR_BHARTI,
eGOBI_IMG_CAR_NTT_DOCOMO,
eGOBI_IMG_CAR_EMOBILE,
eGOBI_IMG_CAR_SOFTBANK,
eGOBI_IMG_CAR_KT_FREETEL,
eGOBI_IMG_CAR_SK_TELCOM2,
eGOBI_IMG_CAR_TELENOR,
eGOBI_IMG_CAR_NETCOM,
eGOBI_IMG_CAR_TELIASONERA,
eGOBI_IMG_CAR_AMX_TELCEL,
eGOBI_IMG_CAR_BRASIL_VIVO,
eGOBI_IMG_CAR_AERIS,
eGOBI_IMG_CAR_ROGERS }

- enum eGobiImageRegion {
  eGOBI_IMG_REG_NA = 0,
  eGOBI_IMG_REG_LA,
  eGOBI_IMG_REG_EU,
  eGOBI_IMG_REG_ASIA,
  eGOBI_IMG_REG_AUS,
  eGOBI_IMG_REG_GLOBAL }

- enum eGobiImageGPS {

eGOBI_IMG_GPS_NONE = 0,
eGOBI_IMG_GPS_STAND_ALONE,
eGOBI_IMG_GPS_ASSISTED,
eGOBI_IMG_GPS_NO_XTRA }

- enum eGobiDeviceSeries {
eGOBI_DEV_SERIES_UNKNOWN = -1,
eGOBI_DEV_SERIES_NON_GOBI = 0,
eGOBI_DEV_SERIES_G3K,
eGOBI_DEV_SERIES_SIERRA_GOBI,
eGOBI_DEV_SERIES_9X15,
eGOBI_DEV_SERIES_9X30,
eGOBI_DEV_SERIES_9X07_9X50,
eGobi_DEV_SERIES_MC83 }

## Functions

- ULONG GetImageStore (WORD imageStorePathSize, CHAR *pImageStorePath)
- ULONG SLQSGetFirmwareInfo (struct qmifwinfo_s *pinfo)
- ULONG SLQSGetImageInfoMC77xx (LPCSTR path, struct qmifwinfo_s *pinfo)
- ULONG SLQSGetImageInfoMC83xx (LPCSTR path, struct qmifwinfo_s *pinfo)
- ULONG SLQSGetImageInfo (LPCSTR path, struct qmifwinfo_s *pinfo)
- ULONG UpgradeFirmware2k (CHAR *pDestinationPath)
- ULONG GetImagesPreference (ULONG *pImageListSize, struct PrefImageList *pImageList)
- ULONG SetImagesPreference (ULONG imageListSize, BYTE *pImageList, ULONG bForceDownload, BYTE modemIndex, ULONG *pImageTypesSize, BYTE *pImageTypes)
- ULONG GetStoredImages (ULONG *pImageListSize, BYTE *pImageList)
- ULONG DeleteStoredImage (ULONG imageInfoSize, BYTE *pImageInfo)
- ULONG SLQSGetImageInfo_9x15 (LPCSTR path, BYTE imgType, struct slqsfwinfo_s *pinfo)
- ULONG SLQSUpgradeFirmware9x07 (CHAR *pDestinationPath)
- ULONG SLQSUpgradeFirmware9x15 (CHAR *pDestinationPath)
- ULONG SLQSGetBootVersionNumber (ULONG *bootversion)
- BOOL SLQSIsSpkgFormatRequired (void)
- void SLQSSetSpkgFormatRequired (BYTE isneeded)
- ULONG upgrade_mc77xx_fw (LPCSTR path)
- void eGetDeviceSeries (struct sGetDeviceSeriesResult *)
- ULONG SLQSSwiGetAllCarrierImages (ULONG *pNumOfItems, struct SWI_STRUCT_CarrierImage *pCarrierImages, char *pFolderPath)
- ULONG DownloadToSlot (CHAR *path, struct slqsfwinfo_s fwImgInfo, struct slqsfwinfo_s priImgInfo, BYTE slot, BYTE forceDownload)
- ULONG SLQSDownloadFirmwareToSlot (CHAR *pPath, BYTE slot_index, BYTE force_download)
- ULONG SLQSGetValidFwPriCombinations (struct ImageList *pStoredImageList, ULONG *pValidCombinationSize, struct SWI_STRUCT_CarrierImage *pValidCombinations)
- ULONG SLQSSetSIMBasedImageSwitching (void)
- void SLQSSetCrashStateCheckIgnore (BOOL ignore)
- ULONG SLQSSwiGetFirmwareCurr (CurrentImgList *pCurrentImgList)
- ULONG upgrade_9x07_9x50_fw (LPCSTR path)
- ULONG SetImagePath (CHAR *path, ULONG isMDM9x15, WORD imgMask, BYTE mdmProtocol)

### 9.15.1 Detailed Description

Firmware Management Service API function prototypes.

### 9.15.2 Macro Definition Documentation

#### 9.15.2.1 #define BUILD_ID_LEN 100

#### 9.15.2.2 #define BUILD_ID_MAX_LEN 255

#### 9.15.2.3 #define DEVICE_OFFLINE 3

#### 9.15.2.4 #define DEVICE_RESET 4

#### 9.15.2.5 #define DEVICE_SHUTDOWN 5

#### 9.15.2.6 #define FIRMWARE_UPDATE_FAIL 0x01

#### 9.15.2.7 #define FIRMWARE_UPDATE_SUCCESS 0x01

#### 9.15.2.8 #define FIRMWARE_UPGRADE_SUCCESS 0x00

#### 9.15.2.9 #define G3K_FIRMWARE_DOWNLOAD 1

#### 9.15.2.10 #define GOBI_LISTENTRIES_MAX 2

#### 9.15.2.11 #define GOBI_MBN_BUILD_ID_STR_LEN 100

#### 9.15.2.12 #define GOBI_MBN_IMG_ID_STR_LEN 16

#### 9.15.2.13 #define GOBI_SET_IMG_PREF_RSPLEN 40

#### 9.15.2.14 #define IMG_ID_LEN 16

#### 9.15.2.15 #define IMGDETAILS_LEN 16

#### 9.15.2.16 #define MAX_IMAGE_IDE_ELEMENTS 50

#### 9.15.2.17 #define PRI_UPDATE_FAIL 0x02

#### 9.15.2.18 #define SLQSFWINFO_APPVERSION_SZ 85

#### 9.15.2.19 #define SLQSFWINFO_BOOTVERSION_SZ 85

#### 9.15.2.20 #define SLQSFWINFO_CARRIER_SZ 20

#### 9.15.2.21 #define SLQSFWINFO_CUR_CARR_NAME 17

#### 9.15.2.22 #define SLQSFWINFO_CUR_CARR_REV 13

#### 9.15.2.23 #define SLQSFWINFO_MODELID_SZ 20

#### 9.15.2.24 #define SLQSFWINFO_PACKAGEID_SZ 85

#### 9.15.2.25 #define SLQSFWINFO_PRIVERSION_SZ 16

#### 9.15.2.26 #define SLQSFWINFO_SKU_SZ 15

#### 9.15.2.27 #define SPKG_FIRMWARE_DOWNLOAD 2

#### 9.15.2.28 #define UNIQUE_ID_LEN 16

### 9.15.3 Enumeration Type Documentation

#### 9.15.3.1 enum eGobiDeviceSeries

enumeration which lists the Device Series

**Enumerator**

***eGOBI_DEV_SERIES_UNKNOWN***
***eGOBI_DEV_SERIES_NON_GOBI***
***eGOBI_DEV_SERIES_G3K***
***eGOBI_DEV_SERIES_SIERRA_GOBI***
***eGOBI_DEV_SERIES_9X15***
***eGOBI_DEV_SERIES_9X30***
***eGOBI_DEV_SERIES_9X07_9X50***
***eGobi_DEV_SERIES_MC83***

#### 9.15.3.2 enum eGobiImageCarrier

enumeration which lists the carrier supported by the image

**Enumerator**

***eGOBI_IMG_CAR_GENERIC***
***eGOBI_IMG_CAR_FACTORY***
***eGOBI_IMG_CAR_NORF***
***eGOBI_IMG_CAR_VERIZON***
***eGOBI_IMG_CAR_SPRINT***
***eGOBI_IMG_CAR_ALLTEL***
***eGOBI_IMG_CAR_BELL***
***eGOBI_IMG_CAR_TELUS***
***eGOBI_IMG_CAR_US***
***eGOBI_IMG_CAR_TELSTRA1***
***eGOBI_IMG_CAR_CHINA_UNICOM***
***eGOBI_IMG_CAR_TELCOM_NZ***
***eGOBI_IMG_CAR_SK_TELCOM1***
***eGOBI_IMG_CAR_RELIANCE1***
***eGOBI_IMG_CAR_TATA***
***eGOBI_IMG_CAR_METROPCS***
***eGOBI_IMG_CAR_LEAP***
***eGOBI_IMG_CAR_KDDI***
***eGOBI_IMG_CAR_IUSACELL***
***eGOBI_IMG_CAR_CHINA_TELECOM***
***eGOBI_IMG_CAR_OMH***
***eGOBI_IMG_CAR_GENERIC_CDMA***
***eGOBI_IMG_CAR_ATT***
***eGOBI_IMG_CAR_VODAFONE***

***eGOBI_IMG_CAR_TMOBILE***
***eGOBI_IMG_CAR_ORANGE***
***eGOBI_IMG_CAR_TELEFONICA***
***eGOBI_IMG_CAR_TELCOM_ITALIA***
***eGOBI_IMG_CAR_3***
***eGOBI_IMG_CAR_O2***
***eGOBI_IMG_CAR_SFR***
***eGOBI_IMG_CAR_SWISSCOM***
***eGOBI_IMG_CAR_CHINA_MOBILE***
***eGOBI_IMG_CAR_TELSTRA2***
***eGOBI_IMG_CAR_SINGTEL_OPTUS***
***eGOBI_IMG_CAR_RELIANCE2***
***eGOBI_IMG_CAR_BHARTI***
***eGOBI_IMG_CAR_NTT_DOCOMO***
***eGOBI_IMG_CAR_EMOBILE***
***eGOBI_IMG_CAR_SOFTBANK***
***eGOBI_IMG_CAR_KT_FREETEL***
***eGOBI_IMG_CAR_SK_TELCOM2***
***eGOBI_IMG_CAR_TELENOR***
***eGOBI_IMG_CAR_NETCOM***
***eGOBI_IMG_CAR_TELIASONERA***
***eGOBI_IMG_CAR_AMX_TELCEL***
***eGOBI_IMG_CAR_BRASIL_VIVO***
***eGOBI_IMG_CAR_AERIS***
***eGOBI_IMG_CAR_ROGERS***

#### 9.15.3.3 enum eGobiImageGPS

enumeration which lists the GPS type supported by the image

**Enumerator**

***eGOBI_IMG_GPS_NONE***
***eGOBI_IMG_GPS_STAND_ALONE***
***eGOBI_IMG_GPS_ASSISTED***
***eGOBI_IMG_GPS_NO_XTRA***

#### 9.15.3.4 enum eGobiImageRegion

enumeration which lists the region supported by the image

**Enumerator**

***eGOBI_IMG_REG_NA***
***eGOBI_IMG_REG_LA***
***eGOBI_IMG_REG_EU***
***eGOBI_IMG_REG_ASIA***
***eGOBI_IMG_REG_AUS***
***eGOBI_IMG_REG_GLOBAL***

#### 9.15.3.5 enum eGobiImageTech

enumeration which lists the technology supported by the image

**Enumerator**

***eGOBI_IMG_TECH_CDMA***

***eGOBI_IMG_TECH_UMTS***

### 9.15.4 Function Documentation

#### 9.15.4.1 ULONG DeleteStoredImage ( ULONG *imageInfoSize*, BYTE * *pImageInfo* )

Used to delete the specified image from the device. This API function is only relevant to devices with the ability to store multiple firmware images(see Device Supported section below).

**Parameters**

| | |
|---|---|
| *imageInfoSize[IN]* | • The size in BYTEs of the image info array |
| *pImageInfo[IN]* | • The image info list array containing information about the image to be deleted.<br>• See ImageElement |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC83x5/SL9090
Timeout: 2 Secs

#### 9.15.4.2 ULONG DownloadToSlot ( CHAR * *path*, struct slqsfwinfo_s *fwImgInfo*, struct slqsfwinfo_s *priImgInfo*, BYTE *slot*, BYTE *forceDownload* )

This API is dedicated for downloading a firmware image into a specific slot. it is used for EM74xx/MC74xx only

**Parameters**

| | |
|---|---|
| *path* | • fully qualified path to firmware image to download |
| *fwImgInfo* | • See struct slqsfwinfo_s |
| *priImgInfo* | • See struct slqsfwinfo_s |

| | |
|---|---|
| *slot* | • slot id in the modem to store the firmware |
| *forceDownload* | • a flag to force download take place. |

**Returns**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

only EM74xx/MC74xx is supported

#### 9.15.4.3 void eGetDeviceSeries ( struct sGetDeviceSeriesResult * )

Name : eGetDeviceSeries

**Parameters**

| | |
|---|---|
| *none* | |

**Returns**

sGetDeviceSeriesResult

**Note**

Get Devie Series

#### 9.15.4.4 ULONG GetImagesPreference ( ULONG * *pImageListSize,* struct PrefImageList * *pImageList* )

restore original alignment from stack Gets the current images preference from the device.

**Parameters**

| | |
|---|---|
| *pImageListSize[-IN/OUT]* | • Upon input, the size of structure ImageList ImageList<br>• Upon successful output, the number of BYTEs copied to the image list array |
| *pImageList[OUT]* | • The caller must supply a pointer to a ImageList structure typecast as a BYTE pointer |

**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

#### 9.15.4.5 ULONG GetImageStore ( WORD *imageStorePathSize,* CHAR * *pImageStorePath* )

Returns the image store folder, i.e., the folder containing one or more carrier-specific image subfolders compatible with the currently connected QC WWAN device.

**Parameters**

| | |
|---|---|
| *imageStorePathSize* | • Maximum number of characters (including NULL terminator) that can be copied to the image store path array. |
| *pImageStorePath[OUT]* | • The path to the image store |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.15.4.6 ULONG GetStoredImages ( ULONG * *pImageListSize,* BYTE * *pImageList* )

restore original alignment from stack Gets the list of images stored on the device.

**Parameters**

| | |
|---|---|
| *pImageListSize[IN/OUT]* | • Upon input, the size of structure ImageList ImageList<br>• Upon successful output, the number of BYTEs copied to the image list array |
| *pImageList[OUT]* | • The caller must supply a pointer to a ImageList structure typecast as a BYTE pointer |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC83x5/SL9090
Timeout: 2 seconds

#### 9.15.4.7 ULONG SetImagePath ( CHAR * *path*, ULONG *isMDM9x15*, WORD *imgMask*, BYTE *mdmProtocol* )

This API is used to set image path for firmware download

**Parameters**

| | |
|---|---|
| *path[IN]* | • fully qualified path to firmware image to download. |
| *isMDM9x15[IN]* | • a flag indicates whether it is a MDM9x15 device |
| *imgMask[IN]* | • image mask to check CWE and NVU files are required or not |
| *mdmProtocol[IN]* | • which firmware download protocol should be used<br>– 0 - SSDP protocol<br>– 1 - Firehose protocol |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 12 seconds

#### 9.15.4.8 ULONG SetImagesPreference ( ULONG *imageListSize*, BYTE * *pImageList*, ULONG *bForceDownload*, BYTE *modemIndex*, ULONG * *pImageTypesSize*, BYTE * *pImageTypes* )

Sets the current images preference on the device. After this function successfully completes, the device must be reset for the selected image preference to be realized. Additionally, when the returned list of image types that require downloading is not empty, the device opens in QDL mode after the reset. At that point, the QDL portion of this API must be used to download the selected image preference to the device.

**Parameters**

| | |
|---|---|
| *imageListSize* | • The size in BYTEs of the image list array |
| *pImageList[IN]* | • The image info list array containing Image Elements<br>– See PrefImageList |
| *bForce-Download[IN]* | • Force device to download images from host? 0 - No Nonzero - Yes |
| *modemIndex* | • Desired storage index for downloaded modem image (optional, a value of 0xFF indicates unspecified) |

| | |
|---|---|
| *pImageTypes-Size[IN/OUT]* | • Upon input, maximum number of elements that download image types array can contain<br>• Upon successful output, number of elements in download image types array |
| *pImageTypes[OUT]* | -The download image types array. |

**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

#### 9.15.4.9 ULONG SLQSDownloadFirmwareToSlot ( CHAR * *pPath*, BYTE *slot_index*, BYTE *force_download* )

This API is used to download firmware to a specific slot id of the modem. It is only applicable for EM74xx variant. This API encapsulates all steps involved in the firmware download process.Hence it is a blocking API call.

This API will not return until the entire process has been completed. This API will takes significant amount of time (in order of minutes, normally should be less than 10 minutes).

This API Performs the following steps:

1. Verifies arguments.
2. Retrieve and store the details of the firmware and the PRI file
3. Enable device state change callback.
4. Enable firmware download callback.
5. Set Image preference on the device and reset the device.
6. Wait for the firmware to download and device to become ready.
7. Check the firmware update status. If fail, return an error.
8. If update status is OK, check if current image preference and preferred image preference( from step 2 ) match
9. If match, firmware download is successful. otherwise, report FW_PREFERENCE_MISMATCH

   10.Disable callbacks and exit.

   The call to this API blocks until step 7 or 10. This could be a significant amount of time ( in order of minutes ). Also note that the device state change callback and firmware download callback are used internally

   within this API. Hence the user application's instance of these callbacks (if any) are cleared. The user must re-enable these callbacks after a call to this API in order to use them.

**Parameters**

| | |
|---|---|
| *pPath[IN]* | • fully qualified path to firmware image to download. |
| *slot_index[IN]* | • slot id in the modem to store the firmware |
| |  |
| *force_download[IN]* | • a flag to force download take place. |

**Returns**

- eQCWWAN_ERR_NONE - Firmware download/Switch success.
- eQCWWAN_ERR_INVALID_ARG - The path input does not contain any image
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL - Firmware download/switch failed
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH - Download success but device offline due to image preference mismatch ( ref. syslogs for cause )

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: NA

#### 9.15.4.10 ULONG SLQSGetBootVersionNumber ( ULONG * *bootversion* )

Gets the boot loader version number

**Parameters**

| | |
|---|---|
| *bootversion[OUT]* | • boot loader version presented by a 4 byte integer |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC9090/SL9090
Timeout: 2 seconds

#### 9.15.4.11 ULONG SLQSGetFirmwareInfo ( struct qmifwinfo_s * *pinfo* )

Returns firmware image information from the connected device

**Parameters**

| | |
|---|---|
| *pinfo[OUT]* | • firmware image information record |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

**Note**

Timeout: 2 Seconds.

#### 9.15.4.12 ULONG SLQSGetImageInfo ( LPCSTR *path,* struct qmifwinfo_s * *pinfo* )

Returns firmware image information from a CWE file or mbn files stored on the host. For CWE, information is returned for the first CWE image found at the specified path. For MBN, the provided path must be located under the image store for the currently connected QC WWAN device. Note that as this API supports multiple firmware image types, it relies on the presence of a supported device. Otherwise, refer to SLQSGetImageInfoMC83xx and SLQSGetImageInfoMC77xx for APIs which do not rely on the presence of a supported device.

**Parameters**

| | |
|---|---|
| *path[IN]* | • fully qualified path to folder containing CWE image or MBN images<br>• should use a "/" at the end of the path. |
| *pinfo[OUT]* | • firmware image information record |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

**Note**

Timeout: N/A

#### 9.15.4.13 ULONG SLQSGetImageInfo_9x15 ( LPCSTR *path,* BYTE *imgType,* struct slqsfwinfo_s * *pinfo* )

Returns firmware image information from a CWE file(s) stored on the host. It does not rely on the presence of a supported device.

**Parameters**

| | |
|---|---|
| *path[IN]* | • fully qualified path to folder containing the image(s)<br>• should use a "/" at the end of the path. |
| *imgType[IN]* | • 2 - Firmware Image( .cwe extension )<br>• 3 - PRI Image ( .nvu extension ) |

| | |
|---|---|
| *pinfo[OUT]* | • firmware image information record |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

**Note**

Device Supported: MC73xx
Timeout: N/A

#### 9.15.4.14 ULONG SLQSGetImageInfoMC77xx ( LPCSTR *path,* struct qmifwinfo_s * *pinfo* )

Returns firmware image information from a SPKGS CWE file stored on the host. The information is returned for the first SPKGS CWE image found at the specified path. This API executes independent of a MC77xx being connected to the target.

**Parameters**

| | |
|---|---|
| *path[IN]* | • fully qualified path to folder containing SPKG CWE image<br>• should use a "/" at the end of the path. |
| *pinfo[OUT]* | • firmware image information record |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

**Note**

Device Supported: MC77xx
Timeout: N/A

#### 9.15.4.15 ULONG SLQSGetImageInfoMC83xx ( LPCSTR *path,* struct qmifwinfo_s * *pinfo* )

Returns firmware image information from an MBN file located on the host. This API executes independent of a MC83xx being connected to the target.

**Parameters**

| | |
|---|---|
| *path[IN]* | • fully qualified path to folder containing MBN file<br>• should use a "/" at the end of the path. |
| *pinfo[OUT]* | • firmware image information record |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values
struct qmifwinfo_s

**Note**

Device Supported: MC83xx/SL9090
Timeout: N/A

#### 9.15.4.16 ULONG SLQSGetValidFwPriCombinations ( struct ImageList * *pStoredImageList*, ULONG * *pValidCombinationSize*, struct SWI_STRUCT_CarrierImage * *pValidCombinations* )

This API distills valid Firmware/PRI combinations from GetStoredImages result

**Parameters**

| | | |
|---|---|---|
| in | *pStoredImage-List* | • image list returned from GetStoredImages |
| in,out | *pValid-CombinationSize* | • number of combination passed in and returned |
| out | *pValid-Combinations* | • valid combinations returned |

**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

#### 9.15.4.17 BOOL SLQSIsSpkgFormatRequired ( void )

Check if SPKG format download is required for SL9090/MC9090, it returns the value whcih was set by API SLQS-SetSpkgFormatRequired()

**Parameters**

| | |
|---|---|
| *none* | |

**Returns**

return TRUE if required, otherwise, return FALSE

**Note**

Device Supported: MC9090/SL9090

#### 9.15.4.18 void SLQSSetCrashStateCheckIgnore ( BOOL *ignore* )

This API is used to set whether ignore crash state checking before proceed firmware download using the API UpgradeFirmware2k().

**Parameters**

| | | |
|---|---|---|
| in | *ignore* | • 0 - crash state checking applied (default value)<br>• 1 - ignore crash state checking |

**Note**

Device Supported: EM73xx/MC73xx, EM74xx/MC74xx

#### 9.15.4.19 ULONG SLQSSetSIMBasedImageSwitching ( void )

This API is used to enable the SIM-based Image Switching. The modem will reboot automatically to take effect of the enabing

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: EM74xx/MC74xx

#### 9.15.4.20 void SLQSSetSpkgFormatRequired ( BYTE *isneeded* )

Set if SPKG format download is required for SL9090/MC9090

**Parameters**

| | |
|---|---|
| *isneeded* | [INPUT] user inputs the firmware download method preference<br>• 1 - Gobi3K download method, use mbn files. This is default value<br>• 2 - SPKG download method, use cwe file |

**Returns**

None

**Note**

Device Supported: MC9090/SL9090

#### 9.15.4.21 ULONG SLQSSwiGetAllCarrierImages ( ULONG * *pNumOfItems,* struct SWI_STRUCT_CarrierImage * *pCarrierImages,* char * *pFolderPath* )

This API gets a list of all images stored on both the host and the device

**Parameters**

| | |
|---|---|
| *pNumOfItems* | • Number of Images{IN/OUT] |
| *pCarrierImages[-OUT]* | • See SWI_STRUCT_CarrierImage |
| *pFolderPath* | • Path of Input folder [IN] |

**Returns**

TRUE/FALSE

**Note**

In case pFolderPath is invalid, API does not return invalid path error as SLQSSwiGetAllCarrierImages get carrier images from device also.

#### 9.15.4.22 ULONG SLQSSwiGetFirmwareCurr ( CurrentImgList * *pCurrentImgList* )

This API gets the currently active images on the device.

**Parameters**

| | |
|---|---|
| *pCurrentImgList* | • Pointer to structure CurrentImgList<br>• See CurrentImgList 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: NA
Device Supported: MC73xx, MC74xx/EM74xx
Timeout: 5 Secs

#### 9.15.4.23 ULONG SLQSUpgradeFirmware9x07 ( CHAR * *pDestinationPath* )

This API is used to download firmware to a WP760x device.

This API Performs the following steps:

1. Verifies arguments.
2. Informs the SDK of the firmware upgrade path
3. Send the DM command to requests the device reset (device will reset after all open handles are released).

   Upon successful completion, the above steps will have been completed, however, the actual upgrade of the firmware will necessarily then follow.

**Parameters**

| | |
|---|---|
| *pDestination-Path[IN]* | • fully qualified path to firmware image to download. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.15.4.24 ULONG SLQSUpgradeFirmware9x15 ( CHAR * *pDestinationPath* )

This API is used to upgrade firmware on a MC73xx device. This API encapsulates all steps involved in the firmware download process. It is an alternative to any firmware download application. Hence it is a blocking API call. This API will not return until the entire process has been completed.

This API Performs the following steps:

1. Verifies arguments.
2. Retrieve and store the details of the firmware and the PRI file
3. Enable device state change callback.
4. Enable firmware download callback.
5. Set Image preference on the device and reset the device.
6. Wait for the firmware to download and device to become ready.

7. Check the firmware update status. If fail, return an error.
8. If update status is OK, check if current image preference and preferred image preference( from step 2 ) match
9. If match, firmware download is successful.
10. If do not match, repeat from step 5 once more.
11. Disable callbacks and exit.

The call to this API blocks until step 7 or 11. This could be a significant amount of time ( in order of minutes ). Also note that the device state change callback and firmware download callback are used internally

within this API. Hence the user application's instance of these callbacks (if any) are cleared. The user must re-enable these callbacks after a call to this API in order to use them.

**Parameters**

| | |
|---|---|
| *pDestination-Path[IN]* | • fully qualified path to firmware image to download. The path must end with a forward slash. |

**Returns**

- eQCWWAN_ERR_NONE - Firmware download/Switch success.
- eQCWWAN_ERR_INVALID_ARG - The path input does not contain any image
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL - Firmware download/switch failed
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH - Download success but device offline due to image preference mismatch ( ref. syslogs for cause )

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: NA

#### 9.15.4.25 ULONG upgrade_9x07_9x50_fw ( LPCSTR *path* )

This API is used to download firmware to a 9x07 or 9x50 device.

**Parameters**

| | |
|---|---|
| *pDestination-Path[IN]* | • fully qualified path to firmware image to download. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 12 seconds

#### 9.15.4.26 ULONG upgrade_mc77xx_fw ( LPCSTR *path* )

#### 9.15.4.27 ULONG UpgradeFirmware2k ( CHAR * *pDestinationPath* )

This API is used to download firmware to a MC77xx or Gobi 3000 device. For SL909/MC9090, SLQSSetSpkgFormatRequired() needs to be called in advance to specify the download method, Gobi3K or SPKG download, please refer to the API SLQSSetSpkgFormatRequired() for more details of the input values. If SLQSSetSpkgFormatRequired() is not called in advance, it will us Gobi3K firmware download method (MBN files) as the default download method.

This API Performs the following steps:

1. Verifies arguments.
2. Verify that device Crash State should be 1 (RESET State).
3. Informs the SDK of the firmware upgrade path
4. Updates the images preference on the currently connected device.
5. Requests the device reset (device will reset after all open handles are released).

   Upon successful completion, the above steps will have been completed, however, the actual upgrade of the firmware will necessarily then follow.

**Parameters**

| | |
|---|---|
| *pDestination-Path[IN]* | • fully qualified path to firmware image to download. The path must end with a forward slash. For a Gobi 3000 device the path should specify the carrier image folder index i.e. "<path\to\carrier\image>/<carrier index>/" where <carrier index>=""> is a valid sub-directory entry. For 9x30 devices if pDestinationPath is not valid on host, it will use pseudo path for image switching. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 12 seconds

## 9.16 qaGobiApiIms.h File Reference

IMS Service API function prototypes.

### Data Structures

- struct SetSIPConfigReq
- struct SetSIPConfigResp
- struct SetRegMgrConfigReq

- struct SetRegMgrConfigResp
- struct SetIMSSMSConfigReq
- struct SetIMSSMSConfigResp
- struct SetIMSUserConfigReq
- struct SetIMSUserConfigResp
- struct SetIMSVoIPConfigReq
- struct SetIMSVoIPConfigResp
- struct GetSIPConfigResp
- struct GetRegMgrConfigParams
- struct GetIMSSMSConfigParams
- struct GetIMSUserConfigParams
- struct GetIMSVoIPConfigResp
- struct imsCfgIndRegisterInfo

## Functions

- ULONG SLQSSetSIPConfig (SetSIPConfigReq *pSetSIPConfigReq, SetSIPConfigResp *pSetSIPConfigResp)
- ULONG SLQSSetRegMgrConfig (SetRegMgrConfigReq *pSetRegMgrConfigReq, SetRegMgrConfigResp *pSetRegMgrConfigResp)
- ULONG SLQSSetIMSSMSConfig (SetIMSSMSConfigReq *pSetIMSSMSConfigReq, SetIMSSMSConfigResp *pSetIMSSMSConfigResp)
- ULONG SLQSSetIMSUserConfig (SetIMSUserConfigReq *pSetIMSUserConfigReq, SetIMSUserConfigResp *pSetIMSUserConfigResp)
- ULONG SLQSSetIMSVoIPConfig (SetIMSVoIPConfigReq *pSetIMSVoIPConfigReq, SetIMSVoIPConfigResp *pSetIMSVoIPConfigResp)
- ULONG SLQSGetSIPConfig (GetSIPConfigResp *pGetSIPConfigResp)
- ULONG SLQSGetRegMgrConfig (GetRegMgrConfigParams *pGetRegMgrConfigParams)
- ULONG SLQSGetIMSSMSConfig (GetIMSSMSConfigParams *pGetIMSSMSConfigParams)
- ULONG SLQSGetIMSUserConfig (GetIMSUserConfigParams *pGetIMSUserConfigParams)
- ULONG SLQSGetIMSVoIPConfig (GetIMSVoIPConfigResp *pGetIMSVoIPConfigResp)
- ULONG SLQSImsConfigIndicationRegister (imsCfgIndRegisterInfo *pImsCfgIndRegisterInfo)

### 9.16.1 Detailed Description

IMS Service API function prototypes.

### 9.16.2 Function Documentation

#### 9.16.2.1 ULONG SLQSGetIMSSMSConfig ( GetIMSSMSConfigParams * *pGetIMSSMSConfigParams* )

This API retrieves the SMS configuration parameters.

**Parameters**

| | |
|---|---|
| *pGetIMSSMSConfigParams[IN/OUT]* | • See GetIMSSMSConfigParams 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.2 ULONG SLQSGetIMSUserConfig ( GetIMSUserConfigParams * *pGetIMSUserConfigParams* )

This API retrieves the IMS User configuration parameters.

**Parameters**

| | |
|---|---|
| *pGetIMSUserConfigParams[IN/OUT]* | • See GetIMSUserConfigParams 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.3 ULONG SLQSGetIMSVoIPConfig ( GetIMSVoIPConfigResp * *pGetIMSVoIPConfigResp* )

This API retrieves the IMS VoIP configuration parameters.

**Parameters**

| | |
|---|---|
| *GetIMSVoIPConfigResp[OUT]* | • See GetIMSVoIPConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.4 ULONG SLQSGetRegMgrConfig ( GetRegMgrConfigParams * *pGetRegMgrConfigParams* )

This API retrieves the registration manager configuration parameters.

**Parameters**

| | |
|---|---|
| *pGetRegMgrConfigParams[IN/OUT]* | • See GetRegMgrConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.5 ULONG SLQSGetSIPConfig ( GetSIPConfigResp * *pGetSIPConfigResp* )

This API retrieves the Session Initiation Protocol(SIP) configuration parameters.

**Parameters**

| | |
|---|---|
| *pGetSIPConfigResp[OUT]* | • See GetSIPConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.6 ULONG SLQSImsConfigIndicationRegister ( imsCfgIndRegisterInfo * *pImsCfgIndRegisterInfo* )

Sets the registration state for different QMI_IMS indications for the requesting control point

**Parameters**

| | |
|---|---|
| *pImsCfgInd-RegisterInfo[IN]* | • Structure containing Indication Register Information.<br>– See imsCfgIndRegisterInfo for more informtaion. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 10 Secs

This API 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.

#### 9.16.2.7 ULONG SLQSSetIMSSMSConfig ( SetIMSSMSConfigReq * *pSetIMSSMSConfigReq,* SetIMSSMSConfigResp * *pSetIMSSMSConfigResp* )

This API sets the IMS SMS configuration parameters for the requesting control point.

**Parameters**

| | |
|---|---|
| *pSetIMSSMS-ConfigReq[IN]* | • See SetIMSSMSConfigReq for more information |
| *pSetIMSSMS-ConfigResp[OU-T]* | • See SetIMSSMSConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.8 ULONG SLQSSetIMSUserConfig ( SetIMSUserConfigReq * *pSetIMSUserConfigReq*, SetIMSUserConfigResp * *pSetIMSUserConfigResp* )

This API sets the IMS user configuration parameters for the requesting control point.

**Parameters**

| | |
|---|---|
| *pSetIMSUserConfigReq[IN]* | • See SetIMSUserConfigReq for more information |
| *pSetIMSUserConfigResp[OUT]* | • See SetIMSUserConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.9 ULONG SLQSSetIMSVoIPConfig ( SetIMSVoIPConfigReq * *pSetIMSVoIPConfigReq*, SetIMSVoIPConfigResp * *pSetIMSVoIPConfigResp* )

This API sets the IMS Voice over Internet Protocol (VoIP) configuration parameters for the requesting control point.

**Parameters**

| | |
|---|---|
| *pSetIMSVoIPConfigReq[IN]* | • See SetIMSVoIPConfigReq for more information |
| *pSetIMSVoIPConfigResp[OUT]* | • See SetIMSVoIPConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.10 ULONG SLQSSetRegMgrConfig ( SetRegMgrConfigReq * *pSetRegMgrConfigReq,* SetRegMgrConfigResp * *pSetRegMgrConfigResp* )

This API sets the IMS registration manager configuration parameters for the requesting control point.

**Parameters**

| | |
|---|---|
| *pSetRegMgrConfigReq[IN]* | • See SetRegMgrConfigReq for more information |
| *pSetRegMgrConfigResp[OUT]* | • See SetRegMgrConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

#### 9.16.2.11 ULONG SLQSSetSIPConfig ( SetSIPConfigReq * *pSetSIPConfigReq,* SetSIPConfigResp * *pSetSIPConfigResp* )

This API sets the IMS Session Initiation Protocol(SIP) configuration parameters for the requesting control point.

**Parameters**

| | |
|---|---|
| *pSetSIPConfigReq[IN]* | • See SetSIPConfigReq for more information |
| *pSetSIPConfigResp[OUT]* | • See SetSIPConfigResp 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: NA
Device Supported: MC73xx, MC74xx and EM74xx
Timeout: 5 seconds

## 9.17 qaGobiApiImsa.h File Reference

IMSA Service API function prototypes.

### Data Structures

- struct IMSAIndRegisterInfo
- struct SupportedMsgList
- struct IMSASupportedMsgInfo
- struct ReqFieldsList
- struct RespFieldsList
- struct IndFieldsList
- struct IMSASupportedFieldsResp
- struct IMSARegistrationStatus
- struct IMSAServiceStatus

### Functions

- ULONG SLQSRegisterIMSAIndication (IMSAIndRegisterInfo *pImsaIndRegisterInfo)
- ULONG SLQSGetIMSASupportedMsg (IMSASupportedMsgInfo *pIMSASupportedMsgInfo)
- ULONG SLQSGetIMSASupportedFields (WORD messageID, IMSASupportedFieldsResp *pIMSASupportedFieldsResp)
- ULONG SLQSGetIMSARegStatus (IMSARegistrationStatus *pIMSARegistrationStatus)
- ULONG SLQSGetIMSAServiceStatus (IMSAServiceStatus *pIMSAServiceStatus)

### 9.17.1 Detailed Description

IMSA Service API function prototypes.

### 9.17.2 Function Documentation

#### 9.17.2.1 ULONG SLQSGetIMSARegStatus ( IMSARegistrationStatus * *pIMSARegistrationStatus* )

Queries the set of messages implemented by the currently running software.

**Parameters**

| | |
|---|---|
| *pIMSA-Registration-Status[OUT]* | • Structure containing response parameters for registration status.<br>– See IMSARegistrationStatus 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: 5 Secs

```
This API is used by a device to get the registration status for
various IMS services for the requesting control point.
```

#### 9.17.2.2 ULONG SLQSGetIMSAServiceStatus ( IMSAServiceStatus * *pIMSAServiceStatus* )

Gets the service status for various IMS services for the requesting control point.

**Parameters**

| | |
|---|---|
| *pIMSAService-Status[OUT]* | • Structure containing response parameters for service status.<br>– See IMSAServiceStatus 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: 5 Secs

```
This API is used by a device to Gets the service status for various
IMS services for the requesting control point.
```

#### 9.17.2.3 ULONG SLQSGetIMSASupportedFields ( WORD *messageID*, IMSASupportedFieldsResp * *pIMSASupportedFieldsResp* )

Queries the set of supported fields implemented by the currently running software.

**Parameters**

| | |
|---|---|
| *messageID[IN]* | • Service Message ID. |
| *pIMSA-Supported-FieldsResp[OU-T]* | • Structure containing Supported Fields Response.<br>– See IMSASupportedFieldsResp 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: 5 Secs

```
This API is used by a device to query the fields supported for a
single command as implemented by the currently running software.
```

#### 9.17.2.4 ULONG SLQSGetIMSASupportedMsg ( IMSASupportedMsgInfo * *pIMSASupportedMsgInfo* )

Queries the set of messages implemented by the currently running software.

**Parameters**

| | |
|---|---|
| *pIMSA-SupportedMsg-Info[OUT]* | • Structure containing Supported Messages Information.<br>– See IMSASupportedMsgInfo 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: 5 Secs

```
This API is used by a device to query the set of messages implemented by
the currently running software
```

#### 9.17.2.5 ULONG SLQSRegisterIMSAIndication ( IMSAIndRegisterInfo * *pImsaIndRegisterInfo* )

Sets the registration state for different QMI_IMSA indications for the requesting control point

**Parameters**

| | |
|---|---|
| *pImsaInd-RegisterInfo[IN]* | • Structure containing Indication Register Information.<br>– See IMSAIndRegisterInfo 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: 5 Secs

```
This API is used by a device to register/deregister for different
QMI IMSA 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.
```

## 9.18 qaGobiApiLoc.h File Reference

Location API function prototypes.

### Data Structures

- struct LOCEventRegisterReqResp
- struct LOCExtPowerStateReqResp
- struct LocApplicationInfo
- struct LOCStartReq
- struct LOCStopReq
- struct SV
- struct SVInfo
- struct GnssData
- struct CellDb
- struct ClkInfo
- struct BdsSV
- struct BdsSVInfo
- struct LocDelAssDataReq
- struct SwiLocGetAutoStartResp
- struct SwiLocSetAutoStartReq
- struct altitudeSrcInfo
- struct LocInjectPositionReq
- struct LocSetCradleMountReq
- struct sensorData
- struct tempratureData
- struct LocInjectSensorDataReq
- struct LocGetServerReq
- struct IPv4Config
- struct IPv6Config
- struct URLAddrInfo
- struct LocSetServerReq

### Macros

- #define MAX_SENSOR_DATA_LEN 64
- #define MAX_TEMP_DATA_LEN 64

### Functions

- ULONG SLQSLOCEventRegister (LOCEventRegisterReqResp ∗pLOCEventRegisterReqResp)
- ULONG SLQSLOCSetExtPowerState (LOCExtPowerStateReqResp ∗pLOCExtPowerStateReqResp)
- ULONG SLQSLOCStart (LOCStartReq ∗pLOCStartReq)
- ULONG SLQSLOCStop (LOCStopReq ∗pLOCStopReq)
- ULONG SLQSLOCSetOpMode (ULONG mode)
- ULONG SLQSLOCDelAssData (LocDelAssDataReq request)
- ULONG SwiLocGetAutoStart (SwiLocGetAutoStartResp ∗resp)
- ULONG SwiLocSetAutoStart (SwiLocSetAutoStartReq ∗req)
- ULONG SLQSLOCInjectUTCTime (ULONGLONG timeMsec, ULONG timeUncMsec)
- ULONG SLQSLOCInjectPosition (LocInjectPositionReq ∗pLocInjectPositionReq)
- ULONG SLQSLOCSetCradleMountConfig (LocSetCradleMountReq ∗pLocSetCradleMountReq)
- ULONG SLQSLOCInjectSensorData (LocInjectSensorDataReq ∗pLocInjectSensorDataReq)
- ULONG SLQSLOCGetBestAvailPos (ULONG xid)
- ULONG SLQSLOCGetServer (LocGetServerReq ∗pLocGetServerReq)
- ULONG SLQSLOCSetServer (LocSetServerReq ∗pLocSetServerReq)

## 9.18.1 Detailed Description

Location API function prototypes.

## 9.18.2 Macro Definition Documentation

### 9.18.2.1 #define MAX_SENSOR_DATA_LEN 64

### 9.18.2.2 #define MAX_TEMP_DATA_LEN 64

## 9.18.3 Function Documentation

### 9.18.3.1 ULONG SLQSLOCDelAssData ( LocDelAssDataReq *request* )

Used by the control point to delete the location engine assistance data

**Parameters**

| | |
|---|---|
| *request[IN]* | • request structure parameters should contain all NULL pointers to delete all assistance data. Otherwise, specify optional fields to be deleted. See LocDelAssDataReq 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: 5 seconds

#### 9.18.3.2 ULONG SLQSLOCEventRegister ( LOCEventRegisterReqResp * *pLOCEventRegisterReqResp* )

Used by the control point to register for events from the location subsystem.

**Parameters**

| | |
|---|---|
| *pLOCEvent-RegisterReq-Resp[IN]* | • See LOCEventRegisterReqResp 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: 5 seconds

#### 9.18.3.3 ULONG SLQSLOCGetBestAvailPos ( ULONG *xid* )

Control point to get the best available position estimate from the location engine.

**Parameters**

| | |
|---|---|
| *xid[IN]* | • Identifies the transaction.<br>• The transaction ID is returned in the Get Best Available Position indication. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

- GPS engine should be started to get best available position.

#### 9.18.3.4 ULONG SLQSLOCGetServer ( LocGetServerReq * *pLocGetServerReq* )

Gets the A-GPS server address from the engine. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through SetLocGetServerCallback callback If successful, the callback also contains the A-GPS server address

**Parameters**

| | |
|---|---|
| *pLocGetServer-Req[IN]* | • See LocGetServerReq 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: 5 seconds

#### 9.18.3.5 ULONG SLQSLOCInjectPosition ( LocInjectPositionReq * *pLocInjectPositionReq* )

Injects a position to the location engine.

**Parameters**

| | |
|---|---|
| *pLocInject-PositionReq[IN]* | • See LocInjectPositionReq 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: 5 seconds

#### 9.18.3.6 ULONG SLQSLOCInjectSensorData ( LocInjectSensorDataReq * *pLocInjectSensorDataReq* )

Control point to to inject sensor data into the GNSS location engine.

**Parameters**

| | |
|---|---|
| *pLocInject-SensorData-Req[IN]* | • See LocInjectSensorDataReq 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: 5 seconds

#### 9.18.3.7 ULONG SLQSLOCInjectUTCTime ( ULONGLONG *timeMsec,* ULONG *timeUncMsec* )

Injects UTC time in the location engine.

**Parameters**

| | |
|---|---|
| *timeMsec[IN]* | • The UTC time since Jan. 1, 1970 |
| *timeUncMsec[IN]* | • The time Uncertainty |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.18.3.8 ULONG SLQSLOCSetCradleMountConfig ( LocSetCradleMountReq * *pLocSetCradleMountReq* )

Control point to set the current cradle mount configuration.

**Parameters**

| | |
|---|---|
| *pLocSetCradleMountReq[IN]* | • See LocSetCradleMountReq 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: 5 seconds

#### 9.18.3.9 ULONG SLQSLOCSetExtPowerState ( LOCExtPowerStateReqResp * *pLOCExtPowerStateReqResp* )

Used by the control point to set the current external power configuration.

**Parameters**

| | |
|---|---|
| *pLOCExtPowerStateReqResp[IN]* | • See LOCExtPowerStateReqResp 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: 5 seconds

#### 9.18.3.10 ULONG SLQSLOCSetOpMode ( ULONG *mode* )

Used by the control point to tells the engine to use the specified operation mode while making the position fixes

**Parameters**

| | |
|---|---|
| *mode[IN]* | • 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 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.18.3.11 ULONG SLQSLOCSetServer ( LocSetServerReq * *pLocSetServerReq* )

Specifies the A-GPS server type and address.

**Parameters**

| | |
|---|---|
| *pLocSetServer-Req[IN]* | • See LocSetServerReq 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: 5 seconds
  - 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.

#### 9.18.3.12 ULONG SLQSLOCStart ( LOCStartReq * *pLOCStartReq* )

Used by the control point to initiate a GPS session.

**Parameters**

| | |
|---|---|
| *pLOCStartReq-Req[IN]* | • See LOCStartReq 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: 5 seconds

#### 9.18.3.13 ULONG SLQSLOCStop ( LOCStopReq * *pLOCStopReq* )

Used by the control point to stop a GPS session.

**Parameters**

| | |
|---|---|
| *pLOCStopReq-Resp[IN]* | • See LOCStopReq 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: 5 seconds

#### 9.18.3.14 ULONG SwiLocGetAutoStart ( SwiLocGetAutoStartResp * *resp* )

Used by the control point to Get Loc Auto Start settings

**Parameters**

| | |
|---|---|
| *resp[OUT]* | • See SwiLocGetAutoStartResp for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.18.3.15 ULONG SwiLocSetAutoStart ( SwiLocSetAutoStartReq * *req* )

Used by the control point to Set Loc Auto Start settings

**Parameters**

| | |
|---|---|
| *req[IN]* | • See SwiLocSetAutoStartReq for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.19 qaGobiApiNas.h File Reference

Network Access Service API function prototypes.

## Data Structures

- struct RFBandInfoElements
- struct RfDedicatedBandInfoElements
- struct RfBandInfoExtFormatElements
- struct RfBandwidthInfoElements
- struct RfBandInfoList
- struct RfDedicatedBandInfo
- struct RfBandInfoExtFormat
- struct RfBandwidthInfo
- struct LTEOperationMode
- struct NasGetRFInfoResp
- struct servSystem
- struct dataSrvCapabilities
- struct currentPLMN
- struct roamIndList
- struct qaQmi3Gpp2TimeZone
- struct detailSvcInfo
- struct CDMASysInfoExt
- struct callBarStatus
- struct qaQmiServingSystemParam
- struct rxSignalStrengthListElement
- struct ecioListElement
- struct errorRateListElement
- struct rsrqInformation
- struct lteSnrinformation
- struct lteRsrpinformation
- struct slqsSignalStrengthInfo
- struct SlqsNas3GppNetworkInfo
- struct _SlqsNas3GppNetworkRAT_
- struct SlqsNasPcsDigit
- struct _slqsNetworkScanInfo
- struct netSelectionPref
- struct acqOrderPref
- struct CSGID
- struct _sysSelectPrefParams
- struct ciotAcqOrderPref
- struct nr5gBandPref
- struct sysSelectPrefParamsExt
- struct _sysSelectPrefInfo
- struct SrvStatusInfo
- struct GSMSrvStatusInfo
- struct sysInfoCommon
- struct CDMASysInfo
- struct HDRSysInfo
- struct GSMSysInfo
- struct WCDMASysInfo
- struct LTESysInfo
- struct AddCDMASysInfo
- struct AddSysInfo
- struct CallBarringSysInfo
- struct nasGetSysInfoResp
- struct CommInfo
- struct LTEInfo
- struct swiModemStatusResp

- struct nasGetHDRColorCodeResp
- struct nasGetTxRxInfoReq
- struct rxInfo
- struct txInfo
- struct nasGetTxRxInfoResp
- struct CDMASSInfo
- struct HDRSSInfo
- struct LTESSInfo
- struct TDSCDMASigInfoExt
- struct nasGetSigInfoResp
- struct nasIndicationRegisterReq
- struct nasPLMNNameReq
- struct nasPLMNNameResp
- struct OperatorPLMNData
- struct operatorPLMNList
- struct serviceProviderName
- struct PLMNNetworkNameData
- struct PLMNNetworkName
- struct operatorNameString
- struct nasOperatorNameResp
- struct nasGet3GPP2SubscriptionInfoReq
- struct namName
- struct dirNum
- struct sidNid
- struct homeSIDNID
- struct minBasedIMSI
- struct trueIMSI
- struct CDMAChannel
- struct nasGet3GPP2SubscriptionInfoResp
- struct nmrCellInfo
- struct GERANInfo
- struct geranInstInfo
- struct UMTSinstInfo
- struct UMTSInfo
- struct CDMAInfo
- struct cellParams
- struct LTEInfoIntrafreq
- struct infoInterFreq
- struct LTEInfoInterfreq
- struct gsmCellInfo
- struct lteGsmCellInfo
- struct LTEInfoNeighboringGSM
- struct wcdmaCellInfo
- struct lteWcdmaCellInfo
- struct LTEInfoNeighboringWCDMA
- struct umtsLTENbrCell
- struct WCDMAInfoLTENeighborCell
- struct nasCellLocationInfoResp
- struct MNRInfo
- struct nasInitNetworkReg
- struct protocolSubtypeElement
- struct HDRPersonalityResp
- struct HDRProtSubtypResp
- struct PSDetachReq
- struct GetErrRateResp

- struct DRCParams
- struct PilotSetParams
- struct PilotSetData
- struct GetHRPDStatsResp
- struct ActPilotPNElement
- struct NetworkStat1x
- struct NetworkStatEVDO
- struct DeviceConfigDetail
- struct DataStatusDetail
- struct NetworkDebugResp
- struct LteCQIParm
- struct RSSIThresh
- struct ECIOThresh
- struct HDRSINRThresh
- struct LTESNRThresh
- struct IOThresh
- struct RSRQThresh
- struct RSRPThresh
- struct LTESigRptCfg
- struct TDSCDMASINRCONFThresh
- struct sigInfo
- struct NasSwiIndReg
- struct CDMARSSIThresh
- struct CDMAECIOThresh
- struct HDRRSSIThresh
- struct HDRECIOThresh
- struct HDRSINRThreshold
- struct HDRIOThresh
- struct GSMRSSIThresh
- struct WCDMARSSIThresh
- struct WCDMAECIOThresh
- struct LTERSSIThresh
- struct LTESNRThreshold
- struct LTERSRQThresh
- struct LTERSRPThresh
- struct LTESigRptConfig
- struct TDSCDMARSCPThresh
- struct TDSCDMARSSIThresh
- struct TDSCDMAECIOThresh
- struct TDSCDMASINRThresh
- struct setSignalStrengthInfo
- struct PhyCaAggScellIndType
- struct PhyCaAggScellDlBw
- struct PhyCaAggScellInfo
- struct PhyCaAggPcellInfo
- struct PhyCaAggScellIndex
- struct nasGetLTECphyCaResp
- struct nasGetLTECphyCa
- struct wcdmaUARFCN
- struct lteEARFCN
- struct ltePCI
- struct nasSwiGetChannelLockResp
- struct nasSwiSetChannelLockReq
- struct timeInfo
- struct GetNetworkTimeResp

- struct RxSigInfo
- struct SccRxInfo
- struct LteSccRxInfoResp
- struct nasSeteDRXParamsReq
- struct nasGeteDRXParamsResp

## Macros

- #define SLQS_SS_INFO_LIST_MAX_ELEMENTS 18
- #define MAX_DESCRIPTION_LENGTH 255
- #define SLQS_SYSTEM_ID_SIZE 16
- #define PLMN_LENGTH 3
- #define MAX_SERV_SYSTEM_RADIO_INTERFACES 0x0A
- #define MAX_DATA_SRV_CAPABILITIES 0x20
- #define NAM_NAME_LENGTH 12
- #define IMSI_M_S1_LENGTH 7
- #define IMSI_M_S2_LENGTH 3
- #define MAX_PILOT_SETS 0xFF
- #define UATISIZE 16
- #define NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE 16
- #define NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE -125.0
- #define NAS_SIG_INFO_MIN_dB_FLOAT_VALUE -10.0
- #define NAS_MAX_SCC_RX_INFO_INSTANCES 255

## Typedefs

- typedef struct _SlqsNas3GppNetworkRAT_ SlqsNas3GppNetworkRAT
- typedef struct _slqsNetworkScanInfo slqsNetworkScanInfo
- typedef struct _sysSelectPrefParams sysSelectPrefParams
- typedef struct _sysSelectPrefInfo sysSelectPrefInfo

## Enumerations

- enum _NAMS_RADIO_IF_TECHNOLOGY_ {
  eNAS_RADIO_IF_GSM = 0x04,
  eNAS_RADIO_IF_UMTS = 0x05,
  eNAS_RADIO_IF_LTE = 0x08,
  eNAS_RADIO_IF_TDSCDMA = 0x09 }
- enum NAS_LTE_CPHY_SCELL_STATE {
  eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED =0x00,
  eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED =0x01,
  eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED =0x02 }
- enum NAS_LTE_CPHY_CA_BW_NRB {
  eNAS_LTE_CPHY_CA_BW_NRB_6 =0x00,
  eNAS_LTE_CPHY_CA_BW_NRB_15 =0x01,
  eNAS_LTE_CPHY_CA_BW_NRB_25 =0x02,
  eNAS_LTE_CPHY_CA_BW_NRB_50 =0x03,
  eNAS_LTE_CPHY_CA_BW_NRB_75 =0x04,
  eNAS_LTE_CPHY_CA_BW_NRB_100 =0x05 }

- enum eSYS_SRV_DOMAIN {
  eSYS_SRV_DOMAIN_NO_SRV = 0x00,
  eSYS_SRV_DOMAIN_CS_ONLY = 0x01,
  eSYS_SRV_DOMAIN_PS_ONLY = 0x02,
  eSYS_SRV_DOMAIN_CS_PS = 0x03,
  eSYS_SRV_DOMAIN_CAMPED = 0x04,
  eSYS_SRV_DOMAIN_UNKNOWN }

## Functions

- ULONG GetSignalStrengths (ULONG *pArraySizes, INT8 *pSignalStrength, ULONG *pRadioInterface)
- ULONG PerformNetworkScan (BYTE *pInstanceSize, BYTE *pInstances)
- ULONG InitiateNetworkRegistration (ULONG regType, WORD mcc, WORD mnc, ULONG rat)
- ULONG GetServingNetwork (ULONG *pRegistrationState, ULONG *pCSDomain, ULONG *pPSDomain, ULONG *pRAN, BYTE *pRadioIfacesSize, BYTE *pRadioIfaces, ULONG *pRoaming, WORD *pMCC, WORD *pMNC, BYTE nameSize, CHAR *pName)
- ULONG GetHomeNetwork (WORD *pMCC, WORD *pMNC, BYTE nameSize, CHAR *pName, WORD *pSID, WORD *pNID)
- ULONG GetServingNetworkCapabilities (BYTE *pDataCapsSize, BYTE *pDataCaps)
- ULONG SetNetworkPreference (ULONG technologyPref, ULONG duration)
- ULONG GetNetworkPreference (ULONG *pTechnologyPref, ULONG *pDuration, ULONG *pPersistentTechnologyPref)
- ULONG GetRFInfo (BYTE *pInstanceSize, struct RFBandInfoElements *pRFBandInfo)
- ULONG SLQSNasGetRFInfo (NasGetRFInfoResp *pGetRFInfoResp)
- ULONG InitiateDomainAttach (ULONG action)
- ULONG GetACCOLC (BYTE *pACCOLC)
- ULONG SetACCOLC (CHAR *spc, BYTE accolc)
- ULONG SetCDMANetworkParameters (CHAR *pSPC, BYTE *pForceRev0, BYTE *pCustomSCP, ULONG *pProtocol, ULONG *pBroadcast, ULONG *pApplication, ULONG *pRoaming)
- ULONG GetCDMANetworkParameters (BYTE *pSCI, BYTE *pSCM, BYTE *pRegHomeSID, BYTE *pRegForeignSID, BYTE *pRegForeignNID, BYTE *pForceRev0, BYTE *pCustomSCP, ULONG *pProtocol, ULONG *pBroadcast, ULONG *pApplication, ULONG *pRoaming)
- ULONG GetANAAAAuthenticationStatus (ULONG *pStatus)
- ULONG SLQSGetServingSystem (qaQmiServingSystemParam *pServingSystem)
- ULONG SLQSSetBandPreference (ULONGLONG bandpreference)
- ULONG SLQSNasIndicationRegister (BYTE systemSelectionInd, BYTE DDTMInd, BYTE servingSystemInd)
- ULONG SLQSGetSignalStrength (struct slqsSignalStrengthInfo *pSignalInfo)
- ULONG SLQSPerformNetworkScan (slqsNetworkScanInfo *pNetworkInfo)
- ULONG SLQSSetSysSelectionPref (sysSelectPrefParams *pSysSelectPrefParams)
- ULONG SLQSSetSysSelectionPrefExt (sysSelectPrefParamsExt *pSysSelectPrefParamsExt)
- ULONG SLQSGetSysSelectionPref (sysSelectPrefInfo *pSysSelectPrefInfo)
- ULONG SLQSNasGetSysInfo (nasGetSysInfoResp *pGetSysInfoResp)
- ULONG SLQSNasSwiModemStatus (swiModemStatusResp *pModemStatusResp)
- ULONG SLQSNasGetHDRColorCode (nasGetHDRColorCodeResp *pGetHDRColorCodeResp)
- ULONG SLQSNasGetTxRxInfo (nasGetTxRxInfoReq *pGetTxRxInfoReq, nasGetTxRxInfoResp *pGetTxRxInfoResp)
- ULONG SLQSNasGetSigInfo (nasGetSigInfoResp *pGetSigInfoResp)
- ULONG SLQSNasIndicationRegisterExt (nasIndicationRegisterReq *pIndicationRegisterReq)
- ULONG SLQSGetPLMNName (nasPLMNNameReq *pPLMNNameReq, nasPLMNNameResp *pPLMNNameResp)
- ULONG SLQSGetOperatorNameData (nasOperatorNameResp *pOperatorNameData)
- ULONG SLQSNasGet3GPP2Subscription (nasGet3GPP2SubscriptionInfoReq *pGet3GPP2SubsInfoReq, nasGet3GPP2SubscriptionInfoResp *pGet3GPP2SubsInfoResp)
- ULONG SLQSNasGetCellLocationInfo (nasCellLocationInfoResp *pNasCellLocationInfoResp)
- ULONG SLQSInitiateNetworkRegistration (nasInitNetworkReg *pNasInitNetRegistrationReg)

- ULONG SLQSSwiGetHDRPersonality (HDRPersonalityResp *pHDRPersonalityResp)
- ULONG SLQSSwiGetHDRProtSubtype (HDRProtSubtypResp *pHDRProtSubtypResp)
- ULONG SLQSSwiPSDetach (PSDetachReq *pPSDetachReq)
- ULONG SLQSGetErrorRate (GetErrRateResp *pGetErrRateResp)
- ULONG SLQSSwiGetHRPDStats (GetHRPDStatsResp *pGetHRPDStatsResp)
- ULONG SLQSSwiNetworkDebug (NetworkDebugResp *pNetworkDebugResp)
- ULONG SLQSSwiGetLteCQI (LteCQIParm *pLteCQIResp)
- ULONG SLQSConfigSigInfo (sigInfo *pSigInfo)
- ULONG SLQSNasSwiIndicationRegister (NasSwiIndReg *pIndRegReq)
- ULONG GetHomeNetwork3GPP2 (WORD *pMCC, WORD *pMNC, BYTE nameSize, CHAR *pName, WORD *pSID, WORD *pNID, WORD *pNw2MCC, WORD *pNw2MNC, BYTE *pNw2DescDisp, BYTE *pNw2DescEnc, BYTE *pNw2DescLen, BYTE *pNw2Name)
- ULONG SLQSNasConfigSigInfo2 (setSignalStrengthInfo *pSetSignalStrengthInfo)
- ULONG SLQSNASGetLTECPHYCaInfo (nasGetLTECphyCa *pLTECPhyCa)
- ULONG SLQSNasIndicationRegisterLTECphyCa (BYTE *bStatus)
- ULONG SLQSNASSwiGetChannelLock (nasSwiGetChannelLockResp *pNasSwiGetChannelLockResp)
- ULONG SLQSNASSwiSetChannelLock (nasSwiSetChannelLockReq *pNasSwiSetChannelLockReq)
- ULONG SLQSGetNetworkTime (GetNetworkTimeResp *pGetNetworkTimeResp)
- ULONG SLQSSwiGetLteSccRxInfo (LteSccRxInfoResp *pLteSccRxInfoResp)
- ULONG SLQSNASSeteDRXParams (nasSeteDRXParamsReq *pNasSeteDRXParamsReq)
- ULONG SLQSNASGeteDRXParams (nasGeteDRXParamsResp *pNasGeteDRXParamsResp)

### 9.19.1 Detailed Description

Network Access Service API function prototypes.

### 9.19.2 Macro Definition Documentation

#### 9.19.2.1 #define IMSI_M_S1_LENGTH 7

#### 9.19.2.2 #define IMSI_M_S2_LENGTH 3

#### 9.19.2.3 #define MAX_DATA_SRV_CAPABILITIES 0x20

#### 9.19.2.4 #define MAX_DESCRIPTION_LENGTH 255

#### 9.19.2.5 #define MAX_PILOT_SETS 0xFF

#### 9.19.2.6 #define MAX_SERV_SYSTEM_RADIO_INTERFACES 0x0A

#### 9.19.2.7 #define NAM_NAME_LENGTH 12

#### 9.19.2.8 #define NAS_MAX_SCC_RX_INFO_INSTANCES 255

#### 9.19.2.9 #define NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE 16

#### 9.19.2.10 #define NAS_SIG_INFO_MIN_dB_FLOAT_VALUE -10.0

#### 9.19.2.11 #define NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE -125.0

#### 9.19.2.12 #define PLMN_LENGTH 3

#### 9.19.2.13 #define SLQS_SS_INFO_LIST_MAX_ELEMENTS 18

#### 9.19.2.14 #define SLQS_SYSTEM_ID_SIZE 16

#### 9.19.2.15 #define UATISIZE 16

### 9.19.3 Typedef Documentation

#### 9.19.3.1 typedef struct _SlqsNas3GppNetworkRAT_ SlqsNas3GppNetworkRAT

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 |

#### 9.19.3.2 typedef struct _slqsNetworkScanInfo slqsNetworkScanInfo

Contain the network scan information.

**Parameters**

| | |
|---|---|
| *pNetworkInfo-Instances[IN/O-UT]* | • 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. |
| *pNetworkInfo[O-UT]* | • Network info instance array<br>– See SlqsNas3GppNetworkInfo for more information |
| *pRATInstances[-IN/OUT]* | • 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. |
| *pRATInfo[OUT]* | • RAT info instance array<br>– See SlqsNas3GppNetworkRAT for more information |
| *pPCSDigit-Instances[IN/O-UT]* | • 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. |

| | |
|---|---|
| *pPCSDigitInfo[-OUT]* | • PCS Digit info instance array<br>– See SlqsNasPcsDigit for more information |
| *pScanResult[OUT]* | • 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 |

#### 9.19.3.3 typedef struct _sysSelectPrefInfo sysSelectPrefInfo

Structure for storing the current preferred system selection settings for the device.

**Parameters**

| | |
|---|---|
| *pEmerMode* | • Optional parameter specifying the emergency Mode<br>• Values:<br>– 0 - OFF (normal)<br>– 1 - ON (Emergency)<br>• function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device. |
| *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>• function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device. |

| | |
|---|---|
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device. |

| | |
|---|---|
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device. |
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device. |

| | |
|---|---|
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device. |

| | |
|---|---|
| *pNetSelPref* | • Optional parameter indicating network selection preference<br>• Values:<br>– 0x00 - Automatic network selection<br>– 0x01 - Manual network selection<br>• function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device. |
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFF if no value is returned by the device. |
| *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<br>• function SLQSGetSysSelectionPref() returns a default value FFFFFFFF if no value is returned by the device. |

**Note**

None

#### 9.19.3.4 typedef struct _sysSelectPrefParams sysSelectPrefParams

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 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 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 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 |

### 9.19.4 Enumeration Type Documentation

#### 9.19.4.1 enum _NAMS_RADIO_IF_TECHNOLOGY_

**Enumerator**

***eNAS_RADIO_IF_GSM***

***eNAS_RADIO_IF_UMTS***

***eNAS_RADIO_IF_LTE***

***eNAS_RADIO_IF_TDSCDMA***

#### 9.19.4.2 enum eSYS_SRV_DOMAIN

**Enumerator**

***eSYS_SRV_DOMAIN_NO_SRV***

***eSYS_SRV_DOMAIN_CS_ONLY***

***eSYS_SRV_DOMAIN_PS_ONLY***

***eSYS_SRV_DOMAIN_CS_PS***

***eSYS_SRV_DOMAIN_CAMPED***

***eSYS_SRV_DOMAIN_UNKNOWN***

#### 9.19.4.3 enum NAS_LTE_CPHY_CA_BW_NRB

**Enumerator**

***eNAS_LTE_CPHY_CA_BW_NRB_6***

***eNAS_LTE_CPHY_CA_BW_NRB_15***

***eNAS_LTE_CPHY_CA_BW_NRB_25***

***eNAS_LTE_CPHY_CA_BW_NRB_50***

***eNAS_LTE_CPHY_CA_BW_NRB_75***

***eNAS_LTE_CPHY_CA_BW_NRB_100***

#### 9.19.4.4 enum NAS_LTE_CPHY_SCELL_STATE

**Enumerator**

***eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED***

***eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED***

***eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED***

### 9.19.5 Function Documentation

#### 9.19.5.1 ULONG GetACCOLC ( BYTE * *pACCOLC* )

Retrieves information about the access overload class (ACCOLC)

**Parameters**

| | |
|---|---|
| *pACCOLC[OUT]* | • ACCOLC : Valid range is 0 to 15 |

**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
Timeout: 2 seconds

#### 9.19.5.2 ULONG GetANAAAAuthenticationStatus ( ULONG * *pStatus* )

AN-AAA authentication status of the device.

**Parameters**

| | |
|---|---|
| *pStatus[OUT]* | • Status of last AN-AAA authentication attempt<br>– 0 - Failure<br>– 1 - Success<br>– 2 - Not Requested |

**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
Timeout: 2 seconds

#### 9.19.5.3 ULONG GetCDMANetworkParameters ( BYTE * *pSCI*, BYTE * *pSCM*, BYTE * *pRegHomeSID*, BYTE * *pRegForeignSID*, BYTE * *pRegForeignNID*, BYTE * *pForceRev0*, BYTE * *pCustomSCP*, ULONG * *pProtocol*, ULONG * *pBroadcast*, ULONG * *pApplication*, ULONG * *pRoaming* )

Gets the current CDMA network parameters

**Parameters**

| | |
|---|---|
| *pSCI[OUT]* | • Slot cycle index<br>– 0xFF-Unknown |
| *pSCM[OUT]* | • Station class mark<br>– 0xFF-Unknown |
| *pRegHomeSID[OUT]* | • Register on home SID<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *pRegForeignSID[OUT]* | • Register on foreign SID<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *pRegForeignNID[OUT]* | • Register on foreign NID<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *pForceRev0[OUT]* | • Force CDMA 1x-EV-DO Rev. 0 mode<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |

| | |
|---|---|
| *pCustomSCP[OUT]* | • Use a custom config for CDMA 1x-EV-DO SCP<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *pProtocol[OUT]* | • 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[OUT]* | • Broadcast mask for custom SCP config<br>– 0x00000001 - Generic broadcast enabled<br>– 0xFFFFFFFF - Unknown |
| *pApplication[OUT]* | • Application mask for custom SCP config<br>– 0x00000001 - SN Multiflow Packet Application<br>– 0x00000002 - SN Enhanced Multiflow Packet Application<br>– 0xFFFFFFFF - Unknown |
| *pRoaming[OUT]* | • Roaming preference<br>– 0 - Automatic<br>– 1 - Home Only<br>– 2 - Affiliated Roaming Only<br>– 3 - Home and Affiliated Roaming<br>– 0xFFFFFFFF - Unknown |

**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
Timeout: 5 seconds

#### 9.19.5.4 ULONG GetHomeNetwork ( WORD * *pMCC,* WORD * *pMNC,* BYTE *nameSize,* CHAR * *pName,* WORD * *pSID,* WORD * *pNID* )

Retrieves information about the home network of the device. For 3GPP2 home network information use GetHomeNetwork3GPP2.

**Parameters**

| | |
|---|---|
| *pMCC[OUT]* | • Mobile country code (UMTS only). |
| *pMNC[OUT]* | • Mobile network code (UMTS only). |
| *nameSize* | • Maximum number of characters (including NULL terminator) that 8 network name array can contain (UMTS only). |
| *pName[OUT]* | • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only). |
| *pSID[OUT]* | • Home network system ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000 |
| *pNID[OUT]* | • Home network ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000 |

**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

#### 9.19.5.5 ULONG GetHomeNetwork3GPP2 ( WORD * *pMCC,* WORD * *pMNC,* BYTE *nameSize,* CHAR * *pName,* WORD * *pSID,* WORD * *pNID,* WORD * *pNw2MCC,* WORD * *pNw2MNC,* BYTE * *pNw2DescDisp,* BYTE * *pNw2DescEnc,* BYTE * *pNw2DescLen,* BYTE * *pNw2Name* )

Retrieves information about the home network of the device. It will extract 3GPP2 Network Information also.

**Parameters**

| | |
|---|---|
| *pMCC[OUT]* | • Mobile country code (UMTS only). |
| *pMNC[OUT]* | • Mobile network code (UMTS only). |
| *nameSize* | • Maximum number of characters (including NULL terminator) that 8 network name array can contain (UMTS only). |
| *pName[OUT]* | • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only). |
| *pSID[OUT]* | • Home network system ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000 |
| *pNID[OUT]* | • Home network ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000 |
| *pNw2MCC[OUT]* | • Mobile country code (3GPP2 only).<br>• Range : 0 to 999 |
| *pNw2MNC[OUT]* | • Mobile network code (3GPP2 only).<br>• Range : 0 to 999 |
| *pNw2DescDisp[-OUT]* | • Network Name Display (3GPP2 only). -Valid Value<br>– 0x00 - Do not display<br>– 0x01 - Display<br>– 0xFF - Unknown |
| *pNw2DescDisp[-OUT]* | • Encoding of the network description (3GPP2 only).<br>• Valid Value<br>– 0x00 - Octet, unspecified<br>– 0x02 - 7-bit ASCII<br>– 0x04 - Unicode<br>– 0x09 - GSM 7-bit default |
| *nw2DescLen[OUT]* | • Network Description Length (3GPP2 only). |
| *pNw2Name[OUT]* | • Network Name (3GPP2 only). |

**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

#### 9.19.5.6 ULONG GetNetworkPreference ( ULONG * *pTechnologyPref,* ULONG * *pDuration,* ULONG * *pPersistentTechnologyPref* )

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 API SLQSGetSysSelectionPref() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pTechnology-Pref[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 |
| *pDuration[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 |
| *pPersistent-TechnologyPref[-OUT]* | • Bit field representing persistent radio technology preference<br>– Same representation as the pTechnologyPref parameter |

**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

#### 9.19.5.7 ULONG GetRFInfo ( BYTE * *pInstanceSize,* struct RFBandInfoElements * *pRFBandInfo* )

Sets the RFInfoList

**Parameters**

| | |
|---|---|
| *pInstanceSize[IN/OUT]* | • 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. |
| *pInstances[OUT]* | • RF info instances array<br>– See RFBandInfoElements 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

#### 9.19.5.8 ULONG GetServingNetwork ( ULONG * *pRegistrationState,* ULONG * *pCSDomain,* ULONG * *pPSDomain,* ULONG * *pRAN,* BYTE * *pRadioIfacesSize,* BYTE * *pRadioIfaces,* ULONG * *pRoaming,* WORD * *pMCC,* WORD * *pMNC,* BYTE *nameSize,* CHAR * *pName* )

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 SLQSNasGetSysInfo() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pRegistrationState[OUT]* | • Registration state:<br>– 0 - Not registered<br>– 1 - Registered<br>– 2 - Searching/Not Registered<br>– 3 - Registration Denied<br>– 4 - Unknown |
| *pCSDomain[OUT]* | • Circuit switch domain status:<br>– 0 - Unknown/Not Applicable<br>– 1 - Attached<br>– 2 - Detached |

| | |
|---|---|
| *pPSDomain[OUT]* | • Packet switch domain status<br>– 0 - Unknown/Not Applicable<br>– 1 - Attached<br>– 2 - Detached |
| *pRAN[OUT]* | • Type of radio access network on which mobile is registered:<br>– 0 - Unknown<br>– 1 - cdma2000 network<br>– 2 - UMTS network |
| *pRadioIfacesSize[IN/OUT]* | • 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. |
| *pRadioIfaces[OUT]* | • An array of Radio Interface Technology<br>– See qaGobiApiTableRadioInterfaces.h for the Radio Interface Technologies |
| *pRoaming[OUT]* | • Roaming indicator |
| *pMCC[OUT]* | • Mobile country code |
| *pMNC[OUT]* | • Mobile network code |
| *nameSize* | • Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks |
| *pName[OUT]* | • Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks |

**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

#### 9.19.5.9 ULONG GetServingNetworkCapabilities ( BYTE * *pDataCapsSize*, BYTE * *pDataCaps* )

Returns information regarding the data capabilities of the system that currently provides service to the device.

**Parameters**

| | |
|---|---|
| *pDataCapsSize[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. |
| *pDataCaps[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 |

**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

#### 9.19.5.10 ULONG GetSignalStrengths ( ULONG * *pArraySizes*, INT8 * *pSignalStrength*, ULONG * *pRadioInterface* )

Returns the available signal strengths ( in dBm ) as measured by the device in an array. The API also provides the corresponding radio 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 API SLQSNasGetSigInfo() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pArraySizes[IN/OUT]* | • Upon input maximum number of elements that each array can contain.<br>• Upon successful output actual number of elements in the array. |
| *pSignalStrength[OUT]* | • Received signal strength array (in dBm) |
| *pRadioInterface[OUT]* | • Radio interface technology array of the signal being measured<br>– See qaGobiApiTableRadioInterfaces.h for Radio Interface info |

**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

#### 9.19.5.11 ULONG InitiateDomainAttach ( ULONG *action* )

Initiates a domain attach/detach of 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 SLQSSetSysSelectionPref() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *action[IN]* | • Domain action to attempt<br>1 - Attach<br>2 - Detach |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 30 seconds

#### 9.19.5.12 ULONG InitiateNetworkRegistration ( ULONG *regType,* WORD *mcc,* WORD *mnc,* ULONG *rat* )

Initiates the network registration process. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSSetSysSelectionPref() and SLQSSetBandPreference() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *regType* | • Registration type<br>– 1 - Automatic<br>– 2 - Manual |
| *mcc* | • Mobile country code |

| | |
|---|---|
| *mnc* | • Mobile network code |
| *rat* | • Radio access technology<br>– 4 - GSM<br>– 5 - UMTS |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 30 seconds

#### 9.19.5.13 ULONG PerformNetworkScan ( BYTE * *pInstanceSize,* BYTE * *pInstances* )

Performs scan for available networks.

**Parameters**

| | |
|---|---|
| *pInstanceSize[IN/OUT]* | • 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. |
| *pInstances[OUT]* | • Network info instance array<br>– See QmiNas3GppNetworkInfo |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 5 minutes

#### 9.19.5.14 ULONG SetACCOLC ( CHAR * *spc,* BYTE *accolc* )

Sets the access overload class (ACCOLC)

**Parameters**

| | |
|---|---|
| *spc[IN]* | • service programming code NULL-terminated string of six digit |
| *ACCOLC[IN]* | • ACCOLC : Valid range is 0 to 15 |

**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
Timeout: 5 seconds

### 9.19.5.15 ULONG SetCDMANetworkParameters ( CHAR * *pSPC,* BYTE * *pForceRev0,* BYTE * *pCustomSCP,* ULONG * *pProtocol,* ULONG * *pBroadcast,* ULONG * *pApplication,* ULONG * *pRoaming* )

Sets the CDMA network parameters. Currently the modified settings will not be utilized until the device has been reset. For this reason, the recommended approach when using SetCDMANetworkParameters is for the application to perform the following steps:

1 - Call SetCDMANetworkParameters() 2 - Call SetPower( 5 ) 3 - Call QCWWANDisconnect() 4 - Reconnect after the device power cycles

**Parameters**

| | |
|---|---|
| *pSPC[IN]* | • Six digit service programming code (not necessary when only the roaming field is being set) |
| *pForceRev0[IN]* | • (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[IN]* | • (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[IN]* | • 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[IN]* | • Broadcast mask for custom SCP config<br>– 0x00000001 - Generic broadcast enabled<br>– 0xFFFFFFFF - Unknown |
| *pApplication[IN]* | • Application mask for custom SCP config<br>– 0x00000001 - SN Multiflow Packet Application<br>– 0x00000002 - SN Enhanced Multiflow Packet Application<br>– 0xFFFFFFFF - Unknown |
| *pRoaming[IN]* | • Roaming preference<br>– 0 - Automatic<br>– 1 - Home Only<br>– 2 - Affiliated Roaming Only<br>– 3 - Home and Affiliated Roaming<br>– 0xFFFFFFFF - Unknown |

**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
Timeout: 5 seconds

#### 9.19.5.16 ULONG SetNetworkPreference ( ULONG *technologyPref,* ULONG *duration* )

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 API SLQSSetSysSelectionPref() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *technologyPref[IN]* | • 2 Byte Bitmask representing radio technology preference<br>– No bits set indicates device to automatically determine the technology to use.<br>– Type of technology<br>* Bit 0 - Technology is 3GPP2<br>* Bit 1 - Technology is 3GPP<br>– Technology-specific protocol bitmask<br>* Bit 2 - Analog<br>· AMPS if 3GPP2, GSM if 3GPP<br>* Bit 3 - Digital<br>· CDMA if 3GPP2, WCDMA if 3GPP<br>* Bit 4 - HDR<br>* Bit 5 - LTE<br>* All other bits are reserved. |
| *duration[IN]* | • Duration of active preference<br>– 0 - Persistent<br>– 1 - Power cycle |

**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

#### 9.19.5.17 ULONG SLQSConfigSigInfo ( sigInfo * *pSigInfo* )

Sets the signal reporting thresholds. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use SLQSNasConfigSigInfo2() instead for new firmware versions and new modules.

**Parameters**

| | |
|---|---|
| *pSigInfo[IN]* | • See sigInfo 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: 5 seconds

#### 9.19.5.18 ULONG SLQSGetErrorRate ( GetErrRateResp * *pGetErrRateResp* )

This API retrieves current error rate information

**Parameters**

| | |
|---|---|
| *pGetErrRate-Resp[OUT]* | • See GetErrRateResp 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: 5 seconds

#### 9.19.5.19 ULONG SLQSGetNetworkTime ( GetNetworkTimeResp * *pGetNetworkTimeResp* )

This API retrieves the last known network time information from the UE.

**Parameters**

| | |
|---|---|
| *pGetNetwork-TimeResp[OUT]* | • See GetNetworkTimeResp 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: 5 seconds

#### 9.19.5.20 ULONG SLQSGetOperatorNameData ( nasOperatorNameResp * *pOperatorNameData* )

Get the operator name data from the network. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSGetPLMNName() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pOperatorNameData[OUT]* | • See nasOperatorNameResp 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: 10 seconds

#### 9.19.5.21 ULONG SLQSGetPLMNName ( nasPLMNNameReq * *pPLMNNameReq*, nasPLMNNameResp * *pPLMNNameResp* )

Get the operator name data from the network.

**Parameters**

| | |
|---|---|
| *pPLMNNameReq[IN]* | • See nasPLMNNameReq for more information |
| *pPLMNNameResp[OUT]* | • See nasPLMNNameResp 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: 10 seconds

#### 9.19.5.22 ULONG SLQSGetServingSystem ( qaQmiServingSystemParam * *pServingSystem* )

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 SLQSNasGetSysInfo() for new firmware versions and new modules. Also report available radio interface technology. If there are more than one radio interface, please choose the right interface(usually the first pair).

- See SLQSSetSysSelectionPref

**Parameters**

| | |
|---|---|
| *pServing-System[OUT]* | • serving system parameters obtained from the system |

**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

#### 9.19.5.23 ULONG SLQSGetSignalStrength ( struct slqsSignalStrengthInfo * *pSignalInfo* )

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 API SLQSNasGetSigInfo() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *pSignalInfo[IN/-OUT]* | • See slqsSignalStrengthInfo 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

#### 9.19.5.24 ULONG SLQSGetSysSelectionPref ( sysSelectPrefInfo * *pSysSelectPrefInfo* )

Queries the different system selection preferences of the device.

**Parameters**

| | |
|---|---|
| *pSysSelectPref-Info[OUT]* | • See sysSelectPrefInfo 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

#### 9.19.5.25 ULONG SLQSInitiateNetworkRegistration ( nasInitNetworkReg * *pNasInitNetRegistrationReg* )

Initiates the network registration process.

**Parameters**

| | |
|---|---|
| *pNasInitNet-Registration-Req[IN]* | • Pointer to structure nasInitNetworkReq<br>– See nasInitNetworkReg 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: UMTS
Timeout: 30 seconds

#### 9.19.5.26 ULONG SLQSNasConfigSigInfo2 ( setSignalStrengthInfo * *pSetSignalStrengthInfo* )

Sets the signal strength reporting thresholds

**Parameters**

| | |
|---|---|
| *pSetSignal-StrengthInfo[IN]* | • See setSignalStrengthInfo 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: 5 seconds
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.

### 9.19.5.27 ULONG SLQSNasGet3GPP2Subscription ( nasGet3GPP2SubscriptionInfoReq * *pGet3GPP2SubsInfoReq*, nasGet3GPP2SubscriptionInfoResp * *pGet3GPP2SubsInfoResp* )

This API retrieves 3GPP2 subscription-related information.

**Parameters**

| | |
|---|---|
| *pGet3GPP2-SubsInfoReq[IN]* | • See nasGet3GPP2SubscriptionInfoReq for more information |
| *pGet3GPP2-SubsInfoResp[-OUT]* | • See nasGet3GPP2SubscriptionInfoResp 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
Timeout: 2 seconds
This command retrieves 3GPP2 subscription-related information. The QMI_ERR_INTERNAL error is returned when no information can be retrieved from the modem.

### 9.19.5.28 ULONG SLQSNasGetCellLocationInfo ( nasCellLocationInfoResp * *pNasCellLocationInfoResp* )

This API retrieves cell location-related information

**Parameters**

| | |
|---|---|
| *pNasCell-LocationInfo-Resp[OUT]* | • See nasCellLocationInfoResp 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
This API retrieves cell location-related information, depending on current serving system.

#### 9.19.5.29 ULONG SLQSNASGeteDRXParams ( nasGeteDRXParamsResp * *pNasGeteDRXParamsResp* )

This API Retrieves the eDRX (extended discontinuous reception) parameters for the device.

**Parameters**

| | |
|---|---|
| *pNasGeteDRX-ParamsResp[IN]* | • See nasGeteDRXParamsResp for more information. |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

**Note**

Timeout: 5 seconds

#### 9.19.5.30 ULONG SLQSNasGetHDRColorCode ( nasGetHDRColorCodeResp * *pGetHDRColorCodeResp* )

This API retrieves the current HDR color code value.

**Parameters**

| | |
|---|---|
| *pGetHDRColor-CodeResp[OUT]* | • See nasGetHDRColorCodeResp 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

#### 9.19.5.31 ULONG SLQSNASGetLTECPHYCaInfo ( nasGetLTECphyCa * *pLTECPhyCa* )

This API Get LTE CPHY Carrier Info

**Parameters**

| | |
|---|---|
| *pLTECPhyCa[IN]* | • See nasGetLTECphyCa for more information. |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

#### 9.19.5.32 ULONG SLQSNasGetRFInfo ( NasGetRFInfoResp * *pGetRFInfoResp* )

Gets radio band/channel information regarding the system currently providing service.

**Parameters**

| | |
|---|---|
| *pGetRFInfoResp[IN/OUT]* | • See NasGetRFInfoResp 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

#### 9.19.5.33 ULONG SLQSNasGetSigInfo ( nasGetSigInfoResp * *pGetSigInfoResp* )

This API queries information regarding the signal strength.

**Parameters**

| | |
|---|---|
| *pGetSigInfoResp[OUT]* | • See nasGetSigInfoResp 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
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

#### 9.19.5.34 ULONG SLQSNasGetSysInfo ( nasGetSysInfoResp * *pGetSysInfoResp* )

Provides the system information. This API is preferred when trying to get the service status info and serving system info. The API SLQSGetServingSystem() reports similar NAS information, but it is deprecated. Please refer to the header description of API SLQSGetServingSystem() for more information.

**Parameters**

| | |
|---|---|
| *pGetSysInfo-Resp[OUT]* | • See nasGetSysInfoResp 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
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.

#### 9.19.5.35 ULONG SLQSNasGetTxRxInfo ( nasGetTxRxInfoReq * *pGetTxRxInfoReq,* nasGetTxRxInfoResp * *pGetTxRxInfoResp* )

This API retrieves the detailed Tx/Rx information.

**Parameters**

| | |
|---|---|
| *pGetTxRxInfo-Req[IN]* | • See nasGetTxRxInfoReq for more information |
| *pGetTxRxInfo-Resp[OUT]* | • See nasGetTxRxInfoResp 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
This command retrieves Tx/Rx information for a radio interface. The Rx chain are included in the response message only if they are enabled.

#### 9.19.5.36 ULONG SLQSNasIndicationRegister ( BYTE *systemSelectionInd,* BYTE *DDTMInd,* BYTE *servingSystemInd* )

Register/De-register from NAS (Network access service) broadcast indications. Some callbacks would not be invoked if the indications are not registered. The details are provided in the parameter description.

**Parameters**

| | |
|---|---|
| *systemSelectionInd[IN]* | • system selection preference indication registration. The following callbacks would not be invoked if the indication is disabled.<br>tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem<br>– 0x00 - for disable<br>– 0x01 - for enable<br>– 0xFF - No change - Specifying this parameter indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device |
| *DDTMInd[IN]* | • DDTM (Data Dedicated Transmission Mode) indication registration.<br>– 0x00 - for disable<br>– 0x01 - for enable<br>– 0xFF - No change - Specifying this parameter indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device |
| *servingSystemInd[IN]* | • Serving system indication registration. The following callbacks would not be invoked if the indication is disabled.<br>tFNBandPreference<br>– 0x00 - for disable<br>– 0x01 - for enable<br>– 0xFF - No change - Specifying this parameter indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

**Note**

Timeout: 2 seconds

#### 9.19.5.37 ULONG SLQSNasIndicationRegisterExt ( nasIndicationRegisterReq * *pIndicationRegisterReq* )

This API Registers/De-registers for different NAS (Network access service) indications.

**Parameters**

| | |
|---|---|
| *pIndication-RegisterReq[IN]* | • See nasIndicationRegisterReq for more information |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

**Note**

Timeout: 2 seconds
This API 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.

#### 9.19.5.38 ULONG SLQSNasIndicationRegisterLTECphyCa ( BYTE * *bStatus* )

This API Registers/De-registers for NAS CPHY Carrier Info.

**Parameters**

| | |
|---|---|
| *bStatus[IN]* | • Values<br>– 0 - De-register.<br>– 1 - Register. |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

#### 9.19.5.39 ULONG SLQSNASSeteDRXParams ( nasSeteDRXParamsReq * *pNasSeteDRXParamsReq* )

This API sets the eDRX (extended discontinuous reception) parameters for the device.

**Parameters**

| | |
|---|---|
| *pNasSeteDRX-ParamsReq[IN]* | • See nasSeteDRXParamsReq for more information. |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

**Note**

Timeout: 5 seconds

#### 9.19.5.40 ULONG SLQSNASSwiGetChannelLock ( nasSwiGetChannelLockResp * *pNasSwiGetChannelLockResp* )

This API queries the channel or cell which the UE is locked into.

**Parameters**

| | |
|---|---|
| *pNasSwiGetChannelLock[OUT]* | • See nasSwiGetChannelLockResp for more information. |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

#### 9.19.5.41 ULONG SLQSNasSwiIndicationRegister ( NasSwiIndReg * *pIndRegReq* )

sets the registration state for different QMI_NAS SWI indications

**Parameters**

| | |
|---|---|
| *pIndRegReq[IN]* | • See NasSwiIndReg 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

#### 9.19.5.42 ULONG SLQSNasSwiModemStatus ( swiModemStatusResp * *pModemStatusResp* )

This API requests the device to return the current status of modem.

**Parameters**

| | |
|---|---|
| *pModemStatus-Resp[OUT]* | • See swiModemStatusResp 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

#### 9.19.5.43 ULONG SLQSNASSwiSetChannelLock ( nasSwiSetChannelLockReq * *pNasSwiSetChannelLockReq* )

This API allows the host to lock the UE to a specific channel or cell.

**Parameters**

| | |
|---|---|
| *pNasSwiSet-ChannelLock-Req[IN]* | • See nasSwiSetChannelLockReq for more information. |

**Returns**

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values.

**Note**

The settings are persistent across reboots.

#### 9.19.5.44 ULONG SLQSPerformNetworkScan ( slqsNetworkScanInfo * *pNetworkInfo* )

Performs scan for available networks and scans for RAT info as well.

**Parameters**

| | |
|---|---|
| *pNetworkInfo[I-N/OUT]* | • See slqsNetworkScanInfo for more information<br>• Valid pointers to the following structure members are mandatory<br>– pNetworkInfoInstances<br>– pNetworkInfo |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 5 minutes

#### 9.19.5.45 ULONG SLQSSetBandPreference ( ULONGLONG *bandpreference* )

Provides information about the band preference.

**Parameters**

| | |
|---|---|
| *bandpreference[-IN]* | • 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 |

**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

#### 9.19.5.46 ULONG SLQSSetSysSelectionPref ( sysSelectPrefParams * *pSysSelectPrefParams* )

Sets the different system selection preferences of the device.

**Parameters**

| | |
|---|---|
| *pSysSelectPrefParams[IN]* | • See sysSelectPrefParams 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

#### 9.19.5.47 ULONG SLQSSetSysSelectionPrefExt ( sysSelectPrefParamsExt * *pSysSelectPrefParamsExt* )

Sets the different system selection preferences of the device.

**Parameters**

| | |
|---|---|
| *pSysSelectPrefParamsExt[IN]* | • See sysSelectPrefParamsExt 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

#### 9.19.5.48 ULONG SLQSSwiGetHDRPersonality ( HDRPersonalityResp * *pHDRPersonalityResp* )

This API retrieves HDR Personality related information

**Parameters**

| | |
|---|---|
| *pHDR-Personality-Resp[OUT]* | • See HDRPersonalityResp 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
Timeout: 5 seconds

#### 9.19.5.49 ULONG SLQSSwiGetHDRProtSubtype ( HDRProtSubtypResp * *pHDRProtSubtypResp* )

This API retrieves HDR Prototype Subtype related information

**Parameters**

| | |
|---|---|
| *pHDRProt-SubtypResp[OU-T]* | • See HDRProtSubtypResp 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
Timeout: 5 seconds

#### 9.19.5.50 ULONG SLQSSwiGetHRPDStats ( GetHRPDStatsResp * *pGetHRPDStatsResp* )

This API retrieves currently acquired HRPD system statistics

**Parameters**

| | |
|---|---|
| *pGetHRPD-StatsResp[OUT]* | • See GetHRPDStatsResp 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: 5 seconds

#### 9.19.5.51 ULONG SLQSSwiGetLteCQI ( LteCQIParm * *pLteCQIResp* )

This API Fetch CQI parameters for LTE data session

**Parameters**

| | |
|---|---|
| *pLteCQIParm[OUT]* | • See LteCQIParm 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: 5 seconds

#### 9.19.5.52 ULONG SLQSSwiGetLteSccRxInfo ( LteSccRxInfoResp * *pLteSccRxInfoResp* )

This API retrieves the LTE Secondary carrier Rx signal level information.

**Parameters**

| | |
|---|---|
| *pGetLteSccRxInforesp[OUT]* | • See GetLteSccRxInforesp 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: 5 seconds

#### 9.19.5.53 ULONG SLQSSwiNetworkDebug ( NetworkDebugResp * *pNetworkDebugResp* )

This API retrieves device and network status details

**Parameters**

| | |
|---|---|
| *pNetworkDebug-Resp[OUT]* | • See NetworkDebugResp 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: 5 seconds

#### 9.19.5.54 ULONG SLQSSwiPSDetach ( PSDetachReq * *pPSDetachReq* )

This API detaches PS connection.

**Parameters**

| | |
|---|---|
| *pPSDetachReq[-IN]* | • See PSDetachReq 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
Timeout: 5 seconds

## 9.20 qaGobiApiOmadm.h File Reference

Open Mobile Alliance Device Management Service API function prototypes.

### Functions

- ULONG OMADMStartSession (ULONG sessionType)
- ULONG OMADMCancelSession ()

- ULONG OMADMGetSessionInfo (ULONG *pSessionState, ULONG *pSessionType, ULONG *pFailureReason, BYTE *pRetryCount, WORD *pSessionPause, WORD *pTimeRemaining)
- ULONG OMADMGetPendingNIA (ULONG *pSessionType, USHORT *pSessionID)

### 9.20.1 Detailed Description

Open Mobile Alliance Device Management Service API function prototypes.

### 9.20.2 Function Documentation

#### 9.20.2.1 ULONG OMADMCancelSession ( )

Cancels an ongoing OMA-DM session.

**Parameters**

| | |
|---|---|
| *None* | |

**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
Timeout: 2 seconds

#### 9.20.2.2 ULONG OMADMGetPendingNIA ( ULONG * *pSessionType,* USHORT * *pSessionID* )

Returns information about the pending network-initiated alert

**Parameters**

| | |
|---|---|
| *SessionType[OUT]* | • Session Type<br>– 0x04 - Network-initiated PRL update<br>– 0x05 - Network-initiated device configure |
| *SessionID[OUT]* | • Session Id<br>– Unique session ID for NIA request |

**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
Timeout: 2 seconds

#### 9.20.2.3 ULONG OMADMGetSessionInfo ( ULONG * *pSessionState,* ULONG * *pSessionType,* ULONG * *pFailureReason,* BYTE * *pRetryCount,* WORD * *pSessionPause,* WORD * *pTimeRemaining* )

Returns information related to the current (or previous if no session is active) OMA-DM session.

**Parameters**

| | |
|---|---|
| *SessionState[OUT]* | • Session state<br>– 0x00 - Complete, information was updated<br>– 0x01 - Complete, update information is unavailable<br>– 0x02 - Failed<br>– 0x03 - Retrying<br>– 0x04 - Connecting<br>– 0x05 - Connected<br>– 0x06 - Authenticated<br>– 0x07 - Mobile Directory Number (MDN) downloaded<br>– 0x08 - Mobile Station Identifier (MSID) downloaded<br>– 0x09 - PRL downloaded<br>– 0x0A - Mobile IP (MIP) profile downloaded |
| *sessionType[OUT]* | • Session State<br>– 0x00 - Client-initiated device configure<br>– 0x01 - Client-initiated PRL update<br>– 0x02 - Client-initiated hands-free activation<br>– 0x03 - Device-initiated hands-free activation<br>– 0x04 - Network-initiated PRL update<br>– 0x05 - Network-initiated device configure |
| *FailureReason[OUT]* | • Session failure reason (when state indicates failure)<br>– 0x00 - Unknown<br>– 0x01 - Network is unavailable<br>– 0x02 - Server is unavailable<br>– 0x03 - Authentication failed<br>– 0x04 - Maximum retry exceeded<br>– 0x05 - Session is cancelled |
| *RetryCount[OUT]* | • Session retry count (when state indicates retrying) |
| *SessionPause[OUT]* | • Session pause timer (in seconds , when state indicates retrying) |

| | |
|---|---|
| *Time-Remaining[OUT]* | • Pause time remaining (in seconds , when state indicates retrying) |

**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
Timeout: 2 seconds

#### 9.20.2.4 ULONG OMADMStartSession ( ULONG *sessionType* )

Starts an OMA-DM session.

**Parameters**

| | |
|---|---|
| *sessionType* | • Session type<br>– 0x00 - Client-initiated device configure<br>– 0x01 - Client-initiated PRL update<br>– 0x02 - Client-initiated hands-free activation |

**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
Timeout: 2 seconds

## 9.21 qaGobiApiPds.h File Reference

Position Determination Service API function prototypes.

### Data Structures

- struct PDSPositionData
- struct GPSStateInfo
- struct PDSPosMethodStateReq

### Macros

- #define DEFAULTBYTEVALUE 0xFF
- #define DEFAULTWORDVALUE 0xFFFF
- #define DEFAULTLONGVALUE 0xFFFFFFFF

### Enumerations

- enum {
  eSetServiceAutomaticTrackingDisable =0,
  eSetServiceAutomaticTrackingEnable =1 }

### Functions

- ULONG GetPDSState (ULONG *pEnabledStatus, ULONG *pTrackingStatus)
- ULONG SetPDSState (ULONG enable)
- ULONG StartPDSTrackingSessionExt (BYTE sessionControl, BYTE sessionType, BYTE sessionOperation, BYTE sessionServerOption, BYTE fixTimeout, ULONG fixInterval, ULONG fixCount, ULONG fixAccuracy)
- ULONG StopPDSTrackingSession ()
- ULONG PDSInjectTimeReference (ULONGLONG systemTime, USHORT systemDiscontinuities)
- ULONG GetPDSDefaults (ULONG *pOperation, BYTE *pTimeout, ULONG *pInterval, ULONG *pAccuracy)
- ULONG SetPDSDefaults (ULONG operation, BYTE timeout, ULONG interval, ULONG accuracy)
- ULONG GetXTRAAutomaticDownload (ULONG *pbEnabled, USHORT *pInterval)
- ULONG SetXTRAAutomaticDownload (ULONG bEnabled, USHORT interval)
- ULONG GetXTRANetwork (ULONG *pPreference)
- ULONG SetXTRANetwork (ULONG preference)
- ULONG GetXTRAValidity (USHORT *pGPSWeek, USHORT *pGPSWeekOffset, USHORT *pDuration)
- ULONG ForceXTRADownload ()
- ULONG GetServiceAutomaticTracking (ULONG *pbAuto)
- ULONG SetServiceAutomaticTracking (ULONG bAuto)
- ULONG GetPortAutomaticTracking (ULONG *pbAuto)
- ULONG SetPortAutomaticTracking (ULONG bAuto)
- ULONG ResetPDSData (ULONG *pGPSDataMask, ULONG *pCellDataMask)
- ULONG SLQSSetAGPSConfig (ULONG *pServerAddress, ULONG *pServerPort, BYTE *pServerURL, BYTE *pServerURLLength, BYTE *pNetworkMode)
- ULONG SLQSPDSInjectAbsoluteTimeReference (ULONGLONG timeMsec, ULONG timeUncMsec, BYTE timeBase, BYTE forceFlag)
- ULONG SLQSGetAGPSConfig (ULONG *pServerAddress, ULONG *pServerPort, BYTE *pServerURL, BYTE *pServerURLLength, BYTE *pNetworkMode)
- ULONG SLQSPDSInjectPositionData (struct PDSPositionData *pPositionData)
- ULONG SLQSPDSDeterminePosition ()
- ULONG SLQSGetGPSStateInfo (GPSStateInfo *pGPSStateInfo)
- ULONG SLQSSetPositionMethodState (PDSPosMethodStateReq *pPDSPosMethodStateReq)

#### 9.21.1 Detailed Description

Position Determination Service API function prototypes.

### 9.21.2 Macro Definition Documentation

#### 9.21.2.1 #define DEFAULTBYTEVALUE 0xFF

#### 9.21.2.2 #define DEFAULTLONGVALUE 0xFFFFFFFF

#### 9.21.2.3 #define DEFAULTWORDVALUE 0xFFFF

### 9.21.3 Enumeration Type Documentation

#### 9.21.3.1 anonymous enum

**Enumerator**

***eSetServiceAutomaticTrackingDisable***

***eSetServiceAutomaticTrackingEnable***

### 9.21.4 Function Documentation

#### 9.21.4.1 ULONG ForceXTRADownload ( )

Forces the XTRA database to be downloaded to the device.

**Parameters**

| | |
|---|---|
| *none* | |

**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

#### 9.21.4.2 ULONG GetPDSDefaults ( ULONG * *pOperation,* BYTE * *pTimeout,* ULONG * *pInterval,* ULONG * *pAccuracy* )

Returns the default tracking session configuration. The tracking session configuration is used when a tracking session is automatically started using SetServiceAutomaticTracking or due to the device detecting an application opening the NMEA port.

**Parameters**

| | |
|---|---|
| *pOperation[OUT]* | • Current session operating mode<br>– 0 - Standalone<br>– 1 - MS based<br>– 2 - MS assisted |

| | |
|---|---|
| *pTimeout[OUT]* | • Maximum amount of time (seconds) to work on each fix, maximum is 255 |
| *pInterval[OUT]* | • Interval (seconds) between fix requests |
| *pAccuracy[OUT]* | • Preferred accuracy threshold (meters) |

**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

#### 9.21.4.3 ULONG GetPDSState ( ULONG * *pEnabledStatus*, ULONG * *pTrackingStatus* )

Returns the current PDS state.

**Parameters**

| | |
|---|---|
| *pEnabled-Status[OUT]* | • Current PDS state<br>– 0 - disable<br>– 1 - enable |
| *pTracking-Status[OUT]* | • Current PDS tracking session state<br>• Values:<br>– 0x00 - Unknown<br>– 0x01 - Inactive<br>– 0x02 - Active |

**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

#### 9.21.4.4 ULONG GetPortAutomaticTracking ( ULONG * *pbAuto* )

Returns the automatic tracking configuration for the NMEA COM port.

**Parameters**

| | |
|---|---|
| *pbAuto[OUT]* | • Automatic tracking enabled for NMEA COM port<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

**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

#### 9.21.4.5 ULONG GetServiceAutomaticTracking ( ULONG * *pbAuto* )

Returns the automatic tracking state for the service.

**Parameters**

| | |
|---|---|
| *pbAuto[OUT]* | • Automatic tracking session started for service<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

**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

#### 9.21.4.6 ULONG GetXTRAAutomaticDownload ( ULONG * *pbEnabled,* USHORT * *pInterval* )

Returns the XTRA automatic database download configuration.

**Parameters**

| | |
|---|---|
| *pbEnabled[OUT]* | • Automatic XTRA download status<br>– 0 - Disabled<br>– 1 - Enabled |
| *pInterval[OUT]* | • Interval (hours) between XTRA downloads |

**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

#### 9.21.4.7 ULONG GetXTRANetwork ( ULONG * *pPreference* )

Returns the XTRA WWAN network preference. When automatic XTRA database downloading is enabled this preference determines which WWAN networks will be used to perform the XTRA database download.

**Parameters**

| | |
|---|---|
| *pPreference[OUT]* | • 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 |

**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

#### 9.21.4.8 ULONG GetXTRAValidity ( USHORT * *pGPSWeek*, USHORT * *pGPSWeekOffset*, USHORT * *pDuration* )

Returns the XTRA database validity period. When automatic XTRA database downloading is enabled the validity period determines when the XTRA database will be updated through a new download.

**Parameters**

| | |
|---|---|
| *pGPSWeek[OUT]* | • Starting GPS week of validity period |
| *pGPSWeekOffset[OUT]* | • Starting GPS week offset (minutes) of validity period |
| *pDuration[OUT]* | • Length of validity period (hours) |

**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

#### 9.21.4.9 ULONG PDSInjectTimeReference ( ULONGLONG *systemTime,* USHORT *systemDiscontinuities* )

Injects a system time into the PDS engine.

**Parameters**

| | |
|---|---|
| *systemTime* | • System time( milliseconds ) |
| *systemDiscontinuities* | • Number of system time discontinuities |

**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

#### 9.21.4.10 ULONG ResetPDSData ( ULONG * *pGPSDataMask,* ULONG * *pCellDataMask* )

Resets the specified PDS data.

**Parameters**

| | |
|---|---|
| *pGPSData-Mask[IN]* | • 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[-IN]* | • 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 |

**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

#### 9.21.4.11 ULONG SetPDSDefaults ( ULONG *operation,* BYTE *timeout,* ULONG *interval,* ULONG *accuracy* )

Sets the default tracking session configuration. The tracking session configuration is used when a tracking session is automatically started using SetServiceAutomaticTracking or due to the device detecting an application opening the NMEA port.

**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) |

**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

#### 9.21.4.12 ULONG SetPDSState ( ULONG *enable* )

Sets the PDS state.

**Parameters**

| | |
|---|---|
| *enable[IN]* | • Desired PDS state<br>– Zero - disable<br>– Non-Zero - enable |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 Seconds

#### 9.21.4.13 ULONG SetPortAutomaticTracking ( ULONG *bAuto* )

Sets the automatic tracking configuration for the NMEA COM port.

**Parameters**

| | |
|---|---|
| *bAuto[IN]* | • Enable automatic tracking for NMEA COM port<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

**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

#### 9.21.4.14 ULONG SetServiceAutomaticTracking ( ULONG *bAuto* )

Sets the automatic tracking state for the service. Tracking session being started using the default session configuration. Auto-tracking continues to generate fixes indefinitely until requested to be disabled. In StartPDSTracking-SessionExt a tracking session get started using the specified session control method and input parameters. After completion of requested no. of position fixes or service times out to perform fix, tracking session ends and GPS service deactivates.

**Parameters**

| | |
|---|---|
| *bAuto[IN]* | • Automatic tracking session started for service<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

**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

#### 9.21.4.15 ULONG SetXTRAAutomaticDownload ( ULONG *bEnabled,* USHORT *interval* )

Sets the XTRA automatic database download configuration.

**Parameters**

| | |
|---|---|
| *bEnabled[IN]* | • Automatic XTRA download status<br>– 0 - Disabled<br>– 1 - Enabled |
| *interval[IN]* | • Interval (hours) between XTRA downloads |

**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

#### 9.21.4.16 ULONG SetXTRANetwork ( ULONG *preference* )

Sets the XTRA WWAN network preference. When automatic XTRA database downloading is enabled this preference determines which WWAN networks will be used to perform the XTRA database download.

**Parameters**

| | |
|---|---|
| *preference[IN]* | • 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 |

**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

#### 9.21.4.17 ULONG SLQSGetAGPSConfig ( ULONG * *pServerAddress*, ULONG * *pServerPort*, BYTE * *pServerURL*, BYTE * *pServerURLLength*, BYTE * *pNetworkMode* )

Gets the PDS AGPS (MS-based) configuration.

**Parameters**

| | |
|---|---|
| *pServer-Address[OUT]* | • IPv4 address of AGPS server. "0" if not set |
| *pServerPort[OU-T]* | • Port number of AGPS server. "0" if not set |
| *pServerURL[O-UT]* | • URL of the AGPS server. "0" if not set |
| *pServerURL-Length[OUT]* | • URL length of AGPS server. "0" if not set |
| *pNetworkMode[-IN]* | • Network Mode of AGPS Server [optional - should be present in Multimode Systems]<br>– 0x00 - UMTS<br>– 0x01 - CDMA |

**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

#### 9.21.4.18 ULONG SLQSGetGPSStateInfo ( GPSStateInfo * *pGPSStateInfo* )

Queries the MSM GPS server for receiver state information

**Parameters**

| | |
|---|---|
| *pGPSStateInfo[-OUT]* | • contains the GPS State Info<br>• See GPSStateInfo 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: 5 Seconds

#### 9.21.4.19 ULONG SLQSPDSDeterminePosition ( )

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions.

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 Seconds

#### 9.21.4.20 ULONG SLQSPDSInjectAbsoluteTimeReference ( ULONGLONG *timeMsec,* ULONG *timeUncMsec,* BYTE *timeBase,* BYTE *forceFlag* )

Injects a absolute time reference into the PDS engine.

**Parameters**

| | |
|---|---|
| *timeMsec[IN]* | • 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[IN]* | • Time uncertainty in milliseconds |
| *timeBase[IN]* | • Time base<br>– 0x00 - GPS (midnight, Jan 6, 1980)<br>– 0x01 - UTC (midnight, Jan 1, 1970) |
| *forceFlag[IN]* | • Force acceptance of data |

**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

#### 9.21.4.21 ULONG SLQSPDSInjectPositionData ( struct PDSPositionData * *pPositionData* )

Injects position data into the PDS engine.

**Parameters**

| | |
|---|---|
| *pPositionData[IN]* | • contains the position data to be injected to the PDS engine |

**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

#### 9.21.4.22 ULONG SLQSSetAGPSConfig ( ULONG * *pServerAddress*, ULONG * *pServerPort*, BYTE * *pServerURL*, BYTE * *pServerURLLength*, BYTE * *pNetworkMode* )

Sets the PDS AGPS (MS-based) configuration.

**Parameters**

| | |
|---|---|
| *pServerAddress[IN]* | • IPv4 address of AGPS server [optional] |
| *pServerPort[IN]* | • Port number of AGPS server [optional - should be present when pServerAddress is present] |
| *pServerURL[IN]* | • URL of the AGPS server [optional] |
| *pServerURLLength[IN]* | • URL length of AGPS server [optional - should be present when pServerURL is present] |
| *pNetworkMode[IN]* | • Network Mode of AGPS Server [optional - should be present in Multimode Systems]<br>– 0x00 - UMTS<br>– 0x01 - CDMA |

**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

#### 9.21.4.23 ULONG SLQSSetPositionMethodState ( PDSPosMethodStateReq * *pPDSPosMethodStateReq* )

Sets the state of positioning methods for the device.

**Parameters**

| | |
|---|---|
| *pPDSPosMethodStateReq[IN]* | • See PDSPosMethodStateReq 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: 5 Seconds

#### 9.21.4.24 ULONG StartPDSTrackingSessionExt ( BYTE *sessionControl*, BYTE *sessionType*, BYTE *sessionOperation*, BYTE *sessionServerOption*, BYTE *fixTimeout*, ULONG *fixInterval*, ULONG *fixCount*, ULONG *fixAccuracy* )

This function starts a PDS tracking session.

**Parameters**

| | |
|---|---|
| *sessionControl[IN]* | • Control method:<br>– 0x0 - Manual |
| *sessionType[IN]* | • Type:<br>– 0x0 - New |
| *sessionOperation[IN]* | • Operating mode:<br>– 0x00 - Standalone<br>– 0x01 - MS-based |
| *sessionServerOption[IN]* | • Location server option:<br>– 0x0 - Default |
| *fixTimeout[IN]* | • Maximum time to work on each fix (in seconds, max 255) |
| *fixCount[IN]* | • Count of position fix requests for this session (must be at least 1) |
| *fixInterval[IN]* | • interval between position fix requests (in seconds) |

| | |
|---|---|
| *fixAccuracy[IN]* | • Preferred accuracy threshold(in meters) |

**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

#### 9.21.4.25 ULONG StopPDSTrackingSession ( )

This function stops a PDS tracking session.

**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

## 9.22 qaGobiApiQos.h File Reference

Quality of Service API function prototypes.

### Data Structures

- struct dataRate
- struct tokenBucket
- struct pktErrRate
- struct swiQosFlow
- struct IPv4Addr
- struct Tos
- struct IPv6Addr
- struct IPv6TrafCls
- struct Port
- struct swiQosFilter
- struct swiQosReq
- struct swiQosIds
- struct swiQosModifyReq
- struct swiQosGranted
- struct NWProfile
- struct sQosFlowStat
- struct sQosStat
- struct sApnExtraParams

### Macros

- #define MAX_QOS_SPEC_PER_APN (10)
- #define MAX_QOS_FILTER_TLV 25

### Functions

- ULONG SLQSQosReset (BYTE instance)
- ULONG SLQSQosReq (BYTE instance, swiQosReq *pQosReq, swiQosIds *pQosResp)
- ULONG SLQSQosRel (BYTE instance, swiQosIds *pQosIds)
- ULONG SLQSQosSuspend (BYTE instance, swiQosIds *pQosIds)
- ULONG SLQSQosResume (BYTE instance, swiQosIds *pQosIds)
- ULONG SLQSQosModify (BYTE instance, swiQosModifyReq *pReq)
- ULONG SLQSQosGetGranted (BYTE instance, ULONG id, swiQosGranted *pGranted)
- ULONG SLQSQosGetFlowStatus (BYTE instance, ULONG id, BYTE *pStatus)
- ULONG SLQSQosGetNetworkStatus (BYTE instance, BYTE *pStatus)
- ULONG SLQSQosGetNWProf (BYTE instance, BYTE *pSz, NWProfile *pProfile)
- ULONG SLQSQosSwiReadDataStats (BYTE instance, ULONG apnId, sQosStat *pQosStat)
- ULONG SLQSQosSwiReadApnExtraParams (BYTE instance, ULONG apnId, sApnExtraParams *pApnExtraParams)
- ULONG SetQosMuxID (BYTE instance)

## 9.22.1 Detailed Description

Quality of Service API function prototypes.

## 9.22.2 Macro Definition Documentation

### 9.22.2.1 #define MAX_QOS_FILTER_TLV 25

### 9.22.2.2 #define MAX_QOS_SPEC_PER_APN (10)

## 9.22.3 Function Documentation

### 9.22.3.1 ULONG SetQosMuxID ( BYTE *instance* )

Bind MUX ID to Qos instance.

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.2 ULONG SLQSQosGetFlowStatus ( BYTE *instance,* ULONG *id,* BYTE * *pStatus* )

Get the status of a QoS flow.

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| in | *id[IN]* | Qos identifier Index identifying the QoS flow that has been negotiated |
| out | *pStatus[OUT]* | Qos status Current QoS instance status:<br>• 0x01 – QMI_QOS_STATUS_ACTIVATED<br>• 0x02 – QMI_QOS_STATUS_SUSPENDED<br>• 0x03 – QMI_QOS_STATUS_GONE |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.3 ULONG SLQSQosGetGranted ( BYTE *instance,* ULONG *id,* swiQosGranted * *pGranted* )

Retrieve the QoS parameters that are in effect for the specified QoS 16 flow as a result of network negotiation

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| in | *id[IN]* | • Qos identifier<br>• Index identifying the QoS flow that has been negotiated |
| in | *pGranted[OUT]* | • Tx/Rx Qos granted flow<br>• See swiQosGranted for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.4 ULONG SLQSQosGetNetworkStatus ( BYTE *instance,* BYTE * *pStatus* )

Queries whether the device is currently on a network that supports QoS

**Parameters**

| | | |
|---|---|---|
| | *instance[IN]* | • QMI instance |
| out | *pStatus[OUT]* | Network QoS support status<br>• 0 – No QoS support in network<br>• 1 – Network supports QoS |

**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

#### 9.22.3.5 ULONG SLQSQosGetNWProf ( BYTE *instance,* BYTE * *pSz,* NWProfile * *pProfile* )

Get network supported QoS profile information

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| | *in/out]* | pSz Number of network supported QoS profiles for one technology |
| out | *pProfile* | Network supported QoS profiles |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.6 ULONG SLQSQosModify ( BYTE *instance,* swiQosModifyReq * *pReq* )

Resume one or more existing QoS flows

**Warning**

NOT IMPLEMENTED

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| | *pReq[IN]* | • See swiQosModifyReq for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.7 ULONG SLQSQosRel ( BYTE *instance,* swiQosIds * *pQosIds* )

Release one or more existing QoS flows

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| | *pQosIds[IN]* | • See swiQosIds for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.8 ULONG SLQSQosReq ( BYTE *instance,* swiQosReq * *pQosReq,* swiQosIds * *pQosResp* )

Triggers QoS negotiation by providing QoS parameters

**Parameters**

| | |
|---|---|
| *instance[IN]* | • QMI instance |
| *pQoSReq[IN]* | • See swiQosReq for more information |
| *pQosResp[OUT]* | • See swiQosIds 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: UMTS/CDMA
Device Supported: MC77XX
Timeout: 2 seconds
At least one pair of optional TLVs must be present; Tx QoS Flow Request and Tx QoS Filter Request TLVs must both be present if either one is present; Rx QoS Flow Request and Rx QoS Filter Request TLVs must both be present if either one is present

#### 9.22.3.9 ULONG SLQSQosReset ( BYTE *instance* )

Reset the QoS service state variables of the requesting control point

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |

**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, UMTS & LTE
Device Supported: MC7750
Timeout: 2 seconds

#### 9.22.3.10 ULONG SLQSQosResume ( BYTE *instance,* swiQosIds * *pQosIds* )

Resume one or more existing QoS flows

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| | *pQosIds[IN]* | • See swiQosIds for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.11 ULONG SLQSQosSuspend ( BYTE *instance,* swiQosIds * *pQosIds* )

Suspend one or more existing QoS flows

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| | *pQosIds[IN]* | • See swiQosIds for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.12 ULONG SLQSQosSwiReadApnExtraParams ( BYTE *instance,* ULONG *apnId,* sApnExtraParams * *pApnExtraParams* )

Queries extra APN parameters that are not reported by existing QCT QMI service

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| in | *apnId* | • APN id |
| out | *pApnExtra-Params* | See sApnExtraParams for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.22.3.13 ULONG SLQSQosSwiReadDataStats ( BYTE *instance*, ULONG *apnId*, sQosStat * *pQosStat* )

Get the current number of packets and bytes sent, dropped and received for each UL, DL bearer and a sum of them for UL and DL direction in the modem.

**Parameters**

| | | |
|---|---|---|
| in | *instance* | • QMI instance |
| in | *apnId* | • APN id |
| out | *pQosStat* | See sQosStat for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.23 qaGobiApiRms.h File Reference

Remote Management Service API function prototypes.

### Functions

- ULONG GetSMSWake (ULONG *pEnabled, ULONG *pWakeMask)
- ULONG SetSMSWake (ULONG bEnable, ULONG wakeMask)

### 9.23.1 Detailed Description

Remote Management Service API function prototypes.

### 9.23.2 Function Documentation

#### 9.23.2.1 ULONG GetSMSWake ( ULONG * *pEnabled*, ULONG * *pWakeMask* )

Queries the state of the SMS wake functionality. When enabled SMS wake functionality results in incoming messages being searched for the configured mask. Upon detection of the mask the incoming message is deleted (i.e. not stored in memory) and the device attempts to wake the host (requires host platform support).

**Parameters**

| | |
|---|---|
| *pEnabled[OUT]* | • SMS wake functionality enabled<br>– 0 - Disabled<br>– 1 - Enabled |
| *pWakeMask[OUT]* | • SMS wake mask to search for incoming messages (only relevant when enabled) |

**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

#### 9.23.2.2 ULONG SetSMSWake ( ULONG *bEnable,* ULONG *wakeMask* )

Configures the SMS wake functionality. When enabled SMS wake functionality results in incoming messages being searched for the configured mask. Upon detection of the mask the incoming message is deleted (i.e. not stored in memory) and the device attempts to wake the host (requires host platform support).

**Parameters**

| | |
|---|---|
| *bEnable* | • Enable SMS wake functionality<br>– Zero - Disable<br>– Non-Zero - Enable |
| *wakeMask* | • SMS wake mask to search for incoming messages (only relevant when enabling) |

**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

## 9.24 qaGobiApiSar.h File Reference

Specific Absorption Rate API function prototypes.

## Enumerations

- enum eQMISARRFState {
  QMI_SAR_RF_STATE_DEFAULT = 0,
  QMI_SAR_RF_STATE_1,
  QMI_SAR_RF_STATE_2,
  QMI_SAR_RF_STATE_3,
  QMI_SAR_RF_STATE_4,
  QMI_SAR_RF_STATE_5,
  QMI_SAR_RF_STATE_6,
  QMI_SAR_RF_STATE_7,
  QMI_SAR_RF_STATE_8 }

## Functions

- ULONG SLQSGetRfSarState (ULONG *pSarRFState)
- ULONG SLQSSetRfSarState (ULONG RfSarState)

### 9.24.1 Detailed Description

Specific Absorption Rate API function prototypes.

### 9.24.2 Enumeration Type Documentation

#### 9.24.2.1 enum eQMISARRFState

This enum contains the SAR RF States

**Parameters**

| | |
|---|---|
| *SAR* | RF State<br>• QMI_SAR_RF_STATE_DEFAULT = 0<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 |

**Enumerator**

***QMI_SAR_RF_STATE_DEFAULT***

***QMI_SAR_RF_STATE_1***

***QMI_SAR_RF_STATE_2***

***QMI_SAR_RF_STATE_3***

***QMI_SAR_RF_STATE_4***

***QMI_SAR_RF_STATE_5***

***QMI_SAR_RF_STATE_6***

***QMI_SAR_RF_STATE_7***

***QMI_SAR_RF_STATE_8***

### 9.24.3 Function Documentation

#### 9.24.3.1 ULONG SLQSGetRfSarState ( ULONG * *pSarRFState* )

Gets the specified RF SAR state.

**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 |

**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

#### 9.24.3.2 ULONG SLQSSetRfSarState ( ULONG *RfSarState* )

Sets the specified RF SAR state.

**Parameters**

| | |
|---|---|
| *sar_rf_state* | • 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 |

#### 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

## 9.25 qaGobiApiSms.h File Reference

Short Message Service API function prototypes.

### Data Structures

- struct slqssendsmsparams_s
- struct BroadcastConfig
- struct _qaQmi3GPPBroadcastCfgInfo
- struct CDMABroadcastConfig
- struct _qaQmi3GPP2BroadcastCfgInfo
- struct cdmaMsgEncodingParams
- struct cdmaMsgDecodingParams
- struct cdmaMsgDecodingParamsExt
- struct wcdmaMsgEncodingParams
- struct wcdmaMsgDecodingParams
- struct wcdmaLongMsgDecodingParams
- struct _transLayerinfo
- struct _getTransLayerInfoResp
- struct _getTransNWRegInfoResp
- struct _getIndicationRegResp
- struct _setIndicationRegReq
- struct smsRouteEntry
- struct smsSetRoutesReq
- struct smsMsgprotocolResp
- struct smsMaxStorageSizeReq
- struct smsMaxStorageSizeResp
- struct messageWaitingInfoContent
- struct getMsgWaitingInfo
- struct slqssendasyncsmsparams_s

### Macros

- #define CONFIG_LEN 0x05
- #define TIME_STAMP_BUF 0x08
- #define ABSOLUTE_VALIDITY 0x08
- #define TIME_DATE_BUF 0x09
- #define MAX_SMS_ROUTES 0x0A
- #define NUM_OF_SET 0xFF

## Typedefs

- typedef struct _qaQmi3GPPBroadcastCfgInfo qaQmi3GPPBroadcastCfgInfo
- typedef struct _qaQmi3GPP2BroadcastCfgInfo qaQmi3GPP2BroadcastCfgInfo
- typedef struct _transLayerinfo transLayerInfo
- typedef struct _getTransLayerInfoResp getTransLayerInfoResp
- typedef struct _getTransNWRegInfoResp getTransNWRegInfoResp
- typedef struct _getIndicationRegResp getIndicationRegResp
- typedef struct _setIndicationRegReq setIndicationRegReq

## Functions

- ULONG SLQSDeleteSMS (ULONG storageType, ULONG *pMessageIndex, ULONG *pMessageTag, BYTE *pMessageMode)
- ULONG SLQSGetSMS (ULONG storageType, ULONG messageIndex, ULONG *pMessageTag, ULONG *pMessageFormat, ULONG *pMessageSize, BYTE *pMessage, BYTE *pMessageMode)
- ULONG SendSMS (ULONG messageFormat, ULONG messageSize, BYTE *pMessage, ULONG *pMessageFailureCode, BYTE *pSmsOnIms)
- ULONG SLQSSendSMS (slqssendsmsparams_s *pSendSmsParams)
- ULONG GetSMSCAddress (BYTE addressSize, CHAR *pSMSCAddress, BYTE typeSize, CHAR *pSMSCType)
- ULONG SetSMSCAddress (CHAR *pSMSCAddress, CHAR *pSMSCType)
- ULONG SaveSMS (ULONG storageType, ULONG messageFormat, ULONG messageSize, BYTE *pMessage, ULONG *pMessageIndex)
- ULONG SLQSGetSMSList (ULONG storageType, ULONG *pRequestedTag, ULONG *pMessageListSize, BYTE *pMessageList, BYTE *pMessageMode)
- ULONG SLQSModifySMSStatus (ULONG storageType, ULONG messageIndex, ULONG messageTag, BYTE *pMessageMode)
- ULONG SLQSGetSmsBroadcastConfig (BYTE mode, qaQmi3GPPBroadcastCfgInfo *pBroadcastConfig, qaQmi3GPP2BroadcastCfgInfo *pCDMABroadcastConfig)
- ULONG SLQSSetSmsBroadcastConfig (BYTE mode, qaQmi3GPPBroadcastCfgInfo *pBroadcastConfig, qaQmi3GPP2BroadcastCfgInfo *pCDMABroadcastConfig)
- ULONG SLQSSetSmsBroadcastActivation (BYTE mode, BYTE broadcastActivate)
- ULONG SLQSCDMAEncodeMOTextMsg (struct cdmaMsgEncodingParams *pCdmaMsgEncodingParams)
- ULONG SLQSCDMADecodeMTTextMsg (struct cdmaMsgDecodingParams *pCdmaMsgDecodingParams)
- ULONG SLQSCDMADecodeMTTextMsgExt (struct cdmaMsgDecodingParamsExt *pCdmaMsgDecodingParamsExt)
- ULONG SLQSWCDMAEncodeMOTextMsg (struct wcdmaMsgEncodingParams *pWcdmaMsgEncodingParams)
- ULONG SLQSWCDMADecodeMTTextMsg (struct wcdmaMsgDecodingParams *pWcdmaMsgDecodingParams)
- ULONG SLQSWCDMADecodeLongTextMsg (struct wcdmaLongMsgDecodingParams *pWcdmaLongMsgDecodingParams)
- ULONG SLQSGetTransLayerInfo (getTransLayerInfoResp *pGetTransLayerInfoResp)
- ULONG SLQSGetTransNWRegInfo (getTransNWRegInfoResp *pGetTransNWRegInfoResp)
- ULONG SLQSGetIndicationRegister (getIndicationRegResp *pGetIndicationRegInfo)
- ULONG SLQSSetIndicationRegister (setIndicationRegReq *pSetIndicationRegReq)
- ULONG SLQSSmsSetRoutes (smsSetRoutesReq *pSetRoutesReq)
- ULONG SLQSSmsGetMessageProtocol (smsMsgprotocolResp *pMessageProtocol)
- ULONG SLQSSmsGetMaxStorageSize (smsMaxStorageSizeReq *pMaxStorageSizeReq, smsMaxStorageSizeResp *pMaxStorageSizeResp)

- ULONG SLQSGetMessageWaiting (getMsgWaitingInfo *pGetMsgWaitingInfoResp)
- ULONG SLQSSendAsyncSMS (slqssendasyncsmsparams_s *pSendSmsParams)
- ULONG SLQSSetSmsStorage (BYTE smsStorage)
- ULONG SLQSSwiGetSMSStorage (ULONG *pSmsStorage)
- ULONG SLQSSendLongSMS (ULONG messageFormat, ULONG messageSize, CHAR *pMessage, BYTE encodingScheme, ULONG *pMessageFailureCode, CHAR *pMobileNum, BYTE *pSmsOnIMS)

### 9.25.1 Detailed Description

Short Message Service API function prototypes.

### 9.25.2 Macro Definition Documentation

#### 9.25.2.1 #define ABSOLUTE_VALIDITY 0x08

#### 9.25.2.2 #define CONFIG_LEN 0x05

#### 9.25.2.3 #define MAX_SMS_ROUTES 0x0A

#### 9.25.2.4 #define NUM_OF_SET 0xFF

#### 9.25.2.5 #define TIME_DATE_BUF 0x09

#### 9.25.2.6 #define TIME_STAMP_BUF 0x08

### 9.25.3 Typedef Documentation

#### 9.25.3.1 typedef struct _getIndicationRegResp getIndicationRegResp

This structure contains Get Indication Register Response parameters

**Parameters**

| | |
|---|---|
| *pRegTrans-LayerInfoEvt* | -<br>• Optional 1 BYTE parameter indicating registration status of transport layer information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• function SLQSGetIndicationRegister() returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |
| *pRegTransNW-RegInfoEvt* | -<br>• Optional 1 BYTEparameter indicating registration status of transport network registration information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• function SLQSGetIndicationRegister() returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |

| | |
|---|---|
| *pRegCallStat-InfoEvt* | -<br>• Optional 1 BYTE parameter indicating registration status of call status information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• function SLQSGetIndicationRegister() returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device. |

#### 9.25.3.2 typedef struct _getTransLayerInfoResp getTransLayerInfoResp

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>• function SLQSGetTransLayerInfo() returns a default value 0xFF if no response is received from the device. |
| *pTransLayerInfo* | • Pointer to structure of transLayerInfo.<br>– Optional parameter<br>– See transLayerInfo for more information<br>• function SLQSGetTransLayerInfo() returns a default value 0xFF for parameter values if no response is received from the device. |

#### 9.25.3.3 typedef struct _getTransNWRegInfoResp getTransNWRegInfoResp

This structure contains transport network registration info parameter

**Parameters**

| | |
|---|---|
| *pRegStatus* | -<br>• Optional 1 BYTE 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>• function SLQSGetTransNWRegInfo() returns a default value 0xFF if no response is received from the device. |

#### 9.25.3.4 typedef struct _qaQmi3GPP2BroadcastCfgInfo qaQmi3GPP2BroadcastCfgInfo

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 CDMABroadcastConfig |

#### 9.25.3.5 typedef struct _qaQmi3GPPBroadcastCfgInfo qaQmi3GPPBroadcastCfgInfo

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 BroadcastConfig |

#### 9.25.3.6 typedef struct _setIndicationRegReq setIndicationRegReq

This structure contains Indication Register request parameters

**Parameters**

| | |
|---|---|
| *pRegTrans-LayerInfoEvt* | -<br>• Optional 1 BYTE 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 1 BYTE 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 1 BYTE 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 |

#### 9.25.3.7 typedef struct _transLayerinfo transLayerInfo

This structure contains Transport Layer Information

**Parameters**

| | |
|---|---|
| *TransType* | • Transport Type<br>– 0x00 - IMS |
| *TransCap* | • Transport Capability<br>• Values:<br>– 0x00 - CDMA<br>– 0x01 - GW |

### 9.25.4 Function Documentation

#### 9.25.4.1 ULONG GetSMSCAddress ( BYTE *addressSize,* CHAR * *pSMSCAddress,* BYTE *typeSize,* CHAR * *pSMSCType* )

Gets the SMS center address.

**Parameters**

| | |
|---|---|
| *addressSize* | • The maximum number of characters (including NULL terminator) that the SMS center address array can contain. |
| *pSMSC-Address[0UT]* | • The SMS center address represented as a NULL terminated string. |
| *typeSize* | • The maximum number of characters (including NULL terminator) that the SMS center address type array can contain. |
| *pSMSCType[0U-T]* | • The SMS center address type represented as a NULL terminated string. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 2 seconds

#### 9.25.4.2 ULONG SaveSMS ( ULONG *storageType,* ULONG *messageFormat,* ULONG *messageSize,* BYTE * *pMessage,* ULONG * *pMessageIndex* )

Saves an SMS message to device memory

**Parameters**

| | |
|---|---|
| *storageType[IN]* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *message-Format[IN]* | • 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[IN]* | • The message contents |
| *pMessage-Index[OUT]* | • The message index assigned by the device |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 10 seconds

#### 9.25.4.3 ULONG SendSMS ( ULONG *messageFormat,* ULONG *messageSize,* BYTE * *pMessage,* ULONG * *pMessageFailureCode,* BYTE * *pSmsOnIms* )

Sends an SMS message for immediate over-the-air transmission

**Parameters**

| | |
|---|---|
| *messageFormat[IN]* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *messageSize[IN]* | • The length of the message contents in bytes |
| *pMessage[IN]* | • The message contents in PDU format contains SMS header and payload message |
| *pSmsOnIms[IN]* | • (Optional) SMS on IMS<br>• The message is to be sent on IMS.<br>– 0x00 Message is not to be sent on IMS.<br>– 0x01 Message is to be sent on IMS.<br>– 0x02 to 0xFF Reserved. |
| *pMessageFailureCode[OUT]* | • (Optional) Message Failure Code<br>• pointer to message failure code. If cause code is not provided, then value will be 0xFFFFFFFF |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 minutes

#### 9.25.4.4 ULONG SetSMSCAddress ( CHAR * *pSMSCAddress,* CHAR * *pSMSCType* )

Sets the SMS center address.

**Parameters**

| | |
|---|---|
| *pSMSC-Address[IN]* | • The SMS center address represented as a NULL terminated string |
| *pSMSCType[IN]* | • The SMS center address type represented as a NULL terminated string (optional). |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.5 ULONG SLQSCDMADecodeMTTextMsg ( struct cdmaMsgDecodingParams * *pCdmaMsgDecodingParams* )

Decodes text message to CDMA PDU message

**Parameters**

| | |
|---|---|
| *pMsgToBe-EncodedCDMA[-IN/OUT]* | • Pointer to structure containing parameters needed for decoding |

**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
Timeout: None

#### 9.25.4.6 ULONG SLQSCDMADecodeMTTextMsgExt ( struct cdmaMsgDecodingParamsExt * *pCdmaMsgDecodingParamsExt* )

Decodes text message to CDMA PDU message Ext

**Parameters**

| | |
|---|---|
| *pCdmaMsg-Decoding-ParamsExt[IN/O-UT]* | • Pointer to structure containing parameters needed for decoding |

**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
Timeout: None

#### 9.25.4.7 ULONG SLQSCDMAEncodeMOTextMsg ( struct cdmaMsgEncodingParams * *pCdmaMsgEncodingParams* )

Encodes text message to CDMA PDU message.

**Parameters**

| | |
|---|---|
| *pMsgToBe-EncodedCDMA[-IN/OUT]* | • SLQS Runtime Settings 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
Timeout: None

#### 9.25.4.8 ULONG SLQSDeleteSMS ( ULONG *storageType,* ULONG * *pMessageIndex,* ULONG * *pMessageTag,* BYTE * *pMessageMode* )

Deletes one or more SMSs from device memory. If both of the optional parameters, messageIndex and messageTag, are NULL, all messages are deleted from the storage location specified in the mandatory storageType parameter. The optional index and tag parameters narrow the range of messages being deleted. If an index is specified, the single message with the index from the specified memory store will be deleted. If a tag is specified, all messages in the specified memory store whose tag matches that specified will be deleted.

There are three ways to use this message:

- Specify storageType only

  - Deletes all messages from memory storage

- Specify storageType and a tag
  - Deletes all messages from memory storage that match the given message tag
- Specify storageType and an index
  - Deletes only the message with the given index from memory storage

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *pMessage-Index[IN]* | • (Optional) message index |
| *pMessageTag[I-N]* | • (Optional) message tag<br>– 0 - Read<br>– 1 - Not read<br>– 2 - Mobile originated and sent<br>– 3 - Mobile originated but not yet sent |
| *pMessage-Mode[IN]* | • (Optional) messgae mode<br>• 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 10 seconds

#### 9.25.4.9 ULONG SLQSGetIndicationRegister ( getIndicationRegResp * *pGetIndicationRegInfo* )

This API provides registration state of different WMS indications.

**Parameters**

| | |
|---|---|
| *pGetIndication-RegInfo* | [OUT]<br>• Pointer to structure of getIndicationRegResp<br>– See getIndicationRegResp 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 Secs

#### 9.25.4.10 ULONG SLQSGetMessageWaiting ( getMsgWaitingInfo * *pGetMsgWaitingInfoResp* )

This API provides information about the message waiting information.

**Parameters**

| | |
|---|---|
| *pGetMsg-WaitingInfoResp* | [OUT]<br>• Pointer to structure of getMsgWaitingInfoResp<br>– See getMsgWaitingInfoResp 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 Secs

#### 9.25.4.11 ULONG SLQSGetSMS ( ULONG *storageType,* ULONG *messageIndex,* ULONG * *pMessageTag,* ULONG * *pMessageFormat,* ULONG * *pMessageSize,* BYTE * *pMessage,* BYTE * *pMessageMode* )

Returns an SMS from 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 |
| *messageIndex* | • Message index |
| *pMessageTag[-OUT]* | • Message tag<br>– 0 - Read<br>– 1 - Not read<br>– 2 - Mobile originated and sent<br>– 3 - Mobile originated but not yet sent |

| | |
|---|---|
| *pMessage-Format[OUT]* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *pMessageSize[IN/OUT]* | • 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. |
| *pMessage[OUT]* | • The message contents array |
| *pMessage-Mode[IN]* | • (Optional) Message Mode<br>• 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.12 ULONG SLQSGetSmsBroadcastConfig ( BYTE *mode,* qaQmi3GPPBroadcastCfgInfo * *pBroadcastConfig,* qaQmi3GPP2BroadcastCfgInfo * *pCDMABroadcastConfig* )

Provides Information about the SMS BroadcastConfiguration

**Parameters**

| | |
|---|---|
| *mode[IN]* | • Mode<br>– 0x00 - CDMA, LTE (if network type is CDMA)<br>– 0x01 - GW, LTE (if network type is UMTS) |
| *pBroadcast-Config[OUT]* | • The data for 3GPP Broadcast Information(Optional). |
| *pCDMA-Broadcast-Config[OUT]* | • The data for 3GPP2 Broadcast Information(Optional). |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.13 ULONG SLQSGetSMSList ( ULONG *storageType,* ULONG * *pRequestedTag,* ULONG * *pMessageListSize,* BYTE * *pMessageList,* BYTE * *pMessageMode* )

Returns the list of SMS messages stored on the device.

**Parameters**

| | |
|---|---|
| *storageType[IN]* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *pRequested-Tag[IN]* | • (Optional) Message tag<br>– 0 - Read<br>– 1 - Not read<br>– 2 - Mobile originated and sent<br>– 3 - Mobile originated but not yet sent |
| *pMessageList-Size[IN/OUT]* | • 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. |
| *pMessageList[O-UT]* | • The message list array |
| *pMessage-Mode[IN]* | • (Optional) Message Mode<br>• 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.14 ULONG SLQSGetTransLayerInfo ( getTransLayerInfoResp * *pGetTransLayerInfoResp* )

This API provides information about the transport layer.

**Parameters**

| | |
|---|---|
| *pGetTransLayer-InfoResp* | [OUT]<br>• Pointer to structure of getTransLayerInfoResp<br>– See getTransLayerInfoResp 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 Secs

#### 9.25.4.15 ULONG SLQSGetTransNWRegInfo ( getTransNWRegInfoResp * *pGetTransNWRegInfoResp* )

This API provides transport layer network registration info.

**Parameters**

| | |
|---|---|
| *pGetTransNW-RegInfoResp* | [OUT]<br>• Pointer to structure of getTransNWRegInfoResp<br>– See getTransNWRegInfoResp 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 Secs

#### 9.25.4.16 ULONG SLQSModifySMSStatus ( ULONG *storageType*, ULONG *messageIndex*, ULONG *messageTag*, BYTE * *pMessageMode* )

Modifies the status of an SMS message saved in storage on the device.

**Parameters**

| | |
|---|---|
| *storageType[IN]* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *messageIndex[IN]* | • Message index |
| *messageTag[IN]* | • Message tag<br>– 0 - Read<br>– 1 - Not read |
| *pMessageMode[IN]* | • (Optional) Message Mode<br>• 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.17 ULONG SLQSSendAsyncSMS ( slqssendasyncsmsparams_s * *pSendSmsParams* )

Sends an SMS message for immediate over-the-air transmission

**Parameters**

| | |
|---|---|
| *pSendSmsParams* | • structure containing the SMS parameters. Refer slqssendasyncsmsparams_s |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 minutes

#### 9.25.4.18 ULONG SLQSSendLongSMS ( ULONG *messageFormat*, ULONG *messageSize*, CHAR * *pMessage*, BYTE *encodingScheme*, ULONG * *pMessageFailureCode*, CHAR * *pMobileNum*, BYTE * *pSmsOnIMS* )

Sends a long SMS message for immediate over-the-air transmission, a short SMS can be sent by this API as well, the input message is text string without any encoding

**Parameters**

| | |
|---|---|
| *messageFormat[IN]* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *messageSize[IN]* | • Message size of the input message text |
| *pMessage[IN]* | • Original message text |
| *encodingScheme[IN]* | • Encoding method to generate the PDU<br>– 0 - 7 bit encoding<br>– 4 - 8 bit encoding<br>– 8 - 16 bit UCS2 encoding<br>– others value will be treated as default 7 bit encoding |
| *pMessageFailureCode[OUT]* | • message failure code. If cause code is not provided, then value will be 0xFFFFFFFF |
| *pMobileNum[IN]* | • Mobile number of the receiver |
| *pSmsOnIMS[IN]* | • A flag indicates whether SMS was sent through IMS |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: 3GPP and 3GPP2, but 3GPP2 does not support multiple sms
Timeout: 5 minutes

#### 9.25.4.19 ULONG SLQSSendSMS ( slqssendsmsparams_s * *pSendSmsParams* )

Sends an SMS message for immediate over-the-air transmission

**Parameters**

| | |
|---|---|
| *pSendSms-Params* | • structure containing the SMS parameters. Refer slqssendsmsparams_s |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 minutes

#### 9.25.4.20 ULONG SLQSSetIndicationRegister ( setIndicationRegReq * *pSetIndicationRegReq* )

This API sets the registration state of different WMS indications.

**Parameters**

| | |
|---|---|
| *pSetIndication-RegReq* | [IN]<br>• Pointer to structure of indicationRegReqParams<br>– See setIndicationRegReq 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 Secs

#### 9.25.4.21 ULONG SLQSSetSmsBroadcastActivation ( BYTE *mode,* BYTE *broadcastActivate* )

Enables or disables the reception of broadcast SMS messages.

**Parameters**

| | |
|---|---|
| *Mode[IN]* | • Mode<br>• 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |
| *broadcast-Activate[IN]* | • 0x00 - Disable broadcast |
| | • 0x01 - Activate broadcast |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.22 ULONG SLQSSetSmsBroadcastConfig ( BYTE *mode*, qaQmi3GPPBroadcastCfgInfo * *pBroadcastConfig*, qaQmi3GPP2BroadcastCfgInfo * *pCDMABroadcastConfig* )

Sets the information about the SMS BroadcastConfiguration

**Parameters**

| | |
|---|---|
| *mode[IN]* | • Mode<br>– 0x00 - CDMA, LTE (if network type is CDMA)<br>– 0x01 - GW, LTE (if network type is UMTS) |
| *pBroadcast-Config[IN]* | • The data for 3GPP Broadcast Information(Optional). |
| *pCDMA-Broadcast-Config[IN]* | • The data for 3GPP2 Broadcast Information(Optional). |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.23 ULONG SLQSSetSmsStorage ( BYTE *smsStorage* )

Sets the SMS Storage on the device

**Parameters**

| | |
|---|---|
| *smsStorage[IN]* | • SMS Storage<br>– 0x01 - device's permanent memory<br>– 0x02 - UICC |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: ALL
Timeout: 5 seconds

#### 9.25.4.24 ULONG SLQSSmsGetMaxStorageSize ( smsMaxStorageSizeReq * *pMaxStorageSizeReq*, smsMaxStorageSizeResp * *pMaxStorageSizeResp* )

This API 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-SizeReq[IN]* | • Request parameters for SmsSLQSGetMaxStorageSize<br>– See smsMaxStorageSizeReq for more information |
| *pMaxStorage-SizeResp[OUT]* | • Response parameters for SmsSLQSGetMaxStorageSize<br>– See smsMaxStorageSizeResp 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 Secs

#### 9.25.4.25 ULONG SLQSSmsGetMessageProtocol ( smsMsgprotocolResp * *pMessageProtocol* )

This API queries the message protocol currently in use for the WMS client.

**Parameters**

| | |
|---|---|
| *pMessage-Protocol* | [OUT}<br>• Pointer to smsMsgprotocolResp<br>– See smsMsgprotocolResp 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 Secs

#### 9.25.4.26 ULONG SLQSSmsSetRoutes ( smsSetRoutesReq * *pSetRoutesReq* )

This API 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* | [IN]<br>• Pointer to structure of smsSetRoutesReq<br>– See smsSetRoutesReq 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 Secs

#### 9.25.4.27 ULONG SLQSSwiGetSMSStorage ( ULONG * *pSmsStorage* )

This API queries the device to return current SMS configuration that is applied to all incoming and outgoing messages.

**Parameters**

| | |
|---|---|
| *pSmsStorage[OUT]* | • Values:<br>– 0x01 - device's permanent memory<br>– 0x02 - UICC |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 Secs

#### 9.25.4.28 ULONG SLQSWCDMADecodeLongTextMsg ( struct wcdmaLongMsgDecodingParams * *pWcdmaLongMsgDecodingParams* )

Decodes WCDMA Long SMS PDU message, returns structure filled with decoded parameters

**Parameters**

| | |
|---|---|
| *pwcdmaMsg-Decoding-Params[IN/OUT]* | • Pointer to parameters required for decoding |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: none

#### 9.25.4.29 ULONG SLQSWCDMADecodeMTTextMsg ( struct wcdmaMsgDecodingParams * *pWcdmaMsgDecodingParams* )

Decodes WCDMA PDU message, returns structure filled with decoded parameters

**Parameters**

| | |
|---|---|
| *pwcdmaMsg-Decoding-Params[IN/OUT]* | • Pointer to parameters required for decoding |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: none

#### 9.25.4.30 ULONG SLQSWCDMAEncodeMOTextMsg ( struct wcdmaMsgEncodingParams * *pWcdmaMsgEncodingParams* )

Returns the encoded WCDMA PDU message.

**Parameters**

| | |
|---|---|
| *pwcdmaMsg-Encoding-Params[IN/OUT]* | • Pointer to parameters Required for encoding |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: None

## 9.26 qaGobiApiSwi.h File Reference

SWI API function prototypes.

### Functions

- ULONG SLQSGetSdkVersion (CHAR **sdkversionpp)
- ULONG SLQSSendRawQMI (BYTE *pReqBuf, USHORT service, USHORT length, ULONG timeout, BYTE **ppInParm, USHORT *pParamLength)
- int SLQSGetPidof (CHAR *pProcName)

### 9.26.1 Detailed Description

SWI API function prototypes.

### 9.26.2 Function Documentation

#### 9.26.2.1 int SLQSGetPidof ( CHAR * *pProcName* )

Internal Wrapper function for enabling invocation of SLQS implementation pidof() function

**Parameters**

| | |
|---|---|
| *pProcName[IN]* | • Process name whose PID is to be retrieved |

**Returns**

pid if process exists else 0

**See Also**

NA

**Note**

NA

#### 9.26.2.2 ULONG SLQSGetSdkVersion ( CHAR ** *sdkversionpp* )

Returns the SDK version string

**Parameters**

| | |
|---|---|
| *ppString[OUT]* | • pointer to pointer of NULL terminated string |

**Returns**

eQCWWAN_ERR_NONE success eQCWWAN_ERR_INVALID_ARG provided pointer is NULL

**Note**

Technology Supported: N/A Timeout: 2 seconds

#### 9.26.2.3 ULONG SLQSSendRawQMI ( BYTE * *pReqBuf,* USHORT *service,* USHORT *length,* ULONG *timeout,* BYTE ** *ppInParm,* USHORT * *pParamLength* )

## 9.27 qaGobiApiSwiAudio.h File Reference

M2M Audio Service API function prototypes.

### Data Structures

- struct GetM2MAudioProfileReq
- struct GetM2MAudioProfileResp
- struct SetM2MAudioProfileReq
- struct GetM2MAudioVolumeReq
- struct GetM2MAudioVolumeResp
- struct SetM2MAudioVolumeReq
- struct PCMparams
- struct SetM2MAudioAVCFGReq
- struct SetM2MAudioLPBKReq
- struct GetM2MSpkrGainReq
- struct GetM2MSpkrGainResp
- struct SetM2MSpkrGainReq
- struct GetM2MAVMuteReq
- struct GetM2MAVMuteResp
- struct SetM2MAVMuteReq

### Macros

- #define MAX_LEN_IFACE_TABLE 255

### Functions

- ULONG SLQSGetM2MAudioProfile (GetM2MAudioProfileReq *pGetM2MAudioProfileReq, GetM2MAudioProfileResp *pGetM2MAudioProfileResp)
- ULONG SLQSSetM2MAudioProfile (SetM2MAudioProfileReq *pSetM2MAudioProfileReq)
- ULONG SLQSGetM2MAudioVolume (GetM2MAudioVolumeReq *pGetM2MAudioVolumeReq, GetM2MAudioVolumeResp *pGetM2MAudioVolumeResp)
- ULONG SLQSSetM2MAudioVolume (SetM2MAudioVolumeReq *pSetM2MAudioVolumeReq)
- ULONG SLQSSetM2MAudioAVCFG (SetM2MAudioAVCFGReq *pSetM2MAudioAVCFGReq)
- ULONG SLQSSetM2MAudioLPBK (SetM2MAudioLPBKReq *pSetM2MAudioLPBKReq)
- ULONG SLQSSetM2MAudioNVDef ()
- ULONG SLQSGetM2MSpkrGain (GetM2MSpkrGainReq *pSpkrGainReq, GetM2MSpkrGainResp *pSpkrGainResp)
- ULONG SLQSSetM2MSpkrGain (SetM2MSpkrGainReq *pSpkrGainReq)
- ULONG SLQSGetM2MAVMute (GetM2MAVMuteReq *pGetM2MAVMuteReq, GetM2MAVMuteResp *pGetM2MAVMuteResp)
- ULONG SLQSSetM2MAVMute (SetM2MAVMuteReq *pSetM2MAVMuteReq)

## 9.27.1 Detailed Description

M2M Audio Service API function prototypes.

## 9.27.2 Macro Definition Documentation

### 9.27.2.1 #define MAX_LEN_IFACE_TABLE 255

## 9.27.3 Function Documentation

### 9.27.3.1 ULONG SLQSGetM2MAudioProfile ( GetM2MAudioProfileReq * *pGetM2MAudioProfileReq*, GetM2MAudioProfileResp * *pGetM2MAudioProfileResp* )

This API gets the profile content.

**Parameters**

| | |
|---|---|
| *pGetM2MAudioProfileReq[OUT]* | • See GetM2MAudioProfileReq for more information |
| *pGetM2MAudioProfileResp[OUT]* | • See GetM2MAudioProfileResp 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: 5 seconds

#### 9.27.3.2 ULONG SLQSGetM2MAudioVolume ( GetM2MAudioVolumeReq * *pGetM2MAudioVolumeReq*, GetM2MAudioVolumeResp * *pGetM2MAudioVolumeResp* )

This API gets the Volume content.

**Parameters**

| | |
|---|---|
| *pGetM2MAudioVolumeReq[IN]* | • See GetM2MAudioVolumeReq for more information |
| *pGetM2MAudioVolumeResp[OUT]* | • See GetM2MAudioVolumeResp 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: 5 seconds

#### 9.27.3.3 ULONG SLQSGetM2MAVMute ( GetM2MAVMuteReq * *pGetM2MAVMuteReq*, GetM2MAVMuteResp * *pGetM2MAVMuteResp* )

This API Gets the AV Mute content.

**Parameters**

| | |
|---|---|
| *pGetM2MAVMuteReq[IN]* | • See GetM2MAVMuteReq for more information |
| *pGetM2MAVMuteResp[OUT]* | • See GetM2MAVMuteResp 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: 5 seconds

#### 9.27.3.4 ULONG SLQSGetM2MSpkrGain ( GetM2MSpkrGainReq * *pSpkrGainReq*, GetM2MSpkrGainResp * *pSpkrGainResp* )

This API Gets the SPKRGAIN content.

**Parameters**

| | |
|---|---|
| *pSpkrGainReq[IN]* | • See GetM2MSpkrGainReq for more information |
| *pSpkrGainResp[OUT]* | • See GetM2MSpkrGainResp 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: 5 seconds

#### 9.27.3.5 ULONG SLQSSetM2MAudioAVCFG ( SetM2MAudioAVCFGReq * *pSetM2MAudioAVCFGReq* )

This API sets the AVCFG content.

**Parameters**

| | |
|---|---|
| *pSetM2MAudioAVCFGReq[IN]* | • See SetM2MAudioAVCFGReq 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: 5 seconds

#### 9.27.3.6 ULONG SLQSSetM2MAudioLPBK ( SetM2MAudioLPBKReq * *pSetM2MAudioLPBKReq* )

This API sets the LPBK content.

**Parameters**

| | |
|---|---|
| *pSetM2MAudioLPBKReq[IN]* | • See SetM2MAudioLPBKReq 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: 5 seconds

#### 9.27.3.7 ULONG SLQSSetM2MAudioNVDef ( )

This API sets the NVDef content.

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

#### 9.27.3.8 ULONG SLQSSetM2MAudioProfile ( SetM2MAudioProfileReq * *pSetM2MAudioProfileReq* )

This API sets the profile content.

**Parameters**

| | |
|---|---|
| *pSetM2MAudio-ProfileReq[IN]* | • See SetM2MAudioProfileReq 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: 5 seconds

#### 9.27.3.9 ULONG SLQSSetM2MAudioVolume ( SetM2MAudioVolumeReq * *pSetM2MAudioVolumeReq* )

This API sets the Volume content.

**Parameters**

| | |
|---|---|
| *pSetM2MAudio-VolumeReq[IN]* | • See SetM2MAudioVolumeReq 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: 5 seconds

#### 9.27.3.10 ULONG SLQSSetM2MAVMute ( SetM2MAVMuteReq * *pSetM2MAVMuteReq* )

This API Sets the AV Mute content.

**Parameters**

| | |
|---|---|
| *pSetM2MAV-MuteReq[IN]* | • See SetM2MAVMuteReq 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: 5 seconds

#### 9.27.3.11 ULONG SLQSSetM2MSpkrGain ( SetM2MSpkrGainReq * *pSpkrGainReq* )

This API Sets the SPKRGAIN content.

**Parameters**

| | |
|---|---|
| *pSpkrGainReq[I-N]* | • See GetM2MSpkrGainReq 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: 5 seconds

## 9.28 qaGobiApiSwiAvms.h File Reference

SWI Open Mobile Alliance Device Management Service API function prototypes SWI AVMS QMI Service revision 1.14.

### Data Structures

- struct _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp
- struct _QmiSwiAvmsEventReportConfigResp
- struct _QmiSwiAvmsEventReportNotificationResp
- struct _QmiSwiAvmsEventReportConnectionRequestResp
- struct _QmiSwiAvmsEventReportWAMSParamChangeResp
- struct _QmiSwiAvmsEventReportPackageIDResp
- struct _QmiSwiAvmsEventReportRegStatusResp
- struct _QmiSwiAvmsEventReportDataSessionStatusResp
- struct _QmiSwiAvmsEventReportSessionTypeResp
- struct _QmiSwiAvmsEventReportHTTPStatusResp
- struct _SLQSAVMSSettingsAPNInfo
- struct _SLQSAVMSSettingsConnectionRetryTimers
- struct _SLQSAVMSSettingsPeriodsInfo
- struct _SLQSAVMSSettings
- struct _QmiSwiAvmsSetSettingsPeriodInfo
- struct _QmiSwiAvmsSetSettingsAPNInfo
- struct _QmiSwiAvmsSetSettingsConnectionRetryTimers
- struct _SLQSAVMSSetSettingsReqPara

### Macros

- #define MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH 49
- #define MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH 29
- #define MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH 29
- #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH 128
- #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH 128
- #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH 1024
- #define QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 200
- #define MAX_AVMS_RETRY_TIMER_NUMBER 8
- #define MAX_SWI_AVMS_SET_SETTING_APN_LENGTH 49
- #define MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH 29
- #define MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH 29
- #define MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 8

## Typedefs

- typedef struct _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp
- typedef struct _QmiSwiAvmsEventReportConfigResp QmiSwiAvmsEventReportConfigResp
- typedef struct _QmiSwiAvmsEventReportNotificationResp QmiSwiAvmsEventReportNotificationResp
- typedef struct _QmiSwiAvmsEventReportConnectionRequestResp QmiSwiAvmsEventReportConnectionRequestResp
- typedef struct _QmiSwiAvmsEventReportWAMSParamChangeResp QmiSwiAvmsEventReportWAMSParamChangeResp
- typedef struct _QmiSwiAvmsEventReportPackageIDResp QmiSwiAvmsEventReportPackageIDResp
- typedef struct _QmiSwiAvmsEventReportRegStatusResp QmiSwiAvmsEventReportRegStatusResp
- typedef struct _QmiSwiAvmsEventReportDataSessionStatusResp QmiSwiAvmsEventReportDataSessionStatusResp
- typedef struct _QmiSwiAvmsEventReportSessionTypeResp QmiSwiAvmsEventReportSessionTypeResp
- typedef struct _QmiSwiAvmsEventReportHTTPStatusResp QmiSwiAvmsEventReportHTTPStatusResp
- typedef struct _SLQSAVMSSettingsAPNInfo SLQSAVMSSettingsAPNInfo
- typedef struct _SLQSAVMSSettingsConnectionRetryTimers SLQSAVMSSettingsConnectionRetryTimers
- typedef struct _SLQSAVMSSettingsPeriodsInfo SLQSAVMSSettingsPeriodsInfo
- typedef struct _SLQSAVMSSettings SLQSAVMSSettings
- typedef struct _QmiSwiAvmsSetSettingsPeriodInfo QmiSwiAvmsSetSettingsPeriodInfo
- typedef struct _QmiSwiAvmsSetSettingsAPNInfo QmiSwiAvmsSetSettingsAPNInfo
- typedef struct _QmiSwiAvmsSetSettingsConnectionRetryTimers QmiSwiAvmsSetSettingsConnectionRetryTimers
- typedef struct _SLQSAVMSSetSettingsReqPara SLQSAVMSSetSettingsReq

## Functions

- ULONG SLQSAVMSStartSession (BYTE sessionType, ULONG *pSessionResponse)
- ULONG SLQSAVMSStopSession (BYTE sessiontype, BYTE *pSessionRep)
- ULONG SLQSAVMSGetSessionInfo (QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp *pBinaryUpdteSessionInfo, QmiSwiAvmsEventReportConfigResp *pAVMSConfig, QmiSwiAvmsEventReportNotificationResp *pNotif, QmiSwiAvmsEventReportPackageIDResp *pPackageID)
- ULONG SLQSAVMSSelection (BYTE selection, ULONG *pDeferTime, BYTE *pClientPerfOperationFlag, BYTE *pPackageID, WORD *pRejectReason)
- ULONG SLQSAVMSGetSettings (SLQSAVMSSettings *settings)
- ULONG SLQSAVMSSetSettings (SLQSAVMSSetSettingsReq Request)
- ULONG SLQSAVMSSetSettingsIgnoreAutoReboot (SLQSAVMSSetSettingsReq Request)
- ULONG SLQSAVMSSetEventReport ()

### 9.28.1 Detailed Description

SWI Open Mobile Alliance Device Management Service API function prototypes SWI AVMS QMI Service revision 1.14.

### 9.28.2 Macro Definition Documentation

#### 9.28.2.1 #define MAX_AVMS_RETRY_TIMER_NUMBER 8

#### 9.28.2.2 #define MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH 49

#### 9.28.2.3 #define MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH 29

#### 9.28.2.4 #define MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH 29

#### 9.28.2.5 #define MAX_SWI_AVMS_SET_SETTING_APN_LENGTH 49

#### 9.28.2.6 #define MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 8

#### 9.28.2.7 #define MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH 29

#### 9.28.2.8 #define MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH 29

#### 9.28.2.9 #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH 1024

#### 9.28.2.10 #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH 128

#### 9.28.2.11 #define QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH 128

#### 9.28.2.12 #define QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 200

### 9.28.3 Typedef Documentation

#### 9.28.3.1 typedef struct _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp

**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 |

#### 9.28.3.2 typedef struct _QmiSwiAvmsEventReportConfigResp QmiSwiAvmsEventReportConfigResp

**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 |

#### 9.28.3.3 typedef struct _QmiSwiAvmsEventReportConnectionRequestResp QmiSwiAvmsEventReport-ConnectionRequestResp

**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 |

#### 9.28.3.4 typedef struct _QmiSwiAvmsEventReportDataSessionStatusResp QmiSwiAvmsEventReportDataSessionStatusResp

**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 |

#### 9.28.3.5 typedef struct _QmiSwiAvmsEventReportHTTPStatusResp QmiSwiAvmsEventReportHTTPStatusResp

**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 |

#### 9.28.3.6 typedef struct _QmiSwiAvmsEventReportNotificationResp QmiSwiAvmsEventReportNotification-Resp

**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 |

#### 9.28.3.7 typedef struct _QmiSwiAvmsEventReportPackageIDResp QmiSwiAvmsEventReportPackageIDResp

**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 |

#### 9.28.3.8 typedef struct _QmiSwiAvmsEventReportRegStatusResp QmiSwiAvmsEventReportRegStatusResp

**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 |

#### 9.28.3.9 typedef struct _QmiSwiAvmsEventReportSessionTypeResp QmiSwiAvmsEventReportSessionTypeResp

**Parameters**

| | |
|---|---|
| *bType* | Session Type.<br>• 0: Bootstrap session.<br>• 1. DM session |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

#### 9.28.3.10 typedef struct _QmiSwiAvmsEventReportWAMSParamChangeResp QmiSwiAvmsEventReportWAMSParamChangeResp

**Parameters**

| | |
|---|---|
| *wWamsChangeMask* | 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 |

#### 9.28.3.11 typedef struct _QmiSwiAvmsSetSettingsAPNInfo QmiSwiAvmsSetSettingsAPNInfo

#### 9.28.3.12 typedef struct _QmiSwiAvmsSetSettingsConnectionRetryTimers QmiSwiAvmsSetSettingsConnectionRetryTimers

#### 9.28.3.13 typedef struct _QmiSwiAvmsSetSettingsPeriodInfo QmiSwiAvmsSetSettingsPeriodInfo

#### 9.28.3.14 typedef struct _SLQSAVMSSetSettingsReqPara SLQSAVMSSetSettingsReq

Sets the AVMS settings requested.

**Parameters**

| | |
|---|---|
| *bAutoConnect[IN]* | • Auto Connect to AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *bAutoReboot[IN]* | • Automatic device reboot when the request is received from AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |

| | |
|---|---|
| *bFwPromptdownload[IN]* | • Start Firmware Download promptly/automatically<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *bFwPromptUpdate[IN]* | • Start Firmware Download promptly/automatically<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *FwAutoSDM[Optional]* | - OMA Automatic UI Alert Response<br>• 0x00 - DISABLED<br>• 0x01 - ENABLED ACCEPT<br>• 0x02 - ENABLED REJECT |
| *PollingTimer[Optional]* | • 0-525600 (min)<br>• 0:disabled |
| *ConnectionRetryTimers[Optional]* | • See QmiSwiAvmsSetSettingsConnectionRetryTimers for more information |
| *APNInfo[Optional]* | • See QmiSwiAvmsSetSettingsAPNInfo for more information |
| *NotifStore[Optional]* | • 0 : Disabled.<br>• 1 : Enabeld. |
| *PeroidInfo[Optional]* | • See QmiSwiAvmsSetSettingsPeriodInfo 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

#### 9.28.3.15 typedef struct _SLQSAVMSSettings SLQSAVMSSettings

#### 9.28.3.16 typedef struct _SLQSAVMSSettingsAPNInfo SLQSAVMSSettingsAPNInfo

#### 9.28.3.17 typedef struct _SLQSAVMSSettingsConnectionRetryTimers SLQSAVMSSettingsConnectionRetryTimers

#### 9.28.3.18 typedef struct _SLQSAVMSSettingsPeriodsInfo SLQSAVMSSettingsPeriodsInfo

### 9.28.4 Function Documentation

#### 9.28.4.1 ULONG SLQSAVMSGetSessionInfo ( QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp * *pBinaryUpdteSessionInfo*, QmiSwiAvmsEventReportConfigResp * *pAVMSConfig*, QmiSwiAvmsEventReportNotificationResp * *pNotif*, QmiSwiAvmsEventReportPackageIDResp * *pPackageID* )

Returns information related to the current AVMS session.

**Parameters**

| | |
|---|---|
| *pBinaryUpdte-SessionInfo[OU-T]* | • See QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp for more information. |
| *pAVMSConfig[-OUT]* | • See QmiSwiAvmsEventReportConfigResp for more information. |
| *pNotif[OUT]* | • See QmiSwiAvmsEventReportNotificationResp for more information. |
| *pPackageID[OU-T]* | • See QmiSwiAvmsEventReportPackageIDResp 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

#### 9.28.4.2 ULONG SLQSAVMSGetSettings ( SLQSAVMSSettings * *settings* )

#### 9.28.4.3 ULONG SLQSAVMSSelection ( BYTE *selection*, ULONG * *pDeferTime*, BYTE * *pClientPerfOperationFlag*, BYTE * *pPackageID*, WORD * *pRejectReason* )

Sends the specified AVMS selection for the current network initiated session.

**Parameters**

| | |
|---|---|
| *selection[IN]* | • AVMS User Selection.<br>– 0x01 - Accept.<br>– 0x02 - Reject.<br>– 0x03 - Defer. |
| *pDeferTime[IN]* | • Defer time in minutes. A value of 0 will cause the prompt to be resent immediately. |

| | |
|---|---|
| *pClientPerf-OperationFlag[I-N]* | • 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]* | • Package ID (relevant to Application binary package)<br>– Package ID of the application binary that the client is referring to for its selection. If this field is absent, the modem will assume the package of the most recent AVMS_EVENT_ID notification. |
| *pRejectReason[-IN]* | • Reject reason. |

**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

#### 9.28.4.4 ULONG SLQSAVMSSetEventReport ( )

Sets AVMS Event Report.

**Parameters**

| | |
|---|---|
| *NULL* | |

**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

#### 9.28.4.5 ULONG SLQSAVMSSetSettings ( SLQSAVMSSetSettingsReq *Request* )

Sets AVMS settings requested.

**Parameters**

| | |
|---|---|
| *Request[IN]* | • See SLQSAVMSSetSettingsReq 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

#### 9.28.4.6 ULONG SLQSAVMSSetSettingsIgnoreAutoReboot ( SLQSAVMSSetSettingsReq *Request* )

Sets AVMS settings requested without Auto Reboot.

**Parameters**

| | |
|---|---|
| *Request[IN]* | • See SLQSAVMSSetSettingsReq 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

#### 9.28.4.7 ULONG SLQSAVMSStartSession ( BYTE *sessionType,* ULONG * *pSessionResponse* )

Starts an AVMS session.

**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. |

| | |
|---|---|
| *pSession-Response[OUT]* | • Session Response<br>– 0x01 -Available.<br>– 0x02 -Not Available.<br>– 0x03 -Check Timed Out |

**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

#### 9.28.4.8 ULONG SLQSAVMSStopSession ( BYTE *sessiontype*, BYTE * *pSessionRep* )

Stop an ongoing AVMS session.

**Parameters**

| | |
|---|---|
| *session[IN]* | • Session<br>– 0x01 - FOTA, to check availability of FW Update.<br>– 0xFF - Cancel any active OMADM session. |
| *pSessionRep[OUT]* | • Session<br>– 0x01 - FOTA session suspended.<br>– 0xFF - Suspend ongoing FOTA session or stop any other active AVMS session. |

**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 qaGobiApiSwiDms.h File Reference

SWI DMS API function prototypes.

### Data Structures

- struct SwiDmsSLQSGetUsbNetNum
- struct MTUSizeReq
- struct mtuSize3gppTlv
- struct hrpdMTUSizeTlv
- struct ehrpdMTUSizeTlv
- struct usbMTUSizeTlv
- struct SwiDmsGetMTUResp

### Functions

- ULONG SLQSSwiDmsGetUsbNetNum (SwiDmsSLQSGetUsbNetNum *pSwiDmsSLQSGetUsbNetNum)
- ULONG SLQSSwiDmsSetUsbNetNum (BYTE nUsbNetNum)
- ULONG SLQSSwiDmsSetMTU (MTUSizeReq *pMTUSizeReq)
- ULONG SLQSSwiDmsGetMTU (SwiDmsGetMTUResp *pSwiDMSGetMTUResp)

## 9.29.1 Detailed Description

SWI DMS API function prototypes.

## 9.29.2 Function Documentation

### 9.29.2.1 ULONG SLQSSwiDmsGetMTU ( SwiDmsGetMTUResp * *pSwiDMSGetMTUResp* )

Returns the preferred MTU Size of the 3GPP, HRPD, EHRPD and USB descriptor Interfaces.

**Parameters**

| | |
|---|---|
| *pSwiDMSGetMTUResp[OUT]* | See SwiDmsGetMTUResp for more information of the input structure |

**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 ULONG SLQSSwiDmsGetUsbNetNum ( SwiDmsSLQSGetUsbNetNum * *pSwiDmsSLQSGetUsbNetNum* )

Used by the control point to get the number of usb networks (QMAP configuration).

**Parameters**

| | |
|---|---|
| *pSwiDmsSLQSGetUsbNetNum[OUT]* | • See SwiDmsSLQSGetUsbNetNum for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.29.2.3 ULONG SLQSSwiDmsSetMTU ( MTUSizeReq * *pMTUSizeReq* )

Sets the preferred MTU size for 3GPP, HRPD, EHRPD and USB Descriptor

**Parameters**

| | |
|---|---|
| *pMTUSizeReq[IN]* | See MTUSizeReq for more information of the input structure |

**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. New MTU size effective after modem reboot.

#### 9.29.2.4 ULONG SLQSSwiDmsSetUsbNetNum ( BYTE *nUsbNetNum* )

Used by the control point to set the number of usb networks (QMAP configuration).

**Parameters**

| | |
|---|---|
| *nUsbNetNum[IN]* | • value for usb net numbers<br>• range is 0-8 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.30 qaGobiApiSwiOmadms.h File Reference

SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6.

### Data Structures

- struct _SLQSOMADMSessionInfo
- struct _SLQSOMADMSettingsReqParams
- struct _SLQSOMADMSettings
- struct _SLQSOMADMSettingsReqParams3

### Typedefs

- typedef struct _SLQSOMADMSessionInfo SLQSOMADMSessionInfo
- typedef struct _SLQSOMADMSettingsReqParams SLQSOMADMSettingsReqParams
- typedef struct _SLQSOMADMSettings SLQSOMADMSettings
- typedef struct _SLQSOMADMSettingsReqParams3 SLQSOMADMSettingsReqParams3

### Functions

- ULONG SLQSOMADMStartSession (ULONG sessionType)
- ULONG SLQSOMADMCancelSession (ULONG session)
- ULONG SLQSOMADMGetSessionInfo (ULONG *pSessionType, SLQSOMADMSessionInfo *pResp)
- ULONG SLQSOMADMSendSelection (ULONG selection)
- ULONG SLQSOMADMGetSettings (ULONG *pbOMADMEnabled, ULONG *pbFOTAdownload, ULONG *pbFOTAUpdate)
- ULONG SLQSOMADMSetSettings (ULONG bFOTAdownload, ULONG bFOTAUpdate)
- ULONG SLQSOMADMSetSettings2 (SLQSOMADMSettingsReqParams *pSLQSOMADMSettingsReqParams)
- ULONG SLQSOMADMGetSettings2 (SLQSOMADMSettings *pSLQSOMADMSettings)
- ULONG SLQSOMADMStartSession2 (ULONG sessionType, ULONG *pFwAvailability)
- ULONG SLQSOMADMSendSelection2 (ULONG selection, ULONG *pDeferTime, ULONG *pRejectReason)
- ULONG SLQSOMADMSetSettings3 (SLQSOMADMSettingsReqParams3 *pSLQSOMADMSettingsReqParams3)

## 9.30.1 Detailed Description

SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6.

## 9.30.2 Typedef Documentation

### 9.30.2.1 typedef struct _SLQSOMADMSessionInfo SLQSOMADMSessionInfo

Structure containing the OMA DM Session Info returned by the device. Also used as input parameter to specify the size of variable parameters. (ref. notes)

**Parameters**

| | |
|---|---|
| *pStatus* | • 1 Byte parameter indicating status(optional)<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 |
| *pUpdate-CompleteStatus* | • 2 byte parameter indicating Update Complete Status(optional)<br>– See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h Update Complete Status |
| *pSeverity* | • 1 byte parameter indicating severity(optional)<br>– 0x01 - Mandatory<br>– 0x02 - Optional |
| *pSourceLength* | • 2 byte parameter indicating Length of Vendor Name String in Bytes.(optional) |
| *pSource* | • Variable length parameter indicating Vendor Name in ASCII(optional) |
| *pPkgName-Length* | • 2 byte parameter indicating Length of Package Name String in Bytes.(optional) |
| *pPkgName* | • Variable length parameter indicating Package Name in ASCII(optional) |
| *pPkgDesc-Length* | • 2 byte parameter indicating Length of Package Description String in Bytes.(optional) |
| *pPkgDescription* | • Variable length parameter indicating Package Description in ASCII(optional) |
| *pDateLength* | • 2 byte parameter indicating Length of Package Description String in Bytes.(optional) |
| *pDate* | • Variable length parameter indicating Package Description in ASCII |
| *pTimeLength* | • 2 byte parameter indicating Length of Time String in Bytes.(optional) |
| *pTime* | • Variable length parameter indicating Time String in ASCII(optional) |

| | |
|---|---|
| *pSessionType* | • 1 byte parameter reflects the last session started for Sprint(optional)<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 |
| *pSessionState* | • 1 byte parameter indicating session state(optional)<br>– 0x01 - idle<br>– 0x02 - active<br>– 0x03 - pending |
| *pRetryCount* | • 1 byte parameter indicating retries left count(optional)<br>– valid values 0 to 6 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

As input parameter the members pSourceLength, pPkgNameLength pPkgDescLength, pDateLength, pTimeLength have to be specified. These should contain the initialized size of pSource, pPkgName, pPkgDescription, pDate, pTime respectively.

#### 9.30.2.2 typedef struct _SLQSOMADMSettings SLQSOMADMSettings

Structure containing the OMA DM settings retrieved from the device

**Parameters**

| | |
|---|---|
| *pOMADMEnabled[OUT]* | • 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. |

| | |
|---|---|
| *pFOTAdownload[OUT]* | • 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>• function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem. |
| *pFOTAUpdate[OUT]* | • 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. |
| *pAutosdm[OUT]* | • 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. |
| *pFwAutoCheck[OUT]* | • 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. |

**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 typedef struct _SLQSOMADMSettingsReqParams SLQSOMADMSettingsReqParams

Structure containing the OMA DM settings to be set on the device

**Parameters**

| | |
|---|---|
| *FOTAdownload* | • 1 Byte parameter indicating support for FOTA Automatic download<br>– 0x00 - Firmware auto download FALSE<br>– 0x01 - Firmware auto download TRUE |
| *FOTAUpdate* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware auto update FALSE<br>– 0x01 - Firmware auto update TRUE |
| *pAutosdm[IN]* | • Optional 1 byte parameter indicating OMA Automatic UI Alert Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled Accept<br>– 0x02 - Enabled Reject |

**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 typedef struct _SLQSOMADMSettingsReqParams3 SLQSOMADMSettingsReqParams3

Structure containing the OMA DM settings to be set on the device

**Parameters**

| | |
|---|---|
| *FOTAdownload* | • 1 Byte parameter indicating support for FOTA Automatic download<br>– 0x00 - Firmware auto download FALSE<br>– 0x01 - Firmware auto download TRUE |
| *FOTAUpdate* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware auto update FALSE<br>– 0x01 - Firmware auto update TRUE |
| *pAutosdm[IN]* | • Optional 1 byte parameter indicating OMA Automatic UI Alert Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled Accept<br>– 0x02 - Enabled Reject |
| *pFwAutoCheck[IN]* | • Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.30.3 Function Documentation

#### 9.30.3.1 ULONG SLQSOMADMCancelSession ( ULONG *session* )

Cancels an ongoing OMA-DM session.

**Parameters**

| | |
|---|---|
| *session[IN]* | • Session<br>– 0x01 - FOTA, to check availability of FW Update<br>– 0xFF - Cancel any active OMADM session |

**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

#### 9.30.3.2 ULONG SLQSOMADMGetSessionInfo ( ULONG * *pSessionType,* SLQSOMADMSessionInfo * *pResp* )

Returns information related to the current (or previous if no session is active) OMA-DM session.

**Parameters**

| | |
|---|---|
| *SessionType[IN]* | • Session type<br>– 0x01 - FOTA<br>– 0xFF - Any active OMADM session. If none active, then previous OMADM session |
| *pResp[IN/OUT]* | • See SLQSOMADMSessionInfo 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

#### 9.30.3.3 ULONG SLQSOMADMGetSettings ( ULONG * *pbOMADMEnabled,* ULONG * *pbFOTAdownload,* ULONG * *pbFOTAUpdate* )

Returns the OMA-DM settings.

**Parameters**

| | |
|---|---|
| *pbOMADMEnabled[OUT]* | • Device 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 |
| *pbFOTAdownload[OUT]* | • Firmware AutoDownload<br>– 0x00 - Firmware auto download FALSE<br>– 0x01 - Firmware autod ownload TRUE<br>– 0x02 - Automatically start downloading, while not roaming<br>– 0x03 - Automatically reject download<br>– 0x04 - Automatically reject download with “Enterprise Reject Policy” |
| *pbFOTAUpdate[OUT]* | • Firmware AutoUpdate<br>– 0x00 - Firmware auto update FALSE<br>– 0x01 - Firmware auto update TRUE |

**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

#### 9.30.3.4 ULONG SLQSOMADMGetSettings2 ( SLQSOMADMSettings * *pSLQSOMADMSettings* )

Retrieves the OMA-DM settings from the device.

**Parameters**

| | |
|---|---|
| *SLQSOMADM-SettingsReq-Params* | • See SLQSOMADMSettings 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

#### 9.30.3.5 ULONG SLQSOMADMSendSelection ( ULONG *selection* )

Sends the specified OMA-DM selection for the current network initiated session.

**Parameters**

| | |
|---|---|
| *selection[IN]* | • OMA-DM NIA Selection<br>– 0x01 - Accept<br>– 0x02 - Reject<br>– 0x03 - Defer |

**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

#### 9.30.3.6 ULONG SLQSOMADMSendSelection2 ( ULONG *selection*, ULONG * *pDeferTime*, ULONG * *pRejectReason* )

Sends the specified OMA-DM selection for the current network initiated session.

**Parameters**

| | |
|---|---|
| *selection[IN]* | • OMA-DM NIA Selection<br>– 0x01 - Accept<br>– 0x02 - Reject<br>– 0x03 - Defer |

| | |
|---|---|
| *pDeferTime[IN]* | • 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]* | • 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. |

**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

#### 9.30.3.7 ULONG SLQSOMADMSetSettings ( ULONG *bFOTAdownload,* ULONG *bFOTAUpdate* )

Sets the OMA-DM settings requested.

**Parameters**

| | |
|---|---|
| *bFOTAdownload[IN]* | • Firmware Auto 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” |
| *bFOTAUpdate[IN]* | • Firmware Auto 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 |

**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

#### 9.30.3.8 ULONG SLQSOMADMSetSettings2 ( SLQSOMADMSettingsReqParams * *pSLQSOMADMSettingsReqParams* )

Sets the settings related to OMADM. These settings are saved on the modem across power cycles.

**Parameters**

| | |
|---|---|
| *pSLQSOMADM-SettingsReq-Params[IN]* | • See SLQSOMADMSettingsReqParams for more information |

**Note**

Timeout: 20 seconds

#### 9.30.3.9 ULONG SLQSOMADMSetSettings3 ( SLQSOMADMSettingsReqParams3 * *pSLQSOMADMSettingsReqParams3* )

Sets the settings related to OMADM. These settings are saved on the modem across power cycles.

**Parameters**

| | |
|---|---|
| *SLQSOMADM-SettingsReq-ParamsExt[IN]* | • See SLQSOMADMSettingsReqParamsExt for more information |

**Note**

Timeout: 20 seconds

#### 9.30.3.10 ULONG SLQSOMADMStartSession ( ULONG *sessionType* )

Starts an OMA-DM session.

**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 |

**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

#### 9.30.3.11 ULONG SLQSOMADMStartSession2 ( ULONG *sessionType*, ULONG * *pFwAvailability* )

Starts an OMA-DM session.

**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 |
| *pFwAvailability[-OUT]* | • 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 |

**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

## 9.31 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

### 9.31.1 Detailed Description

Network Access Service API Band Classes table.

### 9.31.2 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

#### 9.31.2.1 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

## 9.32 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

### 9.32.1 Detailed Description

Call Control Return Reasons table.

### 9.32.2 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

## 9.33 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

### 9.33.1 Detailed Description

Wireless Data Service Call End Reasons.

### 9.33.2 Call end reason codes (Code - Reason)

#### 9.33.2.1 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

#### 9.33.2.2 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

#### 9.33.2.3 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

#### 9.33.2.4 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

#### 9.33.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

#### 9.33.2.6 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

#### 9.33.2.7 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

#### 9.33.2.8 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

#### 9.33.2.9 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

#### 9.33.2.10 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

#### 9.33.2.11 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

#### 9.33.2.12 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

## 9.34 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

### 9.34.1 Detailed Description

Carrier Codes table.

### 9.34.2 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

## 9.35 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

### Macros

- #define __GOBI_API_CODING_SCHEME_H__

### 9.35.1 Detailed Description

Data Coding Scheme.

### 9.35.2 S1

#### 9.35.2.1 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

### 9.35.3 Coding Group Bits 7..4(0001)

#### 9.35.3.1 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.

### 9.35.4 Coding Group Bits 7..4(0010)

#### 9.35.4.1 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

### 9.35.5 Coding Group Bits 7..4(0011)

#### 9.35.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit
  default alphabet, with unspecified handling at the MS

### 9.35.6 Coding Group Bits 7..4(01xx)

#### 9.35.6.1 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

### 9.35.7 Coding Group Bits 7..4(1001)

#### 9.35.7.1 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

### 9.35.8 Coding Group Bits 7..4(1010..1101)

#### 9.35.8.1 Reserved coding groups

### 9.35.9 Coding Group Bits 7..4(1110)

#### 9.35.9.1 Defined by the WAP Forum

### 9.35.10 Coding Group Bits 7..4 (1111)

#### 9.35.10.1 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

### 9.35.11 Macro Definition Documentation

#### 9.35.11.1 #define __GOBI_API_CODING_SCHEME_H__

## 9.36 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

### 9.36.1 Detailed Description

Position Determination Service API GPS Capability Codes.

### 9.36.2 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

## 9.37 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

### 9.37.1 Detailed Description

Device Management Service API Power Modes table.

### 9.37.2 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

## 9.38 qaGobiApiTableRadioInterfaces.h File Reference

Network Access Service API Radio Interfaces table.

### 9.38.1 Detailed Description

Network Access Service API Radio Interfaces table.

### 9.38.2 Radio interface

#### 9.38.2.1 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

## 9.39 qaGobiApiTableRegionCodes.h File Reference

Region Codes table.

### 9.39.1 Detailed Description

Region Codes table.

### 9.39.2 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

## 9.40 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

### 9.40.1 Detailed Description

Voice Service Options.

### 9.40.2 Service Option codes (Code - Reason)

#### 9.40.2.1 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

## 9.41 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

### 9.41.1 Detailed Description

Voice Supplementary Service Information Classes.

### 9.41.2 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

## 9.42 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

### 9.42.1 Detailed Description

Swi Audio related tables.

### 9.42.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

### 9.42.3 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

## 9.43 qaGobiApiTableSwiOMADMSessionStatus.h File Reference

Session Status table.

### 9.43.1 Detailed Description

Session Status table.

### 9.43.2 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

## 9.44 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

### 9.44.1 Detailed Description

Update Complete Status table.

### 9.44.2 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

## 9.45 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

### 9.45.1 Detailed Description

Voice Service Call and supplementary services end reasons.

### 9.45.2 Voice Call and supplementary services end reason codes (Code - Reason)

#### 9.45.2.1 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

#### 9.45.2.2 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

#### 9.45.2.3 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

#### 9.45.2.4 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

#### 9.45.2.5 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

#### 9.45.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

#### 9.45.2.7 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

#### 9.45.2.8 reject causes

- 237 - Service option is not supported

#### 9.45.2.9 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

## 9.46 qaGobiApiTmd.h File Reference

Thermal Mitigation Device API function prototypes.

### Data Structures

- struct mitigationDevList
- struct TmdGetMitigationDevListResp
- struct TmdGetMitigationLvlReq
- struct TmdGetMitigationLvlResp
- struct TmdRegNotMitigationLvlReq
- struct TmdDeRegNotMitigationLvlReq
- struct TmdMitigationLvlIndReq

### Macros

- #define MAX_MITIGATION_DEV_LIST_LEN 255
- #define MAX_MITIGATION_DEV_ID_LEN 255

### Functions

- ULONG SLQSTmdGetMitigationDevList (TmdGetMitigationDevListResp *pTmdGetMitigationDevListResp)
- ULONG SLQSTmdGetMitigationLvl (TmdGetMitigationLvlReq *pTmdGetMitigationLvlReq, TmdGetMitigationLvlResp *pTmdGetMitigationLvlResp)
- ULONG SLQSTmdRegNotMitigationLvl (TmdRegNotMitigationLvlReq *pTmdRegNotMitigationLvlReq)
- ULONG SLQSTmdDeRegNotMitigationLvl (TmdDeRegNotMitigationLvlReq *pTmdDeRegNotMitigationLvlReq)

### 9.46.1 Detailed Description

Thermal Mitigation Device API function prototypes.

### 9.46.2 Macro Definition Documentation

#### 9.46.2.1 #define MAX_MITIGATION_DEV_ID_LEN 255

#### 9.46.2.2 #define MAX_MITIGATION_DEV_LIST_LEN 255

### 9.46.3 Function Documentation

#### 9.46.3.1 ULONG SLQSTmdDeRegNotMitigationLvl ( TmdDeRegNotMitigationLvlReq * *pTmdDeRegNotMitigationLvlReq* )

Deregisters the notification for mitigation device level changes.

**Parameters**

| | |
|---|---|
| *pTmdDeRegNotMitigationLvlReq[IN]* | • See TmdDeRegNotMitigationLvlReq for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.46.3.2 ULONG SLQSTmdGetMitigationDevList ( TmdGetMitigationDevListResp * *pTmdGetMitigationDevListResp* )

Used by the control point to gets the list of mitigation devices from the remote endpoint.

**Parameters**

| | |
|---|---|
| *pTmdGetMitigationDevListResp[OUT]* | • See TmdGetMitigationDevListResp for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.46.3.3 ULONG SLQSTmdGetMitigationLvl ( TmdGetMitigationLvlReq * *pTmdGetMitigationLvlReq*, TmdGetMitigationLvlResp * *pTmdGetMitigationLvlResp* )

Gets the thermal mitigation level for the mitigation device

**Parameters**

| | |
|---|---|
| *pTmdGetMitigationLvlReq[IN]* | • See TmdGetMitigationLvlReq for more information |
| *pTmdGetMitigationLvlResp[OUT]* | • See TmdGetMitigationLvlResp for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.46.3.4 ULONG SLQSTmdRegNotMitigationLvl ( TmdRegNotMitigationLvlReq * *pTmdRegNotMitigationLvlReq* )

Registers for notification of mitigation device level changes.

**Parameters**

| | |
|---|---|
| *pTmdRegNotMitigationLvlReq[IN]* | • See TmdRegNotMitigationLvlReq for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.47 qaGobiApiUim.h File Reference

Uim Service API function prototypes.

### Data Structures

- struct UIMPowerDownReq
- struct fileInfo
- struct UIMRefreshEvent
- struct appStatus
- struct slotInfo
- struct cardStatus
- struct hotSwapStatus
- struct UIMGetCardStatusResp
- struct UIMSessionInformation
- struct setPINProtection
- struct UIMSetPinProtectionReq
- struct remainingRetries
- struct encryptedPIN1
- struct UIMPinResp
- struct verifyUIMPIN
- struct UIMVerifyPinReq
- struct changeUIMPIN
- struct UIMChangePinReq
- struct unblockUIMPIN
- struct UIMUnblockPinReq
- struct UIMEventRegisterReqResp
- struct UIMRefreshOKReq
- struct registerRefresh
- struct UIMRefreshRegisterReq
- struct UIMRefreshCompleteReq
- struct UIMRefreshGetLastEventResp
- struct UIMRefreshGetLastEventReq
- struct UIMGetFileAttributesReq
- struct cardResult
- struct fileAttributes
- struct UIMGetFileAttributesResp
- struct depersonalizationInformation
- struct UIMDepersonalizationReq
- struct UIMDepersonalizationResp

- struct authenticationData
- struct UIMAuthenticateReq
- struct authenticateResult
- struct UIMAuthenticateResp
- struct readResult
- struct readTransparentInfo
- struct UIMReadTransparentReq
- struct UIMReadTransparentResp
- struct UIMPowerUpReq
- struct UIMSlotStatus
- struct UIMSlotsStatus
- struct UIMGetSlotsStatusResp
- struct UIMSwitchSlotReq
- struct personalizationStatus
- struct UIMGetConfigurationReq
- struct UIMGetConfigurationResp

## Macros

- #define MAX_DESCRIPTION_LENGTH 255
- #define MAX_CONTENT_LENGTH 1024
- #define MAX_NO_OF_APPLICATIONS 10
- #define MAX_NO_OF_SLOTS 5
- #define MAX_PUK_LENGTH 8
- #define MAX_PATH_LENGTH 255
- #define MAX_ICCID_LENGTH 255
- #define MAX_SLOTS_STATUS 255
- #define MAX_ACTIVE_PERS_FEATURES 12

## Functions

- ULONG SLQSUIMReset ()
- ULONG SLQSUIMPowerDown (UIMPowerDownReq *pUIMPowerDownReq)
- ULONG SLQSUIMGetCardStatus (UIMGetCardStatusResp *pUIMGetCardStatusResp)
- ULONG SLQSUIMSetPinProtection (UIMSetPinProtectionReq *pUIMSetPinProtectionReq, UIMPinResp *pUIMSetPinProtectionResp)
- ULONG SLQSUIMVerifyPin (UIMVerifyPinReq *pUIMVerifyPinReq, UIMPinResp *pUIMVerifyPinResp)
- ULONG SLQSUIMChangePin (UIMChangePinReq *pUIMChangePinReq, UIMPinResp *pUIMChangePinResp)
- ULONG SLQSUIMUnblockPin (UIMUnblockPinReq *pUIMUnblockPinReq, UIMPinResp *pUIMUnblockPinResp)
- ULONG SLQSUIMEventRegister (UIMEventRegisterReqResp *pUIMEventRegisterReqResp)
- ULONG SLQSUIMRefreshOK (UIMRefreshOKReq *pUIMRefreshOKReq)
- ULONG SLQSUIMRefreshRegister (UIMRefreshRegisterReq *pUIMRefreshRegisterReq)
- ULONG SLQSUIMRefreshComplete (UIMRefreshCompleteReq *pUIMRefreshCompleteReq)
- ULONG SLQSUIMRefreshGetLastEvent (UIMRefreshGetLastEventReq *pUIMRefreshGetLastEventReq, UIMRefreshGetLastEventResp *pUIMRefreshGetLastEventResp)
- ULONG SLQSUIMGetFileAttributes (UIMGetFileAttributesReq *pUIMGetFileAttributesReq, UIMGetFileAttributesResp *pUIMGetFileAttributesResp)
- ULONG SLQSUIMDepersonalization (UIMDepersonalizationReq *pUIMDepersonilizationReq, UIMDepersonalizationResp *pUIMDepersonilizationResp)
- ULONG SLQSUIMAuthenticate (UIMAuthenticateReq *pUIMAuthenticateReq, UIMAuthenticateResp *pUIMAuthenticateResp)

- ULONG SLQSUIMReadTransparent (UIMReadTransparentReq *pUIMReadTransparentReq, UIMReadTransparentResp *pUIMReadTransparentResp)
- ULONG SLQSUIMPowerUp (UIMPowerUpReq *pUIMPowerUpReq)
- ULONG SLQSUIMGetSlotsStatus (UIMGetSlotsStatusResp *pResp)
- ULONG SLQSUIMSwitchSlot (UIMSwitchSlotReq *pReq)
- ULONG SLQSUIMGetConfiguration (UIMGetConfigurationReq *pUIMGetConfigurationReq, UIMGetConfigurationResp *pUIMGetConfigurationResp)

### 9.47.1 Detailed Description

Uim Service API function prototypes.

### 9.47.2 Macro Definition Documentation

#### 9.47.2.1 #define MAX_ACTIVE_PERS_FEATURES 12

#### 9.47.2.2 #define MAX_CONTENT_LENGTH 1024

#### 9.47.2.3 #define MAX_DESCRIPTION_LENGTH 255

#### 9.47.2.4 #define MAX_ICCID_LENGTH 255

#### 9.47.2.5 #define MAX_NO_OF_APPLICATIONS 10

#### 9.47.2.6 #define MAX_NO_OF_SLOTS 5

#### 9.47.2.7 #define MAX_PATH_LENGTH 255

#### 9.47.2.8 #define MAX_PUK_LENGTH 8

#### 9.47.2.9 #define MAX_SLOTS_STATUS 255

### 9.47.3 Function Documentation

#### 9.47.3.1 ULONG SLQSUIMAuthenticate ( UIMAuthenticateReq * *pUIMAuthenticateReq*, UIMAuthenticateResp * *pUIMAuthenticateResp* )

This API executes the authentication algorithm on the card.

**Parameters**

| | |
|---|---|
| *pUIMAuthenticateReq[IN]* | • See UIMAuthenticateReq for more information. |
| *pUIMAuthenticateResp[OUT]* | • See UIMAuthenticateResp 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: 30 Secs

```
This API executes a security command on the card that depends on
the card type.
The response contains the status code received from the card
(SW1 and SW2) when the card responded to the read request.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMAuthenticateCallback
```

### 9.47.3.2 ULONG SLQSUIMChangePin ( UIMChangePinReq * *pUIMChangePinReq*, UIMPinResp * *pUIMChangePinResp* )

This API changes the value of the specified PIN.

**Parameters**

| | |
|---|---|
| *pUIMChange-PinReq[IN]* | • See UIMChangePinReq for more information. |
| *pUIMChange-PinResp[OUT]* | • See UIMPinResp 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: 30 Secs

```
This API changes the value of the specified PIN.
The application must pass both the new and the old values of the
PIN to complete the operation
The same PIN can be used by multiple sessions (i.e., the PIN is
shared between GSM and RUIM in an ICC card).
The PIN is automatically set for all the sessions when the API
is executed.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMChangePinCallback.
```

### 9.47.3.3 ULONG SLQSUIMDepersonalization ( UIMDepersonalizationReq * *pUIMDepersonilizationReq*, UIMDepersonalizationResp * *pUIMDepersonilizationResp* )

This API de-activates or unblocks the personalization on the phone.

**Parameters**

| | |
|---|---|
| *pUIM-Depersonilization-Req[IN]* | • See UIMDepersonalizationReq for more information. |
| *pUIM-Depersonilization-Resp[OUT]* | • See UIMDepersonalizationResp 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: 30 Secs

```
This API deactivates or unblocks the personalization on the phone.
Each feature can be deactivated/unblocked independently of the
other features.
```

#### 9.47.3.4 ULONG SLQSUIMEventRegister ( UIMEventRegisterReqResp * *pUIMEventRegisterReqResp* )

This API Registers for event notifications from the card.

**Parameters**

| | |
|---|---|
| *pUIMEvent-RegisterReq-Resp[IN/OUT]* | • See UIMEventRegisterReqResp 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: 30 Secs

```
This function registers for event notifications from the card.
The client must verify the mask in the response to determine
which events were registered successfully. Events not supported
correctly are not registered. The client can deregister from
all event notifications by passing "0x00000000" bitmask in the
request.
```

#### 9.47.3.5 ULONG SLQSUIMGetCardStatus ( UIMGetCardStatusResp * *pUIMGetCardStatusResp* )

This API retrieves the current status of the card.

**Parameters**

| | |
|---|---|
| *pUIMGetCard-StatusResp[OU-T]* | • See UIMGetCardStatusResp 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: 30 Secs

```
This function retrieves the current status of the card(activated)
and the status of all applications available on the card.
The function also returns support information for the hot-swap
feature and the status of the switch used to detect a card
removal/insertion.

Please use \ref SLQSUIMGetSlotsStatus to retrieves active and
inactivate SIMs status.
```

#### 9.47.3.6 ULONG SLQSUIMGetConfiguration ( UIMGetConfigurationReq * *pUIMGetConfigurationReq*, UIMGetConfigurationResp * *pUIMGetConfigurationResp* )

This API Gets the modem configuration for the UIM module.

**Parameters**

| | |
|---|---|
| *pUIMGet-Configuration-Req[IN]* | • See UIMGetConfigurationReq for more information. |
| *pUIMGet-Configuration-Resp[OUT]* | • See UIMGetConfigurationResp 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: 30 Secs

#### 9.47.3.7 ULONG SLQSUIMGetFileAttributes ( UIMGetFileAttributesReq * *pUIMGetFileAttributesReq*, UIMGetFileAttributesResp * *pUIMGetFileAttributesResp* )

This API retrieves the file attributes for any Elementary File(EF) or Dedicated File(DF) in the card and provides access by the path.

**Parameters**

| | |
|---|---|
| *pUIMGetFileAttributesReq[IN]* | • See UIMGetFileAttributesReq for more information. |
| *pUIMGetFileAttributesResp[OUT]* | • See UIMGetFileAttributesResp 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: 30 Secs

```
This API retrieves the file attributes for any Elementary File(EF)
or Dedicated File(DF) in the card and provides access by the path.
The response contains the status code received from the card
(SW1 and SW2) when the card responded to the select request.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMGetFileAttributesCallback.
```

#### 9.47.3.8 ULONG SLQSUIMGetSlotsStatus ( UIMGetSlotsStatusResp * *pResp* )

This API Retrieves the current of the physical and logical slots.

**Parameters**

| | |
|---|---|
| *pResp[OUT]* | • See UIMGetSlotsStatusResp 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: 30 Secs

#### 9.47.3.9 ULONG SLQSUIMPowerDown ( UIMPowerDownReq * *pUIMPowerDownReq* )

This API powers down the SIM card.

**Parameters**

| | |
|---|---|
| *pUIMPowerDownReq[IN]* | • See UIMPowerDownReq 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: 30 Secs

```
This function powers down the card.
This is usually performed when the phone is switched off or when
it is set to Airplane mode.
```

#### 9.47.3.10 ULONG SLQSUIMPowerUp ( UIMPowerUpReq * *pUIMPowerUpReq* )

This API powers up the SIM card.

**Parameters**

| | |
|---|---|
| *pUIMPowerUp-Req[IN]* | • See UIMPowerUpReq 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: 30 Secs

```
This function powers up the card.
This is usually performed when the phone is switched off or when
it is set to Airplane mode.
```

#### 9.47.3.11 ULONG SLQSUIMReadTransparent ( UIMReadTransparentReq * *pUIMReadTransparentReq,* UIMReadTransparentResp * *pUIMReadTransparentResp* )

This API executes the Read Transparent algorithm on the card.

**Parameters**

| | |
|---|---|
| *pUIMRead-Transparent-Req[IN]* | • See UIMReadTransparentReq for more information. |
| *pUIMRead-Transparent-Resp[OUT]* | • See UIMReadTransparentResp 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: 30 Secs

```
This API Provides read access to any transparent file in the
card and provides access by the path.
The response contains the status code received from the card
(SW1 and SW2) when the card responded to the read request.
The client can pass a token in the request to receive the result
in a subsequent QMI_UIM_READ_TRANSPARENT_IND indication.
```

#### 9.47.3.12 ULONG SLQSUIMRefreshComplete ( UIMRefreshCompleteReq * *pUIMRefreshCompleteReq* )

This API invoked when the client has finished the Refresh procedure.

**Parameters**

| | |
|---|---|
| *pUIMRefresh-CompleteReq[I-N]* | • See UIMRefreshCompleteReq 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: 30 Secs

```
This function is invoked when the client has finished the Refresh
procedure (has reread all the cached files) and communicates this
to the modem. This function enables the terminal response to be
sent to the card
```

#### 9.47.3.13 ULONG SLQSUIMRefreshGetLastEvent ( UIMRefreshGetLastEventReq * *pUIMRefreshGetLastEventReq*, UIMRefreshGetLastEventResp * *pUIMRefreshGetLastEventResp* )

This API provides the ability to retrieve the last refresh event.

**Parameters**

| | |
|---|---|
| *pUIMRefresh-GetLastEvent-Req[IN]* | • See UIMRefreshGetLastEventReq for more information. |
| *pUIMRefresh-GetLastEvent-Resp[OUT]* | • See UIMRefreshGetLastEventResp 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: 30 Secs

```
This function provides the ability to retrieve the last refresh
event. The event information is usually passed in as an indication
from the QMI to the application and is saved by the application at
that time. If the event information is not saved, the client can
retrieve the last refresh event.
Details regarding the Refresh procedure (i.e., the stages and
actions that an application must complete) are described in
document: 80-VM566-1(NAA Refresh High Level Guide )
```

#### 9.47.3.14 ULONG SLQSUIMRefreshOK ( UIMRefreshOKReq * *pUIMRefreshOKReq* )

This API Enables the client to indicate whether it is OK to start the Refresh procedure.

**Parameters**

| | |
|---|---|
| *pUIMRefreshO-KReq[IN]* | • Consist of parameters for SLQSUIMRefreshOK. Please see /ref UIMRefreshOKReq for details. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 30 Secs

```
This function enables the client to indicate whether it is OK to
start the Refresh procedure. This command is used only after a
refresh event is received, which indicates the need to vote.
```

#### 9.47.3.15 ULONG SLQSUIMRefreshRegister ( UIMRefreshRegisterReq * *pUIMRefreshRegisterReq* )

This API Registers for file change notifications triggered by the card.

**Parameters**

| | |
|---|---|
| *pUIMRefresh-RegisterReq[IN]* | • See UIMRefreshRegisterReq 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: 30 Secs

```
This function registers for file change notifications triggered
by the card. The client can specify a list of files. The client
is notified only when one of the files is modified by the Refresh
procedure. This function can be invoked multiple times for each
session type. If the function is invoked twice with the same session
type, the new values overwrite the previous values. The client can
also use this function to stop receiving indications of the refresh.
This API should be invoked prior to the invocation of the
SLQSUIMSetRefreshCallBack for the events to be registered.
```

#### 9.47.3.16 ULONG SLQSUIMReset ( )

This API resets the issuing control points state kept by the service.

**Parameters**

| | |
|---|---|
| *None* | |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 30 Secs

```
The list of events to which the client is registered is emptied.
The client must register again using the SLQSUIMEventReg API
to start receiving the events again. This would mean that the
callback registrations would be reset after this API.
```

#### 9.47.3.17 ULONG SLQSUIMSetPinProtection ( UIMSetPinProtectionReq * *pUIMSetPinProtectionReq*, UIMPinResp * *pUIMSetPinProtectionResp* )

This API enables or disables the protection of the UIM contents by a specific PIN.

**Parameters**

| | |
|---|---|
| *pUIMSetPinProtectionReq[IN]* | • See UIMSetPinProtectionReq for more information. |
| *pUIMSetPinProtectionResp[OUT]* | • See UIMPinResp 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: 30 Secs

```
This API enables or disables the protection of UIM contents by a
specific PIN.
The same PIN can be used by multiple sessions (i.e., the PIN is
shared between GSM and RUIM in an ICC card).
The PIN is automatically set for all the sessions when the API
is executed.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMSetPinProtectionCallback indication.
```

#### 9.47.3.18 ULONG SLQSUIMSwitchSlot ( UIMSwitchSlotReq * *pReq* )

This API Switches the binding between a logical slot and a physical slot.

**Parameters**

| | |
|---|---|
| *pReq[IN]* | • See UIMSwitchSlotReq 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: 30 Secs

```
Please wait at least one second before this API call again.
```

#### 9.47.3.19 ULONG SLQSUIMUnblockPin ( UIMUnblockPinReq * *pUIMUnblockPinReq*, UIMPinResp * *pUIMUnblockPinResp* )

This API unblocks a blocked PIN using the PUK code.

**Parameters**

| | |
|---|---|
| *pUIMUnblock-PinReq[IN]* | • See UIMUnblockPinReq for more information. |
| *pUIMUnblock-PinResp[OUT]* | • See UIMPinResp 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: 30 Secs

```
This API unblocks a blocked PIN using the PUK code.
The client must pass PUK1 to unblock PIN1 or PUK2 to unblock PIN2.
The same PIN can be used by multiple sessions (i.e., the PIN is
shared between GSM and RUIM in an ICC card).
The PIN is automatically set for all the sessions when the API
is executed.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMUnblockPinCallback.
```

#### 9.47.3.20 ULONG SLQSUIMVerifyPin ( UIMVerifyPinReq * *pUIMVerifyPinReq*, UIMPinResp * *pUIMVerifyPinResp* )

This API verifies the PIN before the card content is accessed.

**Parameters**

| | |
|---|---|
| *pUIMVerifyPinReq[IN]* | • See UIMVerifyPinReq for more information. |
| *pUIMVerifyPinResp[OUT]* | • See UIMPinResp 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: 30 Secs

```
This API verifies the PIN before the card content is accessed.
The same PIN can be used by multiple sessions (i.e., the PIN is
shared between GSM and RUIM in an ICC card).
The PIN is automatically set for all the sessions when the API
is executed.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMVerifyPinCallback.
```

## 9.48 qaGobiApiVoice.h File Reference

Voice Service API function prototypes.

## Data Structures

- struct USSInfo
- struct UUSInfo
- struct CUGInfo
- struct calledPartySubAdd
- struct voiceCallRequestParams
- struct alphaIDInfo
- struct ccSUPSType
- struct voiceCallResponseParams
- struct callFwdTypeAndPlan
- struct voiceSetSUPSServiceReq
- struct voiceSetSUPSServiceResp
- struct airTimer
- struct roamTimer
- struct prefVoiceSO
- struct voiceSetConfigReq
- struct voiceSetConfigResp
- struct voiceAnswerCall
- struct CLIRResp
- struct voiceGetCLIRResp
- struct CLIPResp
- struct voiceGetCLIPResp
- struct voiceGetCallWaitInfo
- struct voiceGetCallBarringReq
- struct voiceGetCallBarringResp
- struct voiceGetCallFWReq
- struct callFWInfo
- struct callFWExtInfo
- struct getCallFWInfo
- struct getCallFWExtInfo
- struct voiceGetCallFWResp
- struct voiceSetCallBarringPwdInfo
- struct voiceSetCallBarringPwdResp
- struct callInfo
- struct remotePartyNum
- struct remotePartyName
- struct connectNumInfo
- struct diagInfo
- struct voiceCallInfoReq
- struct voiceCallInfoResp
- struct getAllCallInformation
- struct getAllCallRmtPtyNum
- struct getAllCallRmtPtyName
- struct allCallsUUSInfo
- struct allCallsAlphaIDInfo
- struct allCallsDiagInfo
- struct peerNumberInfo
- struct arrCallInfo
- struct arrRemotePartyNum
- struct arrRemotePartyName
- struct arrAlertingType
- struct arrUUSInfo
- struct arrSvcOption
- struct arrCallEndReason

- struct arrAlphaID
- struct arrConnectPartyNum
- struct arrDiagInfo
- struct arrCalledPartyNum
- struct arrRedirPartyNum
- struct arrAlertingPattern
- struct voiceGetAllCallInfo
- struct voiceManageCallsReq
- struct voiceManageCallsResp
- struct burstDTMFInfo
- struct DTMFLengths
- struct voiceBurstDTMFInfo
- struct voiceContDTMFinfo
- struct voiceStopContDTMFinfo
- struct voiceFlashInfo
- struct voiceSetPrefPrivacy
- struct voiceIndicationRegisterInfo
- struct DTMFInfo
- struct SUPSInfo
- struct newPwdData
- struct COLPResp
- struct COLRResp
- struct CNAPResp
- struct voiceGetConfigReq
- struct curAMRConfig
- struct voiceGetConfigResp
- struct voiceOrigUSSDNoWaitInfo
- struct voiceBindSubscriptionInfo
- struct voiceALSSetLineSwitchInfo
- struct voiceALSSelectLineInfo
- struct voiceGetCOLPResp
- struct voiceGetCOLRResp
- struct voiceGetCNAPResp
- struct USSResp
- struct USSDRespFNetwork

## Macros

- #define MAXUSSDLENGTH 182
- #define MAX_CALL_NO_LEN 81
- #define MAX_DESCRIPTION_LENGTH 255
- #define PASSWORD_LENGTH 4
- #define MAX_NO_OF_CALLS 20

## Enumerations

- enum serviceClassInformation {
  VOICE_SUPS_SRV_CLASS_NONE = 0x00,
  VOICE_SUPS_SRV_CLASS_VOICE = 0x01,
  VOICE_SUPS_SRV_CLASS_DATA = 0x02,
  VOICE_SUPS_SRV_CLASS_FAX = 0x04,
  VOICE_SUPS_SRV_CLASS_SMS = 0x08,
  VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC = 0x10,
  VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC = 0x20,
  VOICE_SUPS_SRV_CLASS_PACKETACCESS = 0x40,
  VOICE_SUPS_SRV_CLASS_PADACCESS = 0x80 }

### Functions

- ULONG OriginateUSSD (BYTE *pInfo)
- ULONG AnswerUSSD (BYTE *pInfo)
- ULONG CancelUSSD ()
- ULONG SLQSVoiceDialCall (voiceCallRequestParams *pCallRequestParams, voiceCallResponseParams *pCallResponseParams)
- ULONG SLQSVoiceEndCall (BYTE *pCallId)
- ULONG SLQSVoiceSetSUPSService (voiceSetSUPSServiceReq *pVoiceSetSUPSServiceReq, voiceSetSUPSServiceResp *pVoiceSetSUPSServiceResp)
- ULONG SLQSVoiceSetConfig (voiceSetConfigReq *pVoiceSetConfigReq, voiceSetConfigResp *pVoiceSetConfigResp)
- ULONG SLQSVoiceAnswerCall (voiceAnswerCall *pVoiceAnswerCall)
- ULONG SLQSVoiceGetCLIR (voiceGetCLIRResp *pVoiceGetCLIRResp)
- ULONG SLQSVoiceGetCLIP (voiceGetCLIPResp *pVoiceGetCLIPResp)
- ULONG SLQSVoiceGetCallWaiting (voiceGetCallWaitInfo *pVoiceGetCallWaitInfo)
- ULONG SLQSVoiceGetCallBarring (voiceGetCallBarringReq *pVoiceGetCallBarringReq, voiceGetCallBarringResp *pVoiceGetCallBarringResp)
- ULONG SLQSVoiceGetCallForwardingStatus (voiceGetCallFWReq *pVoiceGetCallFWReq, voiceGetCallFWResp *pVoiceGetCallFWResp)
- ULONG SLQSVoiceSetCallBarringPassword (voiceSetCallBarringPwdInfo *pVoiceSetCallBarringPwdInfo, voiceSetCallBarringPwdResp *pSetCallBarringPwdResp)
- ULONG SLQSVoiceGetCallInfo (voiceCallInfoReq *pGetCallInfoReq, voiceCallInfoResp *pGetCallInfoResp)
- ULONG SLQSVoiceGetAllCallInfo (voiceGetAllCallInfo *pGetAllCallInfo)
- ULONG SLQSVoiceManageCalls (voiceManageCallsReq *pVoiceManageCallsReq, voiceManageCallsResp *pVoiceManageCallsResp)
- ULONG SLQSVoiceBurstDTMF (voiceBurstDTMFInfo *pBurstDTMFInfo)
- ULONG SLQSVoiceStartContDTMF (voiceContDTMFinfo *pContDTMFInfo)
- ULONG SLQSVoiceStopContDTMF (voiceStopContDTMFinfo *pVoiceStopContDTMFinfo)
- ULONG SLQSVoiceSendFlash (voiceFlashInfo *pFlashInfo)
- ULONG SLQSVoiceSetPreferredPrivacy (voiceSetPrefPrivacy *pSetPrefPrivacy)
- ULONG SLQSVoiceIndicationRegister (voiceIndicationRegisterInfo *pVoiceIndicationRegisterInfo)
- ULONG SLQSVoiceGetConfig (voiceGetConfigReq *pVoiceGetConfigReq, voiceGetConfigResp *pVoiceGetConfigResp)
- ULONG SLQSVoiceOrigUSSDNoWait (voiceOrigUSSDNoWaitInfo *pVoiceOrigUSSDNoWaitInfo)
- ULONG SLQSVoiceBindSubscription (voiceBindSubscriptionInfo *pVoiceBindSubscriptionInfo)
- ULONG SLQSVoiceALSSetLineSwitching (voiceALSSetLineSwitchInfo *pVoiceALSSetLineSwitchInfo)
- ULONG SLQSVoiceALSSelectLine (voiceALSSelectLineInfo *pVoiceALSSelectLineInfo)
- ULONG SLQSVoiceGetCOLP (voiceGetCOLPResp *pVoiceGetCOLPResp)
- ULONG SLQSVoiceGetCOLR (voiceGetCOLRResp *pVoiceGetCOLRResp)
- ULONG SLQSVoiceGetCNAP (voiceGetCNAPResp *pVoiceGetCNAPResp)
- ULONG SLQSOriginateUSSD (struct USSInfo *pReq, struct USSResp *pResp)

### 9.48.1 Detailed Description

Voice Service API function prototypes.

### 9.48.2 Macro Definition Documentation

#### 9.48.2.1 #define MAX_CALL_NO_LEN 81

#### 9.48.2.2 #define MAX_DESCRIPTION_LENGTH 255

#### 9.48.2.3 #define MAX_NO_OF_CALLS 20

#### 9.48.2.4 #define MAXUSSDLENGTH 182

#### 9.48.2.5 #define PASSWORD_LENGTH 4

### 9.48.3 Enumeration Type Documentation

#### 9.48.3.1 enum serviceClassInformation

Service Class information

**Enumerator**

***VOICE_SUPS_SRV_CLASS_NONE***

***VOICE_SUPS_SRV_CLASS_VOICE***

***VOICE_SUPS_SRV_CLASS_DATA***

***VOICE_SUPS_SRV_CLASS_FAX***

***VOICE_SUPS_SRV_CLASS_SMS***

***VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC***

***VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC***

***VOICE_SUPS_SRV_CLASS_PACKETACCESS***

***VOICE_SUPS_SRV_CLASS_PADACCESS***

### 9.48.4 Function Documentation

#### 9.48.4.1 ULONG AnswerUSSD ( BYTE * *pInfo* )

Responds to a USSD request from the network.

**Parameters**

| | |
|---|---|
| *pInfo[IN]* | • USS 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: UMTS
Timeout: 5 mins

#### 9.48.4.2 ULONG CancelUSSD ( )

Cancels an in-progress USSD operation.

**Parameters**

| | |
|---|---|
| *None* | |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 30 Secs

#### 9.48.4.3 ULONG OriginateUSSD ( BYTE * *pInfo* )

Initiates a USSD operation.

**Parameters**

| | |
|---|---|
| *pInfo[IN]* | • USS information<br>• See USSInfo for more details |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 5 mins

#### 9.48.4.4 ULONG SLQSOriginateUSSD ( struct USSInfo * *pReq*, struct USSResp * *pResp* )

Initiates a USSD session.

**Parameters**

| | |
|---|---|
| *pReq* | [IN]<br>• USS information<br>• See USSInfo for more details |
| *pResp* | [OUT]<br>• USS 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: UMTS
Device Supported: MC83x5
Timeout: 5 mins

#### 9.48.4.5 ULONG SLQSVoiceALSSelectLine ( voiceALSSelectLineInfo * *pVoiceALSSelectLineInfo* )

This API allows the user to select the preferred line.

**Parameters**

| | |
|---|---|
| *pVoiceALSSelectLineInfo[IN]* | • See voiceALSSelectLineInfo 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: UMTS
Timeout: 30 Secs
This API allows the user to select the preferred line, and the status is updated on the card. The API is supported only for specific SIM/USIMs that support alternate line service. This command is applicable only in 3GPP devices. A No Effect error is returned if the update on the card fails.

#### 9.48.4.6 ULONG SLQSVoiceALSSetLineSwitching ( voiceALSSetLineSwitchInfo * *pVoiceALSSetLineSwitchInfo* )

This API sets the line switch setting on the card.

**Parameters**

| | |
|---|---|
| *pVoiceALSSetLineSwitchInfo[IN]* | • See voiceALSSetLineSwitchInfo 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: UMTS
Timeout: 30 Secs
This API sets a line to be switchable or unswitchable, and the switch status is updated on the card. The API is supported only for specific SIM/USIMs that support alternate line service. This command is applicable only in 3GPP devices. A No Effect error is returned if the update on the card fails.

#### 9.48.4.7 ULONG SLQSVoiceAnswerCall ( voiceAnswerCall * *pVoiceAnswerCall* )

Answers an incoming voice call.

**Parameters**

| | |
|---|---|
| *pVoiceAnswer-Call[IN/OUT]* | • Pointer to structure of voiceAnswerCall<br>– See voiceAnswerCall 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: 10 Secs
This API is used to answer an incoming voice call when the incoming voice call is the only call present at that time. If there are other calls while an incoming call (waiting call) is received, API "SLQSVoiceSendFlash" can be used case of 3GPP2(CDMA) and API "SLQSVoiceManageCalls" in the case of 3GPP(UMTS). If the result indicates success, the device has started the requested operation and it does not mean that the call has been answered. "SLQSVoiceSetAllCallStatusCallback" can be subscribed to check the call Information/State.

#### 9.48.4.8 ULONG SLQSVoiceBindSubscription ( voiceBindSubscriptionInfo * *pVoiceBindSubscriptionInfo* )

This API binds a subscription type to a specific voice client ID.

**Parameters**

| | |
|---|---|
| *pVoiceBind-Subscription-Info[IN]* | • See voiceBindSubscriptionInfo 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: 30 Secs
Some versions of the modem support the Dual SIM feature. With this feature the modem can register with two different cellular networks simultaneously. Each network registration is associated with a different subscription, e.g., phone number, such that the modem appears to the network to be two different users. By default, the Voice client is bound to the primary subscription. This command allows the Voice client to change this binding. After receiving a successful response to this command, all future commands sent by the client will affect the newly bound subscription only.

#### 9.48.4.9 ULONG SLQSVoiceBurstDTMF ( voiceBurstDTMFInfo * *pBurstDTMFInfo* )

Sends a burst Dual-Tone Multi frequency (DTMF) (applicable only for 3GPP2)

**Parameters**

| | |
|---|---|
| *pBurstDTMF-Info[IN/OUT]* | • Structure containing parameters of burst DTMF.<br>• See voiceBurstDTMFInfo 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
Timeout: 30 Secs
Applicable only for 3GPP2. This API sends a burst DTMF. If API result indicates success, this means the device has started the requested operation. It does not mean that the burst DTMF request has been sent to the network. A burst DTMF request is sent to the current active/alerting call when CallId is set to 0xFF. This API is applicable only in 3GPP2.

#### 9.48.4.10 ULONG SLQSVoiceDialCall ( voiceCallRequestParams * *pCallRequestParams*, voiceCallResponseParams * *pCallResponseParams* )

Originates a voice call (MO call).

**Parameters**

| | |
|---|---|
| *pCallRequest-Params[IN]* | • Pointer to structure of voiceCallRequestParams<br>– See voiceCallRequestParams for more information |
| *pCallResponse-Params[OUT]* | • Pointer to structure of voiceCallResponseParams<br>– See voiceCallResponseParams 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: 30 Secs
This API originates a voice call (MO). If the function returns success with a call_id, the device has started the requested operation. It does not mean that the call has been connected. SLQSVoiceSetAllCallStatusCallback() callback can be subscribed to learn if the call was successful.

#### 9.48.4.11 ULONG SLQSVoiceEndCall ( BYTE * *pCallId* )

This message ends a voice call

**Parameters**

| | |
|---|---|
| *pCallId* | [IN/OUT]<br>• Unique call identifier for the call that must be ended |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 30 Secs
If the function returns success, the device has started the requested operation. It does not mean that the call has been ended. The application should always process the SLQSVoiceSetAllCallStatusCallback() callback to learn if the call was ended.

#### 9.48.4.12 ULONG SLQSVoiceGetAllCallInfo ( voiceGetAllCallInfo * *pGetAllCallInfo* )

This API queries the information associated with all the calls originating or terminating from a particular device.

**Parameters**

| | |
|---|---|
| *pGetAllCallInfo[-OUT]* | • See voiceGetAllCallInfo 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: 30 Secs
This command is used by the control point to get information of all the calls to and fro from the device in progress. The information keeps on updating constantly, as the state of a call changes example,from incoming to conversation to terminated.
This API requires a firmware with at least voice 2.0 support.

### 9.48.4.13 ULONG SLQSVoiceGetCallBarring ( voiceGetCallBarringReq * *pVoiceGetCallBarringReq*, voiceGetCallBarringResp * *pVoiceGetCallBarringResp* )

Queries the status of Call Barring Supplementary Service (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceGetCallBarringReq[IN]* | • Pointer to structure of voiceGetCallBarringReq<br>– See voiceGetCallBarringReq for more information |
| *pVoiceGetCallBarringResp[OUT]* | • Pointer to structure of voiceGetCallBarringResp<br>– See voiceGetCallBarringResp 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: UMTS
Timeout: 30 Secs
This API queries the status of the call barring supplementary service, i.e., to find whether the call barring supplementary service is active and, if active, for which service classes it is active. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

### 9.48.4.14 ULONG SLQSVoiceGetCallForwardingStatus ( voiceGetCallFWReq * *pVoiceGetCallFWReq*, voiceGetCallFWResp * *pVoiceGetCallFWResp* )

Queries the status of Call Forwarding Supplementary Service.

**Parameters**

| | |
|---|---|
| *pVoiceGetCallFWReq[IN]* | • Pointer to structure of voiceGetCallFWReq<br>– See voiceGetCallFWReq for more information |

| | |
|---|---|
| *pVoiceGetCallFWResp[OUT]* | • Pointer to structure of voiceGetCallFWResp<br>– See voiceGetCallFWResp 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: UMTS
Timeout: 30 Secs
This API queries the status of the call forwarding supplementary service, i.e., to find whether the call forwarding supplementary service is active and, if active, for which service classes and call forwarding number it is active. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

### 9.48.4.15 ULONG SLQSVoiceGetCallInfo ( voiceCallInfoReq * *pGetCallInfoReq,* voiceCallInfoResp * *pGetCallInfoResp* )

This API queries the information associated with a call and gives information about a particular call whose call Id is sent in as request.

**Parameters**

| | |
|---|---|
| *pGetCallInfoReq[IN]* | • See voiceCallInfoReq for more information. |
| *pGetCallInfoResp[OUT]* | • See voiceCallInfoResp 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: 30 Secs
If no voice call is in progress or an invalid call_id is sent in the request, an error is returned as the response.
This API requires a firmware with atleast voice 2.0 support.

#### 9.48.4.16 ULONG SLQSVoiceGetCallWaiting ( voiceGetCallWaitInfo * *pVoiceGetCallWaitInfo* )

Queries the status of Call Waiting Supplementary Service (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceGetCallWaitInfo[IN/OUT]* | • Pointer to structure of voiceGetCallWaitInfo<br>– See voiceGetCallWaitInfo 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: UMTS
Timeout: 30 Secs
This API queries the status of the call waiting supplementary service, i.e., to find whether the call waiting supplementary service is active. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

#### 9.48.4.17 ULONG SLQSVoiceGetCLIP ( voiceGetCLIPResp * *pVoiceGetCLIPResp* )

Queries the status of the Calling Line Identification Presentation (CLIP) supplementary service (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceGetCLIPResp[OUT]* | • Pointer to structure of voiceGetCLIPResp<br>– See voiceGetCLIPResp 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: UMTS
Timeout: 30 Secs
This API queries the status of the CLIP supplementary service. The active_status field is only applicable when provision_status is PROVISIONED, i.e., there is not any case where provision_status is NOT_PROVISIONED and active_status is ACTIVE. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

#### 9.48.4.18 ULONG SLQSVoiceGetCLIR ( voiceGetCLIRResp * *pVoiceGetCLIRResp* )

Queries the status of the Calling Line Identification Restriction (CLIR) supplementary service (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceGetCLIR-Resp[OUT]* | • Pointer to structure of voiceGetCLIRResp<br>– See voiceGetCLIRResp 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: UMTS
Timeout: 30 Secs
This API queries the status of the CLIR supplementary service. The active_status field is only applicable when provision_status is PROVISIONED, i.e., there is not any case where provision_status is NOT_PROVISIONED and active_status is ACTIVE. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

#### 9.48.4.19 ULONG SLQSVoiceGetCNAP ( voiceGetCNAPResp * *pVoiceGetCNAPResp* )

Queries the status of the Calling Name Presentation(CNAP) supplementary service (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceGetCNA-PResp[OUT]* | • Pointer to structure of voiceGetCNAPResp<br>– See voiceGetCNAPResp 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: UMTS
Timeout: 30 Secs

This API queries the status of the CNAP supplementary service. A response indicates whether CNAP is active/inactive and provisioned/not provisioned in the network. The active_status field is only applicable when

provision_status is PROVISIONED, i.e., there is not any case where provision_status is NOT_PROVISIONED and active_status is ACTIVE. This API is applicable only in 3GPP devices.

#### 9.48.4.20 ULONG SLQSVoiceGetCOLP ( voiceGetCOLPResp * *pVoiceGetCOLPResp* )

Queries the status of the Connected Line Identification Presentation (COLP) supplementary service (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceGetCOLPResp[OUT]* | • Pointer to structure of voiceGetCOLPResp<br>– See voiceGetCOLPResp 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: UMTS
Timeout: 30 Secs

This API queries the status of the COLP supplementary service. A response indicates whether COLP is active/inactive and provisioned/not provisioned in the network. The active_status field is only applicable when provision_status is PROVISIONED, i.e., there is not any case where provision_status is NOT_PROVISIONED and active_status is ACTIVE. This API is applicable only in 3GPP devices.

#### 9.48.4.21 ULONG SLQSVoiceGetCOLR ( voiceGetCOLRResp * *pVoiceGetCOLRResp* )

Queries the status of the Connected Line Identification Restriction (COLR) supplementary service (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceGetCOLRResp[OUT]* | • Pointer to structure of voiceGetCOLRResp<br>– See voiceGetCOLRResp 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: UMTS
Timeout: 30 Secs

This API queries the status of the COLR supplementary service. A response indicates whether COLR is active/inactive and provisioned/not provisioned in the network. The active_status field is only applicable when provision_status is PROVISIONED, i.e., there is not any case where provision_status is NOT_PROVISIONED and active_status is ACTIVE. This API is applicable only in 3GPP devices.

#### 9.48.4.22 ULONG SLQSVoiceGetConfig ( voiceGetConfigReq * *pVoiceGetConfigReq*, voiceGetConfigResp * *pVoiceGetConfigResp* )

This API retrieves various configuration parameters that control the modem behavior related to circuit switched services.

**Parameters**

| | |
|---|---|
| *pVoiceGetConfigReq* | • Structure containing Get Config request parameters.<br>– See voiceGetConfigReq for more information. |
| *pVoiceGetConfigResp* | • Structure containing Get Config response parameters.<br>– See voiceGetConfigResp 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: 30 Secs
Any invalid value in a request message causes the service point to reject the message without retrieving any configuration information.

#### 9.48.4.23 ULONG SLQSVoiceIndicationRegister ( voiceIndicationRegisterInfo * *pVoiceIndicationRegisterInfo* )

Sets the registration state for different QMI_VOICE indications for the requesting control point

**Parameters**

| | |
|---|---|
| *pVoiceIndicationRegisterInfo[IN]* | • Structure containing Indication Register Information.<br>– See voiceIndicationRegisterInfo 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: 10 Secs

```
This API is used by a device to register/deregister for different
QMI VOICE 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.
```

### 9.48.4.24 ULONG SLQSVoiceManageCalls ( voiceManageCallsReq * *pVoiceManageCallsReq*, voiceManageCallsResp * *pVoiceManageCallsResp* )

Manages the calls by using the supplementary service applicable during the call. In cases of successful API completion if the state of any call is changed, it is indicated using CallBack SLQSVoiceSetAllCallStatusCallBack. If there are other calls while an incoming voice call (waiting call) is received, this API is used to answer the call. This API is applicable only in "3GPP devices".

**Parameters**

| | |
|---|---|
| *pVoiceManage-CallsReq[IN]* | • Request structure of to manage calls. |
| *pVoiceManage-CallsResp[OUT]* | • Response Structure to manage Calls |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS
Timeout: 10 Secs
Applicable only for "3GPP"

### 9.48.4.25 ULONG SLQSVoiceOrigUSSDNoWait ( voiceOrigUSSDNoWaitInfo * *pVoiceOrigUSSDNoWaitInfo* )

This API initiates a USSD operation such that the response for this request is returned immediately and the data is returned via an indication.

**Parameters**

| | |
|---|---|
| *pVoiceOrigUSS-DNoWaitInfo[IN]* | • See voiceOrigUSSDNoWaitInfo 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: UMTS
Timeout: 30 Secs
This API starts a new USSD operation. The response to the request is sent immediately. The response result is sent to the client via the SLQSVoiceOrigUSSDNoWaitCallback. This command is applicable only in 3GPP devices.

#### 9.48.4.26 ULONG SLQSVoiceSendFlash ( voiceFlashInfo * *pFlashInfo* )

This API sends a simple flash message. Applicable only for 3GPP2 devices.

**Parameters**

| | |
|---|---|
| *pFlashInfo[IN/OUT]* | • See voiceFlashInfo 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
Timeout: 10 Secs
If success, it only means the device has started the requested operation and not that the Flash has been sent. If the optional parameter Flash Type is not set, the default flash type is assumed to be a simple flash. If the parameter Flash Type is set to 1 the call ID corresponding to it is either an incoming or waiting call's call ID. If the parameter Flash Type is set to 2 the call ID corresponding to it is a held call's call ID. A Flash request is sent to the appropriate call when call_id is set to 0xFF.

#### 9.48.4.27 ULONG SLQSVoiceSetCallBarringPassword ( voiceSetCallBarringPwdInfo * *pVoiceSetCallBarringPwdInfo*, voiceSetCallBarringPwdResp * *pSetCallBarringPwdResp* )

Sets a Call Barring Password (applicable only for 3GPP).

**Parameters**

| | |
|---|---|
| *pVoiceSetCallBarringPwdInfo[IN]* | • Pointer to structure of voiceSetCallBarringPwdInfo<br>– See voiceSetCallBarringPwdInfo for more information |

| | |
|---|---|
| *pSetCallBarring-PwdResp[OUT]* | • Pointer to structure of voiceSetCallBarringPwdResp<br>– See voiceSetCallBarringPwdResp 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: UMTS
Timeout: 30 Secs
This API changes the call barring supplementary service password. The request is a blocking request, i.e., the response is sent only after confirmation is received from the network. The maximum time it takes for the response to be sent is approximately 30 sec. This API is applicable only in 3GPP devices.

#### 9.48.4.28 ULONG SLQSVoiceSetConfig ( voiceSetConfigReq * *pVoiceSetConfigReq*, voiceSetConfigResp * *pVoiceSetConfigResp* )

This message sets various configuration parameters that control the modem behavior related to circuit-switched services.

**Parameters**

| | |
|---|---|
| *pVoiceSet-ConfigReq[IN]* | • Pointer to structure of voiceSetConfigReq<br>– See voiceSetConfigReq for more information |
| *pVoiceSet-ConfigResp[OUT]* | • Pointer to structure of voiceSetConfigResp<br>– See voiceSetConfigResp 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: 30 Secs
Any invalid value in a request message causes the device to reject the message without updating any configuration information. In the case of a successful update of all requested information, a QMI_ERR_NONE error is returned. In the case where a subset of information failed to be written, a QMI_ERR_INTERNAL error is returned with corresponding optional information requested in the request message.

#### 9.48.4.29 ULONG SLQSVoiceSetPreferredPrivacy ( voiceSetPrefPrivacy * *pSetPrefPrivacy* )

This API sets the voice privacy preference. Applicable only for 3GPP2 devices.

**Parameters**

| | |
|---|---|
| *pSetPref-Privacy[IN]* | • See voiceSetPrefPrivacy 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
Device Supported: SL9090
Timeout: 10 Secs

#### 9.48.4.30 ULONG SLQSVoiceSetSUPSService ( voiceSetSUPSServiceReq * *pVoiceSetSUPSServiceReq,* voiceSetSUPSServiceResp * *pVoiceSetSUPSServiceResp* )

This API manages call-independent supplementary services, e.g., activation of call forwarding (to forward incoming calls to a third party), activation of call barring (to request the network to block some of the call attempts), and activation of call waiting (to be notified of an incoming call even when the user is engaged in an active or held call).

**Parameters**

| | |
|---|---|
| *pVoiceSetSUPS-ServiceReq* | [IN]<br>• Pointer to structure of voiceSetSUPSServiceReq<br>– See voiceSetSUPSServiceReq for more information |
| *pVoiceSetSUPS-ServiceResp* | [OUT]<br>• Pointer to structure of voiceSetSUPSServiceResp<br>– See voiceSetSUPSServiceResp 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: UMTS
Timeout: 30 Secs
Some of the call-independent services are provided by the network operator as part of the service agreement. If they are not provided by default, the user has to explicitly request them. This API provides the facility to the control point for sending the explicit request to the network for enabling/disabling

#### 9.48.4.31 ULONG SLQSVoiceStartContDTMF ( voiceContDTMFinfo * *pContDTMFInfo* )

Starts a continuous DTMF.

**Parameters**

| | |
|---|---|
| *pContDTMF-Info[IN/OUT]* | • Structure containing Continuous DTMF Information.<br>– See voiceContDTMFinfo 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: 30 Secs
This API starts a continuous DTMF. If the API results indicates success, it means that the device has started the requested operation. It does not mean that the start continuous DTMF request has been sent to the network. A start continuous DTMF request is sent to the current active/alerting call when CallId is set to 0xFF.

#### 9.48.4.32 ULONG SLQSVoiceStopContDTMF ( voiceStopContDTMFinfo * *pVoiceStopContDTMFinfo* )

Stops a continuous DTMF.

**Parameters**

| | |
|---|---|
| *pVoiceStopCont-DTMFinfo* | • Structure containing Continuous Stop DTMF Information.<br>– See voiceStopContDTMFinfo for more information.<br>• Start continuous DTMF request is sent to the current active/alerting call when CallId is set to 0xFF.<br>• This is IN/OUT params, value passed by user will packed in request and before unpacking response this will be assigned with an invalid callID value "0". It change to a valid value if received as part of response otherwise Invalid value will be present. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 30 Secs
This API starts a continuous DTMF. If the API results indicates success, it means that the device has started the requested operation. It does not mean that the start continuous DTMF request has been sent to the network. A stop continuous DTMF request is sent to the current active/alerting call when CallId is set to 0xFF.

## 9.49 qaGobiApiWds.h File Reference

Wireless Data Service API function prototypes.

### Data Structures

- struct dataBearerTechnology
- struct dataBearers
- struct QmiWdsIpAddressInfo
- struct WdsIpAddressInfoReq
- struct UMTSQoS
- struct ProfileIdentifier
- struct GPRSQoS
- struct PCSCFIPv4ServerAddressList
- struct PCSCFFQDNAddress
- struct PCSCFFQDNAddressList
- struct Domain
- struct DomainNameList
- struct IPV6AddressInfo
- struct IPV6GWAddressInfo
- struct qmiWdsRunTimeSettings
- struct WdsRunTimeSettings
- struct ssdatasession_params
- struct SLQSDeleteProfileParams
- struct Profile3GPP
- struct Profile3GPP2
- union WdsProfileParam
- struct CreateProfileIn
- struct UMTSMinQoS
- struct GPRSRequestedQoS
- struct TFTIDParams
- struct UMTSReqQoSSigInd
- struct QosClassID
- struct CreateProfileOut
- struct slqsautoconnect
- struct ModifyProfileIn
- struct ModifyProfileOut
- struct _slqs3GPPConfigItem
- struct _GetProfileSettingIn
- struct _GetProfileSettingOut
- struct TrStatInd
- struct wdsSetEventReportReq
- struct swiPDPRuntimeSettingsReq

- struct swiPDPRuntimeSettingsResp
- struct WdsConnectionRateElmnts
- struct WdsConnectionRate
- struct WdsByteTotalsElmnts
- struct WdsByteTotals
- struct GetInstIDResp
- struct GetSessionIDResp
- struct TransferStatInd
- struct getDUNCallInfoReq
- struct ConnectionStatus
- struct ChannelRate
- struct getDUNCallInfoResp
- struct WdsPktStatisticsElmnts
- struct WdsPktStatisticsResp
- struct WdsPktStatisticsReq
- struct CurrNetworkInfo
- struct CurrDataSysStat
- struct swiRMTrasnferStaticsReq
- struct DataULongTlv
- struct DataULongLongTlv
- struct QmiCbkWdsStatisticsIndState
- struct DataBearerTech
- struct DataBearerTechExt
- struct WDSSWICurrentChannelRates
- struct WDSSetLoopbackData
- struct WDSGetLoopbackData
- struct WdsDHCPv4ProfileId
- struct WdsDHCPv4HWConfig
- struct WdsDHCPv4Option
- struct WdsDHCPv4OptionList
- struct WdsDHCPv4Config
- struct WdsClientLeaseChange

## Macros

- #define IPV6_ADDRESS_ARRAY_SIZE 8

## Typedefs

- typedef struct dataBearerTechnology QmiWDSDataBearerTechnology
- typedef struct dataBearers QmiWDSDataBearers
- typedef union WdsProfileParam QmiProfileInfo
- typedef struct _slqs3GPPConfigItem slqs3GPPConfigItem
- typedef struct _GetProfileSettingIn GetProfileSettingIn
- typedef struct _GetProfileSettingOut GetProfileSettingOut

## Enumerations

- enum qmiDataBearerMasks { QMI_WDS_CURRENT_CALL_DB_MASK = 0x01, QMI_WDS_LAST_CALL_DB_MASK = 0x02 }

## Functions

- ULONG SetMobileIP (ULONG mode)
- ULONG GetMobileIP (ULONG *pMode)
- ULONG SetMobileIPParameters (CHAR *pSPC, ULONG *pMode, BYTE *pRetryLimit, BYTE *pRetryInterval, BYTE *pReRegPeriod, BYTE *pReRegTraffic, BYTE *pHAAuthenticator, BYTE *pHA2002bis)
- ULONG SetAutoconnect (ULONG setting)
- ULONG GetAutoconnect (ULONG *pSetting)
- ULONG SetDefaultProfile (ULONG profileType, ULONG *pPDPType, ULONG *pIPAddress, ULONG *pPrimaryDNS, ULONG *pSecondaryDNS, ULONG *pAuthentication, CHAR *pName, CHAR *pAPNName, CHAR *pUsername, CHAR *pPassword)
- ULONG SetDefaultProfileLTE (ULONG profileType, ULONG *pPDPType, ULONG *pIPAddressv4, ULONG *pPrimaryDNSv4, ULONG *pSecondaryDNSv4, USHORT *pIPAddressv6, USHORT *PrimaryDNSv6, USHORT *pSecondaryDNSv6, ULONG *pAuthentication, CHAR *pName, CHAR *pAPNName, CHAR *pUsername, CHAR *pPassword)
- ULONG SetDefaultProfileLTEV2 (ULONG profileType, ULONG *pPDPType, ULONG *pIPAddressv4, ULONG *pPrimaryDNSv4, ULONG *pSecondaryDNSv4, USHORT *pIPAddressv6, USHORT *PrimaryDNSv6, USHORT *pSecondaryDNSv6, ULONG *pAuthentication, CHAR *pName, CHAR *pAPNName, CHAR *pUsername, CHAR *pPassword)
- ULONG GetDefaultProfile (ULONG profileType, ULONG *pPDPType, ULONG *pIPAddress, ULONG *pPrimaryDNS, ULONG *pSecondaryDNS, ULONG *pAuthentication, BYTE nameSize, CHAR *pName, BYTE apnSize, CHAR *pAPNName, BYTE userSize, CHAR *pUsername)
- ULONG GetDefaultProfileLTE (ULONG profileType, ULONG *pPDPType, ULONG *pIPAddressv4, ULONG *pPrimaryDNSv4, ULONG *pSecondaryDNSv4, USHORT *pIPAddressv6, USHORT *pPrimaryDNSv6, USHORT *pSecondaryDNSv6, ULONG *pAuthentication, BYTE nameSize, CHAR *pName, BYTE apnSize, CHAR *pAPNName, BYTE userSize, CHAR *pUsername)
- ULONG GetSessionState (ULONG *pState, BYTE instance)
- ULONG GetPacketStatus (ULONG *pTXPacketSuccesses, ULONG *pRXPacketSuccesses, ULONG *pTXPacketErrors, ULONG *pRXPacketErrors, ULONG *pTXPacketOverflows, ULONG *pRXPacketOverflows, BYTE instance)
- ULONG GetByteTotals (ULONGLONG *pTXTotalBytes, ULONGLONG *pRXTotalBytes, BYTE instance)
- ULONG GetDormancyState (ULONG *pDormancyState, BYTE instance)
- ULONG GetDataBearerTechnology (ULONG *pDataBearer, BYTE instance)
- ULONG SLQSGetDataBearerTechnology (QmiWDSDataBearers *pDataBearers, BYTE instance)
- ULONG GetSessionDuration (ULONGLONG *pDuration, BYTE instance)
- ULONG GetIPAddressLTE (WdsIpAddressInfoReq *)
- ULONG GetConnectionRate (ULONG *pCurrentChannelTXRate, ULONG *pCurrentChannelRXRate, ULONG *pMaxChannelTXRate, ULONG *pMaxChannelRXRate, BYTE instance)
- ULONG GetMobileIPProfile (BYTE index, BYTE *pEnabled, ULONG *pAddress, ULONG *pPrimaryHA, ULONG *pSecondaryHA, BYTE *pRevTunneling, BYTE naiSize, CHAR *pNAI, ULONG *pHASPI, ULONG *pAAASPI, ULONG *pHAState, ULONG *pAAAState)
- ULONG GetLastMobileIPError (ULONG *pError)
- ULONG iSLQSMISetIPFamilyPreference (BYTE IPFamilyPreference, BYTE instance)
- BOOL WDS_IsGobiDevice ()
- ULONG SetActiveMobileIPProfile (CHAR *pSPC, BYTE index)
- ULONG SetMobileIPProfile (CHAR *pSPC, BYTE index, BYTE *pEnabled, ULONG *pAddress, ULONG *pPrimaryHA, ULONG *pSecondaryHA, BYTE *pRevTunneling, CHAR *pNAI, ULONG *pHASPI, ULONG *pAAASPI, CHAR *pMNHA, CHAR *pMNAAA)
- ULONG SLQSGetRuntimeSettings (struct WdsRunTimeSettings *pRunTimeSettings)
- ULONG SLQSSetProfile (ULONG profileType, BYTE profileId, ULONG *pPDPType, ULONG *pIPAddress, ULONG *pPrimaryDNS, ULONG *pSecondaryDNS, ULONG *pAuthentication, CHAR *pName, CHAR *pAPNName, CHAR *pUsername, CHAR *pPassword)
- ULONG SLQSGetProfile (ULONG profileType, BYTE profileId, ULONG *pPDPType, ULONG *pIPAddress, ULONG *pPrimaryDNS, ULONG *pSecondaryDNS, ULONG *pAuthentication, BYTE nameSize, CHAR *pName, BYTE apnSize, CHAR *pAPNName, BYTE userSize, CHAR *pUsername, WORD *pExtendedErrorCode)

- ULONG SLQSStartStopDataSession (struct ssdatasession_params *pin)
- ULONG SLQSDeleteProfile (struct SLQSDeleteProfileParams *pProfileToDelete, WORD *pExtendedErrorCode)
- ULONG SLQSCreateProfile (struct CreateProfileIn *pReq, struct CreateProfileOut *pResp)
- ULONG SLQSAutoConnect (struct slqsautoconnect *pacreq)
- ULONG SLQSModifyProfile (struct ModifyProfileIn *pReq, struct ModifyProfileOut *pResp)
- ULONG SLQSSet3GPPConfigItem (slqs3GPPConfigItem *pSLQS3GPPConfigItem)
- ULONG SLQSGet3GPPConfigItem (slqs3GPPConfigItem *pSLQS3GPPConfigItem)
- ULONG SLQSGetProfileSettings (GetProfileSettingIn *pReq, GetProfileSettingOut *pResp)
- ULONG SLQSWdsSetEventReport (wdsSetEventReportReq *pSetEventReportReq)
- ULONG SLQSWdsSwiPDPRuntimeSettings (swiPDPRuntimeSettingsReq *pPDPRuntimeSettingsReq, swiPDPRuntimeSettingsResp *pPDPRuntimeSettingsResp)
- ULONG iGetConnectionRate (ULONG *pv4sessionId, ULONG *pv6sessionId, struct WdsConnectionRateElmnts *pConnectionRateElmnt)
- ULONG SLQSGetConnectionRate (struct WdsConnectionRate *pConnectionRate)
- ULONG iGetByteTotals (ULONG *pv4sessionId, ULONG *pv6sessionId, struct WdsByteTotalsElmnts *pByteTotalsElmnt)
- ULONG SLQSGetByteTotals (struct WdsByteTotals *pByteTotals)
- ULONG SLQSWdsGoDormant (void)
- ULONG SLQSWdsGoActive (void)
- ULONG SLQSGetSessionState (ULONG *pStateV4, ULONG *pStateV6, BYTE instance)
- ULONG SLQSGetDUNCallInfo (getDUNCallInfoReq *pGetDUNCallInfoReq, getDUNCallInfoResp *pGetDUNCallInfoResp)
- ULONG GetPacketStatistics (struct WdsPktStatisticsReq *pStatMask, struct WdsPktStatisticsElmnts *pPktStatisticsElmnt, BYTE instance)
- ULONG iGetPacketStatistics (ULONG *pV4sessionId, ULONG *pV6sessionId, struct WdsPktStatisticsReq *pStatMask, struct WdsPktStatisticsElmnts *pPktStatisticsElmnt)
- ULONG SLQSGetPacketStatistics (struct WdsPktStatisticsReq *pStatMask, struct WdsPktStatisticsResp *pPktStatistics)
- ULONG SLQSGetCurrDataSystemStat (CurrDataSysStat *pCurrDataSysStat)
- ULONG RMSetTransferStatistics (swiRMTrasnferStaticsReq *pSwiRMTrasnferStaticsReq)
- ULONG SLQSResetPacketStatics ()
- ULONG SLQSGetDataBearerTechnologyExt (DataBearerTechExt *pDataBearerTech, BYTE instance)
- ULONG SLQSGetCurrentChannelRate (WDSSWICurrentChannelRates *pRates, BYTE instance)
- ULONG SLQSSSetLoopback (WDSSetLoopbackData *pReq)
- ULONG SLQSSGetLoopback (WDSGetLoopbackData *data)
- ULONG GetDefaultProfileNum (BYTE profile_type, BYTE profile_family, BYTE *pProfile_no)
- ULONG SetDefaultProfileNum (BYTE profile_type, BYTE profile_family, BYTE profile_index)
- ULONG SLQSSSetDHCPv4ClientConfig (WdsDHCPv4Config *pReq)
- ULONG SLQSSGetDHCPv4ClientConfig (WdsDHCPv4Config *pReqResp)
- ULONG SetMuxID (BYTE instance)

### 9.49.1 Detailed Description

Wireless Data Service API function prototypes.

### 9.49.2 Macro Definition Documentation

#### 9.49.2.1 #define IPV6_ADDRESS_ARRAY_SIZE 8

### 9.49.3 Typedef Documentation

#### 9.49.3.1 typedef struct _GetProfileSettingIn GetProfileSettingIn

This structure contains the input parameters for SLQSGetProfileSettings

**Parameters**

| | |
|---|---|
| *ProfileType* | • Identifies the technology type of the profile<br>– 0x00 - 3GPP<br>– 0x01 - 3GPP2 |
| *ProfileID* | • index identifying the profile |

#### 9.49.3.2 typedef struct _GetProfileSettingOut GetProfileSettingOut

This structure contains the profile settings retrieved by the API SLQSGetProfileSettings

**Parameters**

| | |
|---|---|
| *curProfile* | • Structure containing details of the profile<br>• See QmiProfileInfo for more details |
| *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. |

#### 9.49.3.3 typedef union WdsProfileParam QmiProfileInfo

This union WdsProfileParam consist of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

**Parameters**

| | |
|---|---|
| *SlqsProfile3GPP* | • 3GPP profile |
| *SlqsProfile3GPP2* | • 3GPP2 profile |

#### 9.49.3.4 typedef struct dataBearers QmiWDSDataBearers

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 qmiDataBearerMasks for bit-mask positions. |

| | |
|---|---|
| *pCurData-Bearer-Technology[OU-T]* | • current data bearer technology value.<br>– NULL if the parameter is not required |
| *pLastCallData-Bearer-Technology[OU-T]* | • last call data bearer technology value.<br>– NULL if the parameter is not required |

#### 9.49.3.5 typedef struct dataBearerTechnology QmiWDSDataBearerTechnology

Structure to hold the current data bearer technology values

**Parameters**

| | |
|---|---|
| *pCurrent-Network[OUT]* | • current selected network<br>– 0 - UNKNOWN<br>– 1 - 3GPP2<br>– 2 - 3GPP |
| *pRatMask[OUT]* | • Radio Access Technology (RAT) mask to indicate the type of technology (RAT mask value of zero indicates that this field is ignored)<br>– 0x8000 - NULL Bearer<br>– 0x0000 - DO_NOT_CARE CDMA RAT mask<br>– 0x01 - CDMA_1X<br>– 0x02 - EVDO_REV0<br>– 0x04 - EVDO_REVA UMTS RAT mask<br>– 0x01 - WCDMA<br>– 0x02 - GPRS<br>– 0x04 - HSDPA<br>– 0x08 - HSUPA<br>– 0x10 - EDGE<br>– 0x20 - LTE<br>– 0x40 - HSDPA+<br>– 0x80 - DC_HSDPA+ |
| *pSoMask[OUT]* | • Service Option (SO) mask to indicate the SO or type of application (SO mask value of zero indicates that this field is ignored)<br>– 0x00 - DO_NOT_CARE CDMA 1X SO mask<br>– 0x01 - CDMA_1X_IS95<br>– 0x02 - CDMA_1X_IS2000<br>– 0x04 - CDMA_1X_IS2000_REL_A CDMA EV-DO Rev A SO mask<br>– 0x01 - EVDO_REVA_DPA<br>– 0x02 - EVDO_REVA_MFPA<br>– 0x04 - EVDO_REVA_EMPA<br>– 0x08 - EVDO_REVA_EMPA_EHRPD |

#### 9.49.3.6 typedef struct _slqs3GPPConfigItem slqs3GPPConfigItem

This structure contains the 3gpp Configuration Item information.

**Parameters**

| | |
|---|---|
| *pLTEAttach-Profile* | • 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>– function SLQSGet3GPPConfigItem() returns a default value 255 if no LTE Attach Profile is configured<br>• This setting is deprecated on MC/EM74xx |
| *pProfileList* | • 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* | • Always Connect Default PDN<br>– A single BYTE value indicating the status of Always connect default PDN<br>* 0 - disabled<br>* 1 - enabled<br>– Optional parameter<br>– function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present |
| *p3gppRelease* | • 3gpp release<br>– A single BYTE value indicating the 3gpp release<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<br>– Optional parameter<br>– function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present |

| | |
|---|---|
| *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>– The new equivalent option for "pLTEAttachProfile" on 74xx modems is "pLTEAttachProfile-List". 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 |

### 9.49.4 Enumeration Type Documentation

#### 9.49.4.1 enum qmiDataBearerMasks

Bit mask values to indicate the presence of data bearer information for the current and last data calls

**Enumerator**

***QMI_WDS_CURRENT_CALL_DB_MASK***

***QMI_WDS_LAST_CALL_DB_MASK***

### 9.49.5 Function Documentation

#### 9.49.5.1 ULONG GetAutoconnect ( ULONG * *pSetting* )

Returns the current auto connect data session setting.

**Parameters**

| | |
|---|---|
| *pSetting[OUT]* | • NDIS auto connect setting<br>– 0 - Disabled<br>– 1 - Enabled |

**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

#### 9.49.5.2 ULONG GetByteTotals ( ULONGLONG * *pTXTotalBytes,* ULONGLONG * *pRXTotalBytes,* BYTE *instance* )

Returns the Rx/Tx byte counts since the start of the last packet data session for IPV4 session only.

**Parameters**

| | |
|---|---|
| *pTXTotalBytes[-OUT]* | • Bytes transmitted without error |
| *pRXTotalBytes[-OUT]* | • Bytes received without error |
| *instance* | • PDP instance |

**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, Rx/Tx byte counts for IPV4 only

#### 9.49.5.3 ULONG GetConnectionRate ( ULONG * *pCurrentChannelTXRate,* ULONG * *pCurrentChannelRXRate,* ULONG * *pMaxChannelTXRate,* ULONG * *pMaxChannelRXRate,* BYTE *instance* )

Returns connection rate information for the packet data connection. This API is not applicable when multiple data session is up. For multiple PDN, please use API SLQSGetConnectionRate()

**Parameters**

| | |
|---|---|
| *pCurrent-ChannelTX-Rate[OUT]* | • Current channel Tx rate (in bps) |
| *pCurrent-ChannelRX-Rate[OUT]* | • Current channel Rx rate (in bps) |
| *pMaxChannelT-XRate[OUT]* | • Maximum Tx rate (bps) that may be assigned to device by serving system. |
| *pMaxChannelR-XRate[OUT]* | • Maximum Rx rate (bps) that may be assigned to device by serving system. |
| *instance* | • PDP instance |

**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

#### 9.49.5.4 ULONG GetDataBearerTechnology ( ULONG * *pDataBearer,* BYTE *instance* )

Retrieves the current data bearer technology (only valid when connected). This API is deprecated on MC73xx/E-M73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSGetDataBearerTechnologyExt() for new firmware versions and new modules.

**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 |
| *instance* | • PDP instance |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

The QMI command of this API is deprecated. Use SLQSGetDataBearerTechnologyExt() for new modules (chipset 9x15, 9x30). Version Introduced: Major-1, Minor-12 Version Deprecated: Major-1, Minor-40 Timeout: 2 seconds

#### 9.49.5.5 ULONG GetDefaultProfile ( ULONG *profileType,* ULONG * *pPDPType,* ULONG * *pIPAddress,* ULONG * *pPrimaryDNS,* ULONG * *pSecondaryDNS,* ULONG * *pAuthentication,* BYTE *nameSize,* CHAR * *pName,* BYTE *apnSize,* CHAR * *pAPNName,* BYTE *userSize,* CHAR * *pUsername* )

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

**Parameters**

| | |
|---|---|
| *profileType* | • Type of profile<br>– 0 - UMTS |
| *pPDPType[OUT]* | • 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 |
| *pIPAddress[OUT]* | • Preferred IPv4 address to be assigned to device |
| *pPrimaryDNS[OUT]* | • Primary DNS Ipv4 address preference |
| *pSecondaryDNS[OUT]* | • Secondary DNS Ipv4 address preference |
| *pAuthentication[OUT]* | • 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. |
| *nameSize* | • Maximum number of characters (including NULL terminator) that profile name array can contain. |
| *pName[OUT]* | • Profile name |
| *apnSize* | • Maximum number of characters (including NULL terminator) that APN name array can contain |

| | |
|---|---|
| *pAPNName[OUT]* | • 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. |
| *pUsername[OUT]* | • Username used during network authentication |

**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

### 9.49.5.6 ULONG GetDefaultProfileLTE ( ULONG *profileType*, ULONG * *pPDPType*, ULONG * *pIPAddressv4*, ULONG * *pPrimaryDNSv4*, ULONG * *pSecondaryDNSv4*, USHORT * *pIPAddressv6*, USHORT * *pPrimaryDNSv6*, USHORT * *pSecondaryDNSv6*, ULONG * *pAuthentication*, BYTE *nameSize*, CHAR * *pName*, BYTE *apnSize*, CHAR * *pAPNName*, BYTE *userSize*, CHAR * *pUsername* )

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

**Parameters**

| | |
|---|---|
| *profileType* | • Type of profile<br>– 0 - UMTS |
| *pPDPType[OUT]* | • 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 |
| *pIPAddressv4[OUT]* | • Preferred IPv4 addr to be assigned to device |
| *pPrimaryDNSv4[OUT]* | • Primary DNS Ipv4 address preference |

| | |
|---|---|
| *pSecondaryDNSv4[OUT]* | • Secondary DNS Ipv4 address preference |
| *pIPAddressv6[OUT]* | • Preferred IPv6 addr to be assigned to device Space for storing 8 element array for the IPv6 addresses is allocated by the application. The IP Address will be retrieved in the big endian format. For example User buffer contents: [<U0>..<U7>]<br>IPv6 address:<br>1234:2A01:....:5678<br>U0 corresponds to 1234<br>U1 corresponds to 2A01<br>– ----------— – -—<br>– ----------— – -—<br>U7 corresponds to 5678 |
| *pPrimaryDNSv6[OUT]* | • Primary DNS Ipv6 address preference |
| *pSecondaryDNSv6[OUT]* | • Secondary DNS Ipv6 address preference |
| *pAuthentication[OUT]* | • 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. |
| *nameSize* | • Maximum number of characters (including NULL terminator) that Profile name array can contain |
| *pName[OUT]* | • Profile name |
| *apnSize* | • Maximum number of characters (including NULL terminator) that APN name array can contain |
| *pAPNName[IN]* | • 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. |

| | |
|---|---|
| *pUsername[OUT]* | • Username used during network authentication |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: LTE
Timeout: 2 seconds

#### 9.49.5.7 ULONG GetDefaultProfileNum ( BYTE *profile_type*, BYTE *profile_family*, BYTE * *pProfile_no* )

This API to Get default profile number

**Parameters**

| | |
|---|---|
| *profile_type* | [IN]<br>• 0 - 3GPP<br>• 1 - 3GPP2 |
| *profile_family* | [IN]<br>• 0 - Embedded<br>• 1 - Tethered |
| *pProfile_no* | [OUT] |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.8 ULONG GetDormancyState ( ULONG * *pDormancyState*, BYTE *instance* )

Returns the dormancy state of the current packet data session when connected.

**Parameters**

| | |
|---|---|
| *pDormancyState[OUT]* | • Dormancy state of current packet data session<br>– 1 - Traffic channel dormant<br>– 2 - Traffic channel active |

| | |
|---|---|
| *instance* | • PDP instance |

**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

#### 9.49.5.9 ULONG GetIPAddressLTE ( WdsIpAddressInfoReq * )

Returns the current packet data session IP address(es)

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

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: LTE
Timeout: 2 seconds.

#### 9.49.5.10 ULONG GetLastMobileIPError ( ULONG * *pError* )

Returns the last mobile IP error.

**Parameters**

| | |
|---|---|
| *pError[OUT]* | • Status of last MIP call (or attempt)<br>– Zero - Success<br>– NonZero - Error code<br>See qaGobiApiTableCallEndReasons.h for Mobile IP Error codes |

**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
Device Supported: MC83x5
Timeout: 2 seconds

#### 9.49.5.11 ULONG GetMobileIP ( ULONG * *pMode* )

Returns the current mobile IP setting.

**Parameters**

| | |
|---|---|
| *mode[OUT]* | • Mobile IP setting<br>– 0 - Mobile IP off (simple IP only)<br>– 1 - Mobile IP preferred<br>– 2 - Mobile IP only |

**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
Timeout: 2 seconds

#### 9.49.5.12 ULONG GetMobileIPProfile ( BYTE *index*, BYTE * *pEnabled*, ULONG * *pAddress*, ULONG * *pPrimaryHA*, ULONG * *pSecondaryHA*, BYTE * *pRevTunneling*, BYTE *naiSize*, CHAR * *pNAI*, ULONG * *pHASPI*, ULONG * *pAAASPI*, ULONG * *pHAState*, ULONG * *pAAAState* )

Returns the specified mobile IP profile settings.

**Parameters**

| | |
|---|---|
| *index* | • Mobile IP profile ID |
| *pEnabled[OUT]* | • Profile enabled:<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |

| | |
|---|---|
| *pAddress[OUT]* | • Home IPv4 address:<br>– 0xFFFFFFFF - Unknown |
| *pPrimaryHA[OUT]* | • Primary home agent IPv4 address<br>– 0xFFFFFFFF - Unknown |
| *pSecondaryHA[OUT]* | • Secondary home agent IPv4 address<br>– 0xFFFFFFFF - Unknown |
| *pRevTunneling[OUT]* | • Reverse tunneling enabled<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Unknown |
| *naiSize* | • The maximum number of characters (including NULL terminator) that the NAI array can contain. |
| *pNAI[OUT]* | • Network access identifier string |
| *pHASPI[OUT]* | • Home agent security parameter index |
| *pAAASPI[OUT]* | • AAA server security parameter index<br>– 0xFFFFFFFF - Unknown |
| *pHAState[OUT]* | • Home agent key state<br>– 0 - Unset<br>– 1 - Set, default value<br>– 2 - Set, modified from default<br>– 3 - 0xFFFFFFFF - Unknown |
| *pAAAState[OUT]* | • AAA key state<br>– 0 - Unset<br>– 1 - Set, default value<br>– 2 - Set, modified from default<br>– 3 - 0xFFFFFFFF - Unknown |

**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
Timeout: 2 seconds

### 9.49.5.13 ULONG GetPacketStatistics ( struct WdsPktStatisticsReq * *pStatMask*, struct WdsPktStatisticsElmnts * *pPktStatisticsElmnt*, BYTE *instance* )

Returns the current packet transfer counter values from the device. Since the start of the last packet data session.

**Parameters**

| | |
|---|---|
| *pStatMask[IN]* | • See WdsPktStatisticsReq for more information |
| *pPktStatistics-Elmnt[OUT]* | • See WdsPktStatisticsElmnts for more information |
| *instance* | • PDP instance |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 seconds

### 9.49.5.14 ULONG GetPacketStatus ( ULONG * *pTXPacketSuccesses*, ULONG * *pRXPacketSuccesses*, ULONG * *pTXPacketErrors*, ULONG * *pRXPacketErrors*, ULONG * *pTXPacketOverflows*, ULONG * *pRXPacketOverflows*, BYTE *instance* )

Returns the packet data transfer statistics since the start of the current packet data.

**Parameters**

| | |
|---|---|
| *pTXPacket-Successes[OUT]* | • Number of packets transmitted without error |
| *pRXPacket-Successes[OUT]* | • Number of packets received without error |
| *pTXPacket-Errors[OUT]* | • Number of outgoing packets with framing errors |

| | |
|---|---|
| *pRXPacket-Errors[OUT]* | • Number of incoming packets with framing errors |
| *pTXPacket-Overflows[OUT]* | • Number of packets dropped because Tx buffer overflowed |
| *pRXPacket-Overflows[OUT]* | • Number of packets dropped because Rx buffer overflowed |
| *instance* | • PDP instance |

**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

#### 9.49.5.15 ULONG GetSessionDuration ( ULONGLONG * *pDuration*, BYTE *instance* )

Returns the duration of the current packet data session.

**Parameters**

| | |
|---|---|
| *pDuration[OUT]* | • Duration of the current packet session in milliseconds |
| *instance* | • PDP instance |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS/CDMA
Device Supported: MC83x5, MC7700/50
Timeout: 2 seconds

#### 9.49.5.16 ULONG GetSessionState ( ULONG * *pState,* BYTE *instance* )

Returns the state of the current packet data session.

**Parameters**

| | |
|---|---|
| *pState[OUT]* | • Current link status<br>– 1 - DISCONNECTED<br>– 2 - CONNECTED<br>– 3 - SUSPENDED (not supported)<br>– 4 - AUTHENTICATING |
| *instance* | • PDP instance |

**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

#### 9.49.5.17 ULONG iGetByteTotals ( ULONG * *pv4sessionId,* ULONG * *pv6sessionId,* struct WdsByteTotalsElmnts * *pByteTotalsElmnt* )

#### 9.49.5.18 ULONG iGetConnectionRate ( ULONG * *pv4sessionId,* ULONG * *pv6sessionId,* struct WdsConnectionRateElmnts * *pConnectionRateElmnt* )

#### 9.49.5.19 ULONG iGetPacketStatistics ( ULONG * *pV4sessionId,* ULONG * *pV6sessionId,* struct WdsPktStatisticsReq * *pStatMask,* struct WdsPktStatisticsElmnts * *pPktStatisticsElmnt* )

#### 9.49.5.20 ULONG iSLQSMISetIPFamilyPreference ( BYTE *IPFamilyPreference,* BYTE *instance* )

#### 9.49.5.21 ULONG RMSetTransferStatistics ( swiRMTrasnferStaticsReq * *pSwiRMTrasnferStaticsReq* )

This API request the device to fetch current data system transfer Statistics.

**Parameters**

| | |
|---|---|
| *pswiRMTrasnfer-StaticsReq[IN]* | • See swiRMTrasnferStaticsReq for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.22 ULONG SetActiveMobileIPProfile ( CHAR * *pSPC,* BYTE *index* )

Sets active mobile IP profile.

**Parameters**

| | |
|---|---|
| *pSPC[IN]* | • NULL-terminated string representing six digit service programming code |
| *index[IN]* | • Index of the profile to be set as the active profile |

**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
Timeout: 2 seconds

#### 9.49.5.23 ULONG SetAutoconnect ( ULONG *setting* )

Sets the auto connect data session setting.

**Parameters**

| | |
|---|---|
| *setting[IN]* | • NDIS autoconnect setting<br>– 0 - Disabled<br>– 1 - Enabled |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

When enabling, timeout is 5 minutes,
When disabling, timeout is 5 seconds

#### 9.49.5.24 ULONG SetDefaultProfile ( ULONG *profileType,* ULONG * *pPDPType,* ULONG * *pIPAddress,* ULONG * *pPrimaryDNS,* ULONG * *pSecondaryDNS,* ULONG * *pAuthentication,* CHAR * *pName,* CHAR * *pAPNName,* CHAR * *pUsername,* CHAR * *pPassword* )

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 - UMTS |
| *pPDPType[IN]* | • 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 |
| *pIPAddress[IN]* | • Preferred IPv4 addr to be assigned to device (optional) |
| *pPrimaryDNS[IN]* | • Primary DNS Ipv4 address preference (optional) |
| *pSecondaryDNS[IN]* | • Secondary DNS Ipv4 address preference (optional) |
| *pAuthentication[IN]* | • 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[IN]* | • profile Name (optional) |
| *pAPNName[IN]* | • 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. |
| *pUsername[IN]* | • Username used during network authentication (optional) |

| | |
|---|---|
| *pPassword[IN]* | • Password used during network authentication (optional) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout is 2 seconds.

### 9.49.5.25 ULONG SetDefaultProfileLTE ( ULONG *profileType,* ULONG * *pPDPType,* ULONG * *pIPAddressv4,* ULONG * *pPrimaryDNSv4,* ULONG * *pSecondaryDNSv4,* USHORT * *pIPAddressv6,* USHORT * *PrimaryDNSv6,* USHORT * *pSecondaryDNSv6,* ULONG * *pAuthentication,* CHAR * *pName,* CHAR * *pAPNName,* CHAR * *pUsername,* CHAR * *pPassword* )

Writes the default profile settings to the device. The default profile is used to establish an auto connect data session.

**Parameters**

| | |
|---|---|
| *profileType* | • Type of profile<br>– 0 - UMTS |
| *pPDPType[IN]* | • 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 |
| *pIPAddressv4[IN]* | • Preferred IPv4 address to be assigned to device (optional) |
| *pPrimaryDNSv4[IN]* | • Primary DNS Ipv4 address preference (optional) |
| *pSecondaryDNSv4[IN]* | • Secondary DNS Ipv4 address preference (optional) |
| *pIPAddressv6[IN]* | • Preferred IPv6 address to be assigned to device (optional) |
| *pPrimaryDNSv6[IN]* | • Primary DNS Ipv6 address preference (optional) |

| | |
|---|---|
| *pSecondaryDNSv6[IN]* | • Secondary DNS Ipv6 address preference (optional) |
| *pAuthentication[IN]* | • 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[IN]* | • profile Name (optional) |
| *pAPNName[IN]* | • 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 |
| *pUsername[IN]* | • Username used during network authentication (optional) |
| *pPassword[IN]* | • Password used during network authentication (optional) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Deprecated, please use SetDefaultProfileLTEV2 instead
Technology Supported: LTE
Timeout: 2 seconds

### 9.49.5.26 ULONG SetDefaultProfileLTEV2 ( ULONG *profileType*, ULONG * *pPDPType*, ULONG * *pIPAddressv4*, ULONG * *pPrimaryDNSv4*, ULONG * *pSecondaryDNSv4*, USHORT * *pIPAddressv6*, USHORT * *PrimaryDNSv6*, USHORT * *pSecondaryDNSv6*, ULONG * *pAuthentication*, CHAR * *pName*, CHAR * *pAPNName*, CHAR * *pUsername*, CHAR * *pPassword* )

Writes the default profile settings to the device. The default profile is used to establish an auto connect data session.

**Parameters**

| | |
|---|---|
| *profileType* | • Type of profile<br>– 0 - UMTS |
| *pPDPType[IN]* | • 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 |
| *pIPAddressv4[IN]* | • Preferred IPv4 address to be assigned to device (optional) |
| *pPrimaryDNSv4[IN]* | • Primary DNS Ipv4 address preference (optional) |
| *pSecondaryDNSv4[IN]* | • Secondary DNS Ipv4 address preference (optional) |
| *pIPAddressv6[IN]* | • Preferred IPv6 addr to be assigned to device (optional) |
| *pPrimaryDNSv6[IN]* | • Primary DNS Ipv6 address preference (optional) |
| *pSecondaryDNSv6[IN]* | • Secondary DNS Ipv6 address preference (optional) |
| *pAuthentication[IN]* | • 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[IN]* | • profile Name (optional) |
| *pAPNName[IN]* | • 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 |

| | |
|---|---|
| *pUsername[IN]* | • Username used during network authentication (optional) |
| *pPassword[IN]* | • Password used during network authentication (optional) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: LTE
Timeout: 2 seconds
Replaces deprecated Function SetDefaultProfileLTE

#### 9.49.5.27 ULONG SetDefaultProfileNum ( BYTE *profile_type,* BYTE *profile_family,* BYTE *profile_index* )

This API to Set default profile number

**Parameters**

| | |
|---|---|
| *profile_type* | [IN]<br>• 0 - 3GPP<br>• 1 - 3GPP2 |
| *profile_family* | [IN]<br>• 0 - Embedded<br>• 1 - Tethered |
| *profile_index* | [IN] |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.28 ULONG SetMobileIP ( ULONG *mode* )

Sets the current mobile IP setting.

**Parameters**

| | |
|---|---|
| *mode[IN]* | • Mobile IP setting<br>– 0 - Mobile IP off (simple IP only)<br>– 1 - Mobile IP preferred<br>– 2 - Mobile IP only |

**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
Timeout: 2 seconds

### 9.49.5.29 ULONG SetMobileIPParameters ( CHAR * *pSPC*, ULONG * *pMode*, BYTE * *pRetryLimit*, BYTE * *pRetryInterval*, BYTE * *pReRegPeriod*, BYTE * *pReRegTraffic*, BYTE * *pHAAuthenticator*, BYTE * *pHA2002bis* )

Sets the specified mobile IP parameters.

**Parameters**

| | |
|---|---|
| *pSPC[IN]* | • NULL-terminated string representing six digit service programming code. |
| *pMode[IN]* | • Mode to be set (optional)<br>– 0 - Mobile IP off (simple IP only)<br>– 1 - Mobile IP preferred<br>– 2 - Mobile IP only |
| *pRetryLimit[IN]* | • Registration retry attempt limit (optional) |
| *pRetryInterval[IN]* | • Registration retry attempt interval used to determine the time between registration attempts (optional) |
| *pReRegPeriod[IN]* | • Period (in minutes) to attempt re-registration before current registration expires (optional) |
| *pReRegTraffic[IN]* | • Re-registration only if traffic since last attempt (optional)<br>– Zero - Disabled<br>– NonZero - Enabled |

| | |
|---|---|
| *pHA-Authenticator[IN]* | • MH-HA authenticator calculator (optional)<br>– Zero - Disabled<br>– NonZero - Enabled |
| *pHA2002bis[IN]* | • MH-HA RFC 2002bis authentication instead of RFC2002 (optional)<br>– Zero - Disabled<br>– NonZero - Enabled |

**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
Device Supported: None
Timeout: 2 seconds

### 9.49.5.30 ULONG SetMobileIPProfile ( CHAR * *pSPC*, BYTE *index*, BYTE * *pEnabled*, ULONG * *pAddress*, ULONG * *pPrimaryHA*, ULONG * *pSecondaryHA*, BYTE * *pRevTunneling*, CHAR * *pNAI*, ULONG * *pHASPI*, ULONG * *pAAASPI*, CHAR * *pMNHA*, CHAR * *pMNAAA* )

Sets the mobile IP parameters.

**Parameters**

| | |
|---|---|
| *pSPC[IN]* | • Six digit service programming code string |
| *index[IN]* | • Index of the profile to modify |
| *pEnabled[IN]* | • (Optional) Enable profile? 0 - Disabled Nonzero - Enabled |
| *pAddress[IN]* | • (Optional) Home IPv4 address |
| *pPrimaryHA[IN]* | • (Optional) Primary home agent IPv4 address |
| *pSecondaryHA[IN]* | • (Optional) Secondary home agent IPv4 address |
| *pRevTunneling[IN]* | • (Optional) Enable reverse tunneling? 0 - Disabled Nonzero - Enabled |

| | |
|---|---|
| *pNAI[IN]* | • (Optional) Network access identifier string |
| *pHASPI[IN]* | • (Optional) Home agent security parameter index |
| *pAAASPI[IN]* | • (Optional) AAA server security parameter index |
| *pMNHA[IN]* | • (Optional) MN-HA key string |
| *pMNAAA[IN]* | • (Optional) MN-AAA key string |

**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
Timeout: 2 seconds

#### 9.49.5.31 ULONG SetMuxID ( BYTE *instance* )

This API is used for QMAP feature, it binds the MUX id before starting a data session or registering a WDS indcation, it is mandatory to do this for QMAP feature.

**Parameters**

| | |
|---|---|
| *instance* | [IN]<br>• QMI Instance ID, value 0-8, it will be mapped to mux id 0x80-0x88 by this API |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.32 ULONG SLQSAutoConnect ( struct slqsautoconnect * *pacreq* )

Returns auto connect settings

**Parameters**

| | |
|---|---|
| *slqsautoconnect[-IN]* | • SLQS auto connect settings |

**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/UMTS
Device Supported: MC83x5, MC7700
Timeout: 2 seconds

#### 9.49.5.33 ULONG SLQSCreateProfile ( struct CreateProfileIn * *pReq,* struct CreateProfileOut * *pResp* )

Create a new profile with the specified parameters. Note that some firmware versions do not support the optional Profile ID parameter. In this case an error will be returned and the caller can subsequently create a profile by specifying a NULL pointer for the Profile ID parameter. The Profile ID pertaining to the newly created profile is returned in the response structure (pResp).

**Parameters**

| | |
|---|---|
| *pReq[IN]* | • SLQS Create profile Information |
| *pResp[OUT]* | • SLQS profile identifier 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: UMTS
Device Supported: MC83x5, MC7700
Timeout: 2 seconds

#### 9.49.5.34 ULONG SLQSDeleteProfile ( struct SLQSDeleteProfileParams * *pProfileToDelete,* WORD * *pExtendedErrorCode* )

Deletes a configured profile stored on the device. The deletion of a profile does not affect profile index assignments.

**Parameters**

| | |
|---|---|
| *pProfileToDelete[IN]* | • Information about the profile to be deleted.<br>• See SLQSDeleteProfileParams for more details. |
| *pExtendedErrorCode[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. |

**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

#### 9.49.5.35 ULONG SLQSGet3GPPConfigItem ( slqs3GPPConfigItem * *pSLQS3GPPConfigItem* )

Reads the 3gpp configuration item.

**Parameters**

| | |
|---|---|
| *pSLQS3GPPConfigItem[OUT]* | • See slqs3GPPConfigItem 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: UMTS/LTE
Timeout: 2 seconds

#### 9.49.5.36 ULONG SLQSGetByteTotals ( struct WdsByteTotals * *pByteTotals* )

This API request the device to fetch ByteTotals for IPV4 and IPV6.

**Parameters**

| | |
|---|---|
| *pByteTotals[IN/-OUT]* | • See WdsByteTotals for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.37 ULONG SLQSGetConnectionRate ( struct WdsConnectionRate * *pConnectionRate* )

This API request the device to fetch ConnectionRate. It can be used for both mono and multiple PDN use case.

**Parameters**

| | |
|---|---|
| *pConnection-Rate[IN/OUT]* | • See WdsConnectionRate 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: UMTS/CDMA
Device Supported: MC77XX
Timeout: 2 seconds

#### 9.49.5.38 ULONG SLQSGetCurrDataSystemStat ( CurrDataSysStat * *pCurrDataSysStat* )

This API request the device to fetch current data system status.

**Parameters**

| | |
|---|---|
| *pCurrDataSys-Stat[IN/OUT]* | • See CurrDataSysStat for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 5 seconds\n
```

#### 9.49.5.39 ULONG SLQSGetCurrentChannelRate ( WDSSWICurrentChannelRates * *pRates*, BYTE *instance* )

This API Queries current bitrate of a packet data connection.

**Parameters**

| | |
|---|---|
| *pRates* | [IN]<br>• See WDSSWICurrentChannelRates 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 feature depends on custom feature setting IPCHANNELRATEEN which can be set via SLQSSetCustFeaturesV2

Timeout: 2 seconds

#### 9.49.5.40 ULONG SLQSGetDataBearerTechnology ( QmiWDSDataBearers * *pDataBearers*, BYTE *instance* )

Retrieves the data bearer technology values for current and/or last data calls. The device must be in a data call for this function to execute successfully.

**Parameters**

| | |
|---|---|
| *qmiWDSDataBearers[OUT]* | • Indicates the current and the last call data bearer technology. Should not be NULL, on input. |
| *instance* | • PDP instance |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Technology Supported: UMTS/CDMA
Device Supported: MC7750,GOBI,MC7700
Timeout: 2 seconds

#### 9.49.5.41 ULONG SLQSGetDataBearerTechnologyExt ( DataBearerTechExt * *pDataBearerTech*, BYTE *instance* )

This API Get Data Bearer Technology. This is a new API to replace API GetDataBearerTechnology(). see the description of GetDataBearerTechnology() to get more information

**Parameters**

| | |
|---|---|
| *pDataBearer-Tech* | [IN]<br>• See DataBearerTechExt for more information |
| *instance* | [IN]<br>• PDP instance |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.42 ULONG SLQSGetDUNCallInfo ( getDUNCallInfoReq * *pGetDUNCallInfoReq*, getDUNCallInfoResp * *pGetDUNCallInfoResp* )

This API queries the current modem connection status.

**Parameters**

| | |
|---|---|
| *pGetDUNCall-InfoReq[IN]* | • See getDUNCallInfoReq for more information |
| *pGetDUNCall-InfoResp[OUT]* | • See getDUNCallInfoResp 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: 5 seconds

#### 9.49.5.43 ULONG SLQSGetPacketStatistics ( struct WdsPktStatisticsReq * *pStatMask,* struct WdsPktStatisticsResp * *pPktStatistics* )

This API request the device to fetch current packet transfer counter values from the device

**Parameters**

| | |
|---|---|
| *pStatMask[IN]* | – See WdsPktStatisticsReq for more information |
| *pPktStatistics[OUT]* | – See WdsPktStatisticsResp for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 5 seconds\n
```

#### 9.49.5.44 ULONG SLQSGetProfile ( ULONG *profileType,* BYTE *profileId,* ULONG * *pPDPType,* ULONG * *pIPAddress,* ULONG * *pPrimaryDNS,* ULONG * *pSecondaryDNS,* ULONG * *pAuthentication,* BYTE *nameSize,* CHAR * *pName,* BYTE *apnSize,* CHAR * *pAPNName,* BYTE *userSize,* CHAR * *pUsername,* WORD * *pExtendedErrorCode* )

Reads the profile settings from the device for the specified profile id.

**Parameters**

| | |
|---|---|
| *profileType* | • Type of profile<br>– 0 - UMTS |
| *profileId* | • Index of the configured profile for which settings are read<br>– Value between 1 - 16 (EM/MC73xx or earlier)<br>– Value between 1 - 24 (EM/MC74xx onwards) |
| *pPDPType[OUT]* | • 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 |
| *pIPAddress[OUT]* | • Preferred IPv4 address to be assigned to device |
| *pPrimaryDNS[OUT]* | • Primary DNS Ipv4 address preference |

| | |
|---|---|
| *pSecondaryDNS[OUT]* | • Secondary DNS Ipv4 address preference |
| *pAuthentication[OUT]* | • 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. |
| *nameSize* | • Maximum number of characters (including NULL terminator) that profile name array can contain. |
| *pName[OUT]* | • Profile name |
| *apnSize* | • Maximum number of characters (including NULL terminator) that APN name array can contain |
| *pAPNName[OUT]* | • 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. |
| *pUsername[OUT]* | • Username used during network authentication |
| *pExtendedErrorCode* | • 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. |

**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

#### 9.49.5.45 ULONG SLQSGetProfileSettings ( GetProfileSettingIn * *pReq*, GetProfileSettingOut * *pResp* )

Retrieves a profile(3GPP/3GPP2) with the specified parameters.

**Parameters**

| | |
|---|---|
| *pReq[IN]* | • details of the profile to be fetched<br>• See GetProfileSettingIn for more information |
| *pResp[OUT]* | • The profile settings and/or extended error code returned by the device based on input parameters.<br>• See GetProfileSettingOut 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

#### 9.49.5.46 ULONG SLQSGetRuntimeSettings ( struct WdsRunTimeSettings * *pRunTimeSettings* )

Returns the packet data session settings currently in use.

**Parameters**

| | |
|---|---|
| *pRunTime-Settings[OUT]* | • SLQS Runtime Settings 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

#### 9.49.5.47 ULONG SLQSGetSessionState ( ULONG * *pStateV4,* ULONG * *pStateV6,* BYTE *instance* )

Returns the state of the current packet data session.

**Parameters**

| | |
|---|---|
| *pStateV4[OUT]* | • Current link status for IPV4 Session<br>– 1 - DISCONNECTED<br>– 2 - CONNECTED<br>– 3 - SUSPENDED (not supported)<br>– 4 - AUTHENTICATING |
| *pStateV6[OUT]* | • Current link status for IPV6 Session<br>– 1 - DISCONNECTED<br>– 2 - CONNECTED<br>– 3 - SUSPENDED (not supported)<br>– 4 - AUTHENTICATING |
| *instance* | • PDP instance |

**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

#### 9.49.5.48 ULONG SLQSModifyProfile ( struct ModifyProfileIn * *pReq,* struct ModifyProfileOut * *pResp* )

Modify a profile(3GPP/3GPP2) with the specified parameters.

**Parameters**

| | |
|---|---|
| *pReq[IN]* | • Contains parameters which can be modified |
| *pResp[OUT]* | • Contains parameters which indicates modification success or failure |

**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

#### 9.49.5.49 ULONG SLQSResetPacketStatics ( )

This API request the device to reset packet data transfer statistics.

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.50 ULONG SLQSSet3GPPConfigItem ( slqs3GPPConfigItem * *pSLQS3GPPConfigItem* )

Sets the 3gpp configuration item.

**Parameters**

| | |
|---|---|
| *pSLQS3GPP-ConfigItem[IN]* | • See slqs3GPPConfigItem 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: UMTS/LTE
Timeout: 2 seconds

#### 9.49.5.51 ULONG SLQSSetProfile ( ULONG *profileType,* BYTE *profileId,* ULONG * *pPDPType,* ULONG * *pIPAddress,* ULONG * *pPrimaryDNS,* ULONG * *pSecondaryDNS,* ULONG * *pAuthentication,* CHAR * *pName,* CHAR * *pAPNName,* CHAR * *pUsername,* CHAR * *pPassword* )

Writes the profile settings for the specified profile Id.

**Parameters**

| | |
|---|---|
| *profileType* | • Type of profile<br>– 0 - UMTS |
| *profileId* | • Profile number to be modified<br>– Value between 1 - 16 (EM/MC73xx or earlier)<br>– Value between 1 - 24 (EM/MC74xx onwards) |
| *pPDPType[IN]* | • 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 |
| *pIPAddress[IN]* | • Preferred IPv4 address to be assigned to device (optional) |
| *pPrimaryDNS[IN]* | • Primary DNS Ipv4 address preference (optional) |
| *pSecondaryDNS[IN]* | • Secondary DNS Ipv4 address preference (optional) |
| *pAuthentication[IN]* | • 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[IN]* | • profile Name (optional) |
| *pAPNName[IN]* | • 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. |
| *pUsername[IN]* | • Username used during network authentication (optional) |

| | |
|---|---|
| *pPassword[IN]* | • Password used during network authentication (optional) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout is 2 seconds.

#### 9.49.5.52 ULONG SLQSSGetDHCPv4ClientConfig ( WdsDHCPv4Config * *pReqResp* )

This API gets the DHCP Client V4 Configuration.

**Parameters**

| | |
|---|---|
| *pReq* | [IN]<br>• See WdsDHCPv4Config for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.53 ULONG SLQSSGetLoopback ( WDSGetLoopbackData * *data* )

This API to Get the value of loopback mode and multiplier.

**Parameters**

| | |
|---|---|
| *pReq* | [IN]<br>• See WDSGetLoopbackData for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.54 ULONG SLQSSSetDHCPv4ClientConfig ( WdsDHCPv4Config * *pReq* )

This API sets the DHCP Client V4 Configuration.

**Parameters**

| | |
|---|---|
| *pReq* | [IN]<br>• See WdsDHCPv4Config for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.55 ULONG SLQSSSetLoopback ( WDSSetLoopbackData * *pReq* )

This API to Enable/disable Data Loopback Mode and set the value of loopback multiplier.

**Parameters**

| | |
|---|---|
| *pReq* | [IN]<br>• See WDSSetLoopbackData for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

```
Timeout: 2 seconds\n
```

#### 9.49.5.56 ULONG SLQSStartStopDataSession ( struct ssdatasession_params * *pin* )

Starts or stops a 3GPP/3GPP2 data session on a preconfigured profile. To set the IP family for the data session, execute SLQSSetIPFamilyPreference prior to calling this API.

**Parameters**

| | |
|---|---|
| *pin[IN]* | • ssdatasession_params structure<br>• See ssdatasession_params for more details |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 5 Minutes
Use SLQSSetProfile to configure 3GPP profiles

#### 9.49.5.57 ULONG SLQSWdsGoActive ( void )

Forces the device to immediately reestablish the traffic channel on the serving radio interface

**Parameters**

| | |
|---|---|
| *None* | |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

This channel can go dormant any time after it has been reactivated. There is no assurance that the channel remains active for any guaranteed period. Timeout: 5 seconds

#### 9.49.5.58 ULONG SLQSWdsGoDormant ( void )

Forces the device to immediately drop the traffic channel on the serving radio interface

**Parameters**

| | |
|---|---|
| *None* | |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

This channel can be reactivated as soon as data is sent over the network interface. There is no assurance that the channel remains dormant for any guaranteed period. Timeout: 5 seconds

#### 9.49.5.59 ULONG SLQSWdsSetEventReport ( wdsSetEventReportReq * *pSetEventReportReq* )

This API sets the wireless data connection state reporting conditions for the requesting control point.

**Parameters**

| | |
|---|---|
| *pSetEventReportReq[IN]* | • See wdsSetEventReportReq 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
The control point event reporting state variables are modified to reflect the settings indicated in the request message. The service maintains a set of state variables for each control point. Relevant wireless data connection state changes are communicated to the registered WDS control point via the SLQSWdsSetEventReportCallBack. The AT command equivalents to this command are AT+CMER, AT+CIND, and AT+CIEV

#### 9.49.5.60 ULONG SLQSWdsSwiPDPRuntimeSettings ( swiPDPRuntimeSettingsReq * *pPDPRuntimeSettingsReq*, swiPDPRuntimeSettingsResp * *pPDPRuntimeSettingsResp* )

This API requests the device to return the active PDP context associated with a context id.

**Parameters**

| | |
|---|---|
| *pPDPRuntimeSettingsReq[IN]* | • See swiPDPRuntimeSettingsReq for more information |
| *pPDPRuntimeSettingsResp[OUT]* | • See swiPDPRuntimeSettingsResp 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: UMTS/CDMA
Device Supported: MC77XX
Timeout: 2 seconds
The AT command equivalent to this command is AT+CGCONTRDP

#### 9.49.5.61 BOOL WDS_IsGobiDevice ( )

## 9.50 qaNasGetRFBandInfo.h File Reference

### Data Structures

- struct QmiNasGetRFBandInfoResp

### Enumerations

- enum eQMI_NAS_GET_RF_INFO_RESP {
  eTLV_RF_BAND_INFO = 0x01,
  eTLV_RF_DEDICATED_BAND_INFO = 0x10,
  eTLV_RF_EXT_FORMAT_BAND_INFO = 0x11,
  eTLV_RF_BANDWIDTH_INFO = 0x12,
  eTLV_LTE_OPERATIONAL_MODE = 0x13 }

### Functions

- enum eQCWWANError PkQmiNasGetRFBandInfo (WORD *pMlength, BYTE *pBuffer)
- enum eQCWWANError UpkQmiNasGetRFBandInfo (BYTE *pMdmResp, struct QmiNasGetRFBandInfoResp *pApiResp)

### 9.50.1 Enumeration Type Documentation

#### 9.50.1.1 enum eQMI_NAS_GET_RF_INFO_RESP

Enumerator

***eTLV_RF_BAND_INFO***

***eTLV_RF_DEDICATED_BAND_INFO***

***eTLV_RF_EXT_FORMAT_BAND_INFO***

***eTLV_RF_BANDWIDTH_INFO***

***eTLV_LTE_OPERATIONAL_MODE***

### 9.50.2 Function Documentation

#### 9.50.2.1 enum eQCWWANError PkQmiNasGetRFBandInfo ( WORD * *pMlength*, BYTE * *pBuffer* )

#### 9.50.2.2 enum eQCWWANError UpkQmiNasGetRFBandInfo ( BYTE * *pMdmResp*, struct QmiNasGetRFBandInfoResp * *pApiResp* )

## 9.51 qaNasPerformNetworkScan.h File Reference

### Data Structures

- struct QmiNas3GppNetworkInfo
- struct QmiNasPerformNetworkScanResp

### Macros

- #define QMI_NAS_NETSTATUS_MASK 0x03
- #define QMI_NAS_MAX_INSTANCES 20
- #define INDEX_ZERO 0
- #define ROAMING_INDEX 2
- #define FORBIDDEN_INDEX 4
- #define PREFERRED_INDEX 6
- #define MAX_DESCRIPTION_LENGTH 255

### Enumerations

- enum eQMI_NAS_PERFORM_NETWORK_SCAN_RESP { eTLV_3GPP_NETWORK_INFO = 0x10 }

### Functions

- enum eQCWWANError PkQmiNasPerformNetworkScan (WORD *pMlength, BYTE *pParamField)
- enum eQCWWANError UpkQmiNasPerformNetworkScan (BYTE *pMdmResp, struct QmiNasPerformNetworkScanResp *pAipResp)

## 9.51.1 Macro Definition Documentation

#### 9.51.1.1 #define FORBIDDEN_INDEX 4

#### 9.51.1.2 #define INDEX_ZERO 0

#### 9.51.1.3 #define MAX_DESCRIPTION_LENGTH 255

#### 9.51.1.4 #define PREFERRED_INDEX 6

#### 9.51.1.5 #define QMI_NAS_MAX_INSTANCES 20

#### 9.51.1.6 #define QMI_NAS_NETSTATUS_MASK 0x03

#### 9.51.1.7 #define ROAMING_INDEX 2

## 9.51.2 Enumeration Type Documentation

#### 9.51.2.1 enum eQMI_NAS_PERFORM_NETWORK_SCAN_RESP

Enumerator

***eTLV_3GPP_NETWORK_INFO***

## 9.51.3 Function Documentation

#### 9.51.3.1 enum eQCWWANError PkQmiNasPerformNetworkScan ( WORD * *pMlength*, BYTE * *pParamField* )

#### 9.51.3.2 enum eQCWWANError UpkQmiNasPerformNetworkScan ( BYTE * *pMdmResp*, struct QmiNasPerformNetworkScanResp * *pAipResp* )

# 9.52 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 }

## 9.52.1 Enumeration Type Documentation

### 9.52.1.1 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

#### 9.52.1.2 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***

## 9.53 qos.h File Reference

### Data Structures

- struct unpack_qos_SLQSQosGetNetworkStatus_t
- struct pack_qos_SLQSQosSwiReadApnExtraParams_t
- struct unpack_qos_SLQSQosSwiReadApnExtraParams_t

- struct pack_qos_SLQSQosSwiReadDataStats_t
- struct unpack_QosFlowStat_t
- struct unpack_qos_SLQSQosSwiReadDataStats_t
- struct unpack_qos_SLQSSetQosNWStatusCallback_ind_t
- struct unpack_qos_SLQSSetQosStatusCallback_ind_t
- struct unpack_qos_SLQSSetQosPriEventCallback_ind_t
- struct pack_qos_SLQSSetQosEventCallback_t
- struct unpack_qos_QosFlowInfoState_t
- struct unpack_qos_dataRate_t
- struct unpack_qos_tokenBucket_t
- struct unpack_qos_pktErrRate_t
- struct unpack_qos_swiQosFlow_t
- struct unpack_qos_IPv4Addr_t
- struct unpack_qos_Tos_t
- struct unpack_qos_IPv6Addr_t
- struct unpack_qos_IPv6TrafCls_t
- struct unpack_qos_Port_t
- struct unpack_qos_swiQosFilter_t
- struct unpack_qos_QosFlowInfo_t
- struct unpack_qos_SLQSSetQosEventCallback_ind_t
- struct qos_BindDataPortPeripheralEndPointID_t
- struct qos_BindDataPortMuxID_t
- struct qos_BindDataPortSIODataPort_t
- struct pack_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)

- 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)

### 9.53.1 Macro Definition Documentation

#### 9.53.1.1 #define LITEQMI_MAX_QOS_FILTERS 25

#### 9.53.1.2 #define LITEQMI_MAX_QOS_FLOW_PER_APN_STATS 10

#### 9.53.1.3 #define LITEQMI_MAX_QOS_FLOWS 8

### 9.53.2 Function Documentation

#### 9.53.2.1 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

#### 9.53.2.2 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

### 9.53.2.3 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

#### 9.53.2.4 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

#### 9.53.2.5 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

#### 9.53.2.6 int unpack_qos_BindDataPort ( uint8_t * *pResp*, uint16_t *respLen* )

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

#### 9.53.2.7 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

#### 9.53.2.8 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

#### 9.53.2.9 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

#### 9.53.2.10 int unpack_qos_SLQSSetQosEventCallback ( uint8_t * *pResp*, uint16_t *respLen* )

Function to unpack enable QoS event indications command's response This maps to SLQSSetQosEventCallback

**Parameters**

| | | |
|---|---|---|
| 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

#### 9.53.2.11 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

#### 9.53.2.12 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

#### 9.53.2.13 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

### 9.53.2.14 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

## 9.54 rms.h File Reference

### Data Structures

- struct unpack_rms_GetSMSWake_t
- struct pack_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)

### 9.54.1 Macro Definition Documentation

#### 9.54.1.1 #define __LITEQMI_RMS_H__

### 9.54.2 Function Documentation

#### 9.54.2.1 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

#### 9.54.2.2 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

#### 9.54.2.3 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

#### 9.54.2.4 int unpack_rms_SetSMSWake ( uint8_t * *pResp*, uint16_t *respLen* )

Set SMS Wake 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

## 9.55 sar.h File Reference

### Data Structures

- struct unpack_sar_SLQSGetRfSarState_t
- struct pack_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)

### 9.55.1 Function Documentation

#### 9.55.1.1 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

#### 9.55.1.2 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

#### 9.55.1.3 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

#### 9.55.1.4 int unpack_sar_SLQSSetRfSarState ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the specified RF SAR 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

## 9.56 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 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 pack_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 sms_routeEntry
- struct sms_setRoutesReq
- struct pack_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 pack_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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)
- 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)

## 9.56.1 Macro Definition Documentation

### 9.56.1.1 #define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255

### 9.56.1.2 #define MAX_MS_TRANSFER_ROUTE_MSG 256

### 9.56.1.3 #define MAX_MSC_ADDRESS_SIZE 256

### 9.56.1.4 #define MAX_MSE_TWS_MSG 1254

### 9.56.1.5 #define MAX_SMS_LIST_SIZE 255

### 9.56.1.6 #define MAX_SMS_MESSAGE_SIZE 2048

### 9.56.1.7 #define SMS_CONFIG_LEN 0x05

### 9.56.1.8 #define SMS_MAX_SMS_ROUTES 0x0A

### 9.56.1.9 #define SMS_NUM_OF_SET 0xFF

### 9.56.1.10 #define SMSC_TYPE_LEN 0x03

## 9.56.2 Enumeration Type Documentation

### 9.56.2.1 enum eqmiCbkSetStatus

**Enumerator**

***LITEQMI_QMI_CBK_PARAM_RESET***

***LITEQMI_QMI_CBK_PARAM_SET***

***LITEQMI_QMI_CBK_PARAM_NOCHANGE***

## 9.56.3 Function Documentation

### 9.56.3.1 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

#### 9.56.3.2 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

#### 9.56.3.3 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

#### 9.56.3.4 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

### 9.56.3.5 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

### 9.56.3.6 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

#### 9.56.3.7 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

#### 9.56.3.8 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

#### 9.56.3.9 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

#### 9.56.3.10 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

#### 9.56.3.11 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

#### 9.56.3.12 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

#### 9.56.3.13 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

#### 9.56.3.14 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

#### 9.56.3.15 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

#### 9.56.3.16 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

#### 9.56.3.17 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

#### 9.56.3.18 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

### 9.56.3.19 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

### 9.56.3.20 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

#### 9.56.3.21 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

#### 9.56.3.22 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

#### 9.56.3.23 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

#### 9.56.3.24 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

#### 9.56.3.25 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

#### 9.56.3.26 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

### 9.56.3.27 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

### 9.56.3.28 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

### 9.56.3.29 int unpack_sms_SetSMSCAddress ( uint8_t * *pResp*, uint16_t *respLen* )

Set the SMS center address 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

#### 9.56.3.30 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

#### 9.56.3.31 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

#### 9.56.3.32 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

#### 9.56.3.33 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

#### 9.56.3.34 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

#### 9.56.3.35 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

#### 9.56.3.36 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

#### 9.56.3.37 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

#### 9.56.3.38 int unpack_sms_SLQSModifySMSStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSModifySMSStatus_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

#### 9.56.3.39 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

#### 9.56.3.40 int unpack_sms_SLQSSendAsyncSMS ( uint8_t * *pResp*, uint16_t *respLen* )

Sends an SMS message for immediate over-the-air transmission 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

#### 9.56.3.41 int unpack_sms_SLQSSetIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the registration state of different WMS indications 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

#### 9.56.3.42 int unpack_sms_SLQSSetSmsBroadcastActivation ( uint8_t * *pResp*, uint16_t *respLen* )

Enables or disables the reception of broadcast SMS messages 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

#### 9.56.3.43 int unpack_sms_SLQSSetSmsBroadcastConfig ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the information about the SMS BroadcastConfiguration 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

#### 9.56.3.44 int unpack_sms_SLQSSetSmsStorage ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the SMS Storage on the device 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

### 9.56.3.45 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

### 9.56.3.46 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

### 9.56.3.47 int unpack_sms_SLQSSmsSetRoutes ( uint8_t * *pResp*, uint16_t *respLen* )

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 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.56.3.48 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

#### 9.56.3.49 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

#### 9.56.3.50 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

#### 9.56.3.51 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

#### 9.56.3.52 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

## 9.57 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

## 9.57.1 Detailed Description

SWI data types.

## 9.57.2 Macro Definition Documentation

#### 9.57.2.1 #define QMI_NO_LTE_FW_SUPPORT 0

#### 9.57.2.2 #define QMI_TLV_PLACEHOLDER 0x8F

#### 9.57.2.3 #define SWI_API

#### 9.57.2.4 #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.

## 9.57.3 Typedef Documentation

#### 9.57.3.1 typedef int BOOL

#### 9.57.3.2 typedef unsigned char BYTE

#### 9.57.3.3 typedef char CHAR

#### 9.57.3.4 typedef float FLOAT

#### 9.57.3.5 typedef signed int INT32

#### 9.57.3.6 typedef signed char INT8

#### 9.57.3.7 typedef const char* LPCSTR

#### 9.57.3.8 typedef signed short SHORT

#### 9.57.3.9 typedef unsigned int ULONG

#### 9.57.3.10 typedef unsigned long long ULONGLONG

#### 9.57.3.11 typedef unsigned short USHORT

#### 9.57.3.12 typedef unsigned short WORD

## 9.58 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

### 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)
- 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)

### 9.58.1 Function Documentation

#### 9.58.1.1 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

#### 9.58.1.2 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

#### 9.58.1.3 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

#### 9.58.1.4 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

### 9.58.1.5 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

### 9.58.1.6 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

#### 9.58.1.7 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

#### 9.58.1.8 int unpack_swidms_SLQSSwiDmsSetUsbNetNum ( uint8_t * *pResp*, uint16_t *respLen* )

unpack function to set usb net numbers for QMAP configuration

**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

## 9.59 swiloc.h File Reference

### Data Structures

- struct unpack_swiloc_SwiLocGetAutoStart_t
- struct pack_swiloc_SwiLocSetAutoStart_t

### Functions

- int pack_swiloc_SwiLocGetAutoStart (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swiloc_SwiLocGetAutoStart (uint8_t *pResp, uint16_t respLen, unpack_swiloc_SwiLocGetAutoStart_t *pOutput)
- int pack_swiloc_SwiLocSetAutoStart (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiloc_SwiLocSetAutoStart_t *reqArg)
- int unpack_swiloc_SwiLocSetAutoStart (uint8_t *pResp, uint16_t respLen)

### 9.59.1 Function Documentation

#### 9.59.1.1 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

#### 9.59.1.2 int pack_swiloc_SwiLocSetAutoStart ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiloc_SwiLocSetAutoStart_t * reqArg )

Set Auto Start pack

**Parameters**

| | | |
|---|---|---|
| in | *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

#### 9.59.1.3 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

#### 9.59.1.4 int unpack_swiloc_SwiLocSetAutoStart ( uint8_t * *pResp*, uint16_t *respLen* )

Set Auto Start 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

## 9.60 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)

## 9.60.1 Macro Definition Documentation

### 9.60.1.1 #define LITEQMI_MAX_SWIOMA_STR_LEN 255

## 9.60.2 Function Documentation

### 9.60.2.1 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

#### 9.60.2.2 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

#### 9.60.2.3 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

#### 9.60.2.4 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 SLQSOMADMGetSettings2

**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

#### 9.60.2.5 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

#### 9.60.2.6 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

#### 9.60.2.7 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

#### 9.60.2.8 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

#### 9.60.2.9 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

#### 9.60.2.10 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

### 9.60.2.11 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

### 9.60.2.12 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

#### 9.60.2.13 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

#### 9.60.2.14 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

#### 9.60.2.15 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

## 9.61 SWIWWANCMAPI.h File Reference

## 9.62 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)

### 9.62.1 Macro Definition Documentation

#### 9.62.1.1 #define MAX_DESCRIPTION_LENGTH 255

#### 9.62.1.2 #define MAX_ICCID_LENGTH 255

#### 9.62.1.3 #define MAX_NO_OF_APPLICATIONS 10

#### 9.62.1.4 #define MAX_NO_OF_SLOTS 5

#### 9.62.1.5 #define MAX_SLOTS_STATUS 255

#### 9.62.1.6 #define UIM_MAX_ACTIVE_PERS_FEATURES 12

#### 9.62.1.7 #define UIM_MAX_CONTENT_LENGTH 1024

#### 9.62.1.8 #define UIM_MAX_DESCRIPTION_LENGTH 255

#### 9.62.1.9 #define UIM_MAX_NO_OF_APPLICATIONS 10

#### 9.62.1.10 #define UIM_MAX_NO_OF_SLOTS 5

#### 9.62.1.11 #define UIM_UINT8_MAX_STRING_SZ 255

### 9.62.2 Function Documentation

#### 9.62.2.1 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

#### 9.62.2.2 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

#### 9.62.2.3 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

#### 9.62.2.4 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

#### 9.62.2.5 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

#### 9.62.2.6 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

#### 9.62.2.7 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

#### 9.62.2.8 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

#### 9.62.2.9 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

#### 9.62.2.10 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

### 9.62.2.11 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

### 9.62.2.12 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

#### 9.62.2.13 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

#### 9.62.2.14 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

#### 9.62.2.15 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

#### 9.62.2.16 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

#### 9.62.2.17 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

#### 9.62.2.18 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

### 9.62.2.19 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

### 9.62.2.20 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

#### 9.62.2.21 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

#### 9.62.2.22 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

#### 9.62.2.23 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

#### 9.62.2.24 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

#### 9.62.2.25 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

#### 9.62.2.26 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

#### 9.62.2.27 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

#### 9.62.2.28 int unpack_uim_SLQSUIMEventRegister ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMEvent-Register_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

#### 9.62.2.29 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

#### 9.62.2.30 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

#### 9.62.2.31 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

#### 9.62.2.32 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

#### 9.62.2.33 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

#### 9.62.2.34 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

#### 9.62.2.35 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

#### 9.62.2.36 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

#### 9.62.2.37 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

#### 9.62.2.38 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

#### 9.62.2.39 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

#### 9.62.2.40 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

#### 9.62.2.41 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

#### 9.62.2.42 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

#### 9.62.2.43 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

## 9.63 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)

### 9.63.1 Detailed Description

### 9.63.2 S1

- 3GPP Mobile Radio Interface Layer 3 Specification : Core Network Protocols; Stage 3 (Release 7)
- 3GPP TS 24.008 V7.0.0 (2005-06)

### 9.63.3 Macro Definition Documentation

#### 9.63.3.1 #define BARRING_PASSWORD_LENGTH 4

#### 9.63.3.2 #define MAX_VOICE_CALL_NO_LEN 81

#### 9.63.3.3 #define MAX_VOICE_DESCRIPTION_LENGTH 255

#### 9.63.3.4 #define MAXVOICEUSSDLENGTH 182

#### 9.63.3.5 #define VOICE_MAX_NO_OF_CALLS 20

### 9.63.4 Enumeration Type Documentation

#### 9.63.4.1 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***

### 9.63.5 Function Documentation

#### 9.63.5.1 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

#### 9.63.5.2 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

#### 9.63.5.3 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

#### 9.63.5.4 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

#### 9.63.5.5 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

#### 9.63.5.6 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

#### 9.63.5.7 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

#### 9.63.5.8 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

#### 9.63.5.9 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

#### 9.63.5.10 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

#### 9.63.5.11 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

#### 9.63.5.12 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

#### 9.63.5.13 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

#### 9.63.5.14 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

#### 9.63.5.15 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

### 9.63.5.16 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

### 9.63.5.17 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

### 9.63.5.18 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

#### 9.63.5.19 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

#### 9.63.5.20 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

#### 9.63.5.21 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

### 9.63.5.22 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

### 9.63.5.23 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

### 9.63.5.24 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

#### 9.63.5.25 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

#### 9.63.5.26 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

#### 9.63.5.27 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

#### 9.63.5.28 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

#### 9.63.5.29 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

#### 9.63.5.30 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

#### 9.63.5.31 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

#### 9.63.5.32 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

#### 9.63.5.33 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

#### 9.63.5.34 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

#### 9.63.5.35 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

#### 9.63.5.36 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

#### 9.63.5.37 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

#### 9.63.5.38 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

#### 9.63.5.39 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

#### 9.63.5.40 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

#### 9.63.5.41 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

#### 9.63.5.42 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

#### 9.63.5.43 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

#### 9.63.5.44 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

### 9.63.5.45 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

### 9.63.5.46 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

#### 9.63.5.47 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

#### 9.63.5.48 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

#### 9.63.5.49 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

### 9.63.5.50 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

### 9.63.5.51 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

### 9.63.5.52 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

#### 9.63.5.53 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

#### 9.63.5.54 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

#### 9.63.5.55 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

### 9.63.5.56 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

### 9.63.5.57 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

### 9.63.5.58 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

#### 9.63.5.59 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

#### 9.63.5.60 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

#### 9.63.5.61 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

### 9.63.5.62 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

### 9.63.5.63 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

### 9.63.5.64 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

#### 9.63.5.65 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

#### 9.63.5.66 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

#### 9.63.5.67 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

### 9.63.5.68 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

### 9.63.5.69 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

### 9.63.5.70 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

#### 9.63.5.71 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

#### 9.63.5.72 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

## 9.64 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)

### 9.64.1 Macro Definition Documentation

#### 9.64.1.1 #define BYT_STAT_STAT_MASK 0X000000C0

#### 9.64.1.2 #define IPV6_ADDRESS_ARRAY_SIZE 8

#### 9.64.1.3 #define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24

#### 9.64.1.4 #define PACK_WDS_IPV4 4

#### 9.64.1.5 #define PACK_WDS_IPV6 6

#### 9.64.1.6 #define WDS_DHCP_MAX_NUM_OPTIONS 30

#### 9.64.1.7 #define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */

### 9.64.2 Typedef Documentation

#### 9.64.2.1 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 |
| *SlqsProfile3GP-P2* | 3GPP2 profile<br>• See LibpackProfile3GPP2 |

### 9.64.3 Enumeration Type Documentation

#### 9.64.3.1 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***

### 9.64.4 Function Documentation

#### 9.64.4.1 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

#### 9.64.4.2 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

#### 9.64.4.3 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

#### 9.64.4.4 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

#### 9.64.4.5 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

#### 9.64.4.6 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

#### 9.64.4.7 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

#### 9.64.4.8 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

#### 9.64.4.9 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

#### 9.64.4.10 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

#### 9.64.4.11 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

#### 9.64.4.12 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

#### 9.64.4.13 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

#### 9.64.4.14 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

#### 9.64.4.15 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

#### 9.64.4.16 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

#### 9.64.4.17 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

#### 9.64.4.18 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

#### 9.64.4.19 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

#### 9.64.4.20 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

#### 9.64.4.21 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

#### 9.64.4.22 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

#### 9.64.4.23 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

#### 9.64.4.24 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

#### 9.64.4.25 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

#### 9.64.4.26 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

#### 9.64.4.27 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

#### 9.64.4.28 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

#### 9.64.4.29 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

#### 9.64.4.30 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

#### 9.64.4.31 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

#### 9.64.4.32 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

#### 9.64.4.33 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

#### 9.64.4.34 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

#### 9.64.4.35 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

#### 9.64.4.36 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

#### 9.64.4.37 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

#### 9.64.4.38 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

#### 9.64.4.39 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

#### 9.64.4.40 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

#### 9.64.4.41 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

#### 9.64.4.42 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

#### 9.64.4.43 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

#### 9.64.4.44 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

#### 9.64.4.45 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

#### 9.64.4.46 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

#### 9.64.4.47 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

#### 9.64.4.48 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

#### 9.64.4.49 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

#### 9.64.4.50 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

#### 9.64.4.51 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

#### 9.64.4.52 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

#### 9.64.4.53 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

#### 9.64.4.54 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

#### 9.64.4.55 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

#### 9.64.4.56 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

#### 9.64.4.57 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

#### 9.64.4.58 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

#### 9.64.4.59 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

#### 9.64.4.60 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

#### 9.64.4.61 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

#### 9.64.4.62 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

#### 9.64.4.63 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

#### 9.64.4.64 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

#### 9.64.4.65 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

#### 9.64.4.66 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

#### 9.64.4.67 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

#### 9.64.4.68 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

#### 9.64.4.69 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

#### 9.64.4.70 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

#### 9.64.4.71 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

#### 9.64.4.72 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

#### 9.64.4.73 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

#### 9.64.4.74 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

#### 9.64.4.75 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

#### 9.64.4.76 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

#### 9.64.4.77 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

#### 9.64.4.78 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

#### 9.64.4.79 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

#### 9.64.4.80 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

#### 9.64.4.81 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

#### 9.64.4.82 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

#### 9.64.4.83 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

#### 9.64.4.84 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

#### 9.64.4.85 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 |

#### 9.64.4.86 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

#### 9.64.4.87 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

#### 9.64.4.88 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

#### 9.64.4.89 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

#### 9.64.4.90 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

#### 9.64.4.91 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

#### 9.64.4.92 int unpack_wds_SLQSSetDHCPv4ClientConfig ( 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

#### 9.64.4.93 int unpack_wds_SLQSSetLoopback ( uint8_t * *pResp*, uint16_t *respLen* )

Enable/disable Data Loopback Mode and set the value of loopback multiplier unpack

**Parameters**

| | | |
|---|---|---|
| in | *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

#### 9.64.4.94 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

#### 9.64.4.95 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

#### 9.64.4.96 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

#### 9.64.4.97 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

#### 9.64.4.98 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

#### 9.64.4.99 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, 1369
_GetProfileSettingIn, 66
    ProfileID, 66
    ProfileType, 66
_GetProfileSettingOut, 66
    curProfile, 67
    pExtErrCode, 67
_MitigationDevInfo, 69
    deviceID, 69
    deviceIdLen, 69
_QmiSwiAvmsEventReportBinaryUpdateSessionInfo-
        Resp, 73
    bBinaryType, 75
    bSerity, 75
    bState, 75
    bUserInputRequest, 75
    szDescription, 75
    szName, 75
    szVersion, 75
    TlvPresent, 75
    ulPkgDownloadComplete, 75
    ulPkgDownloadSize, 75
    wDescriptionLength, 75
    wNameLength, 75
    wUpdateCompeteStatus, 75
    wUserInputTimeout, 75
    wVersionLength, 75
_QmiSwiAvmsEventReportConfigResp, 75
    bState, 76
    bUserInputRequest, 76
    szAlertMsg, 76
    TlvPresent, 76
    wAlertMsgLength, 76
    wUserInputTimeout, 76
_QmiSwiAvmsEventReportConnectionRequestResp,
        76
    bUserInputRequest, 77
    TlvPresent, 77
    wUserInputTimeout, 77
_QmiSwiAvmsEventReportDataSessionStatusResp, 77
    bType, 77
    TlvPresent, 77
    wErrorCode, 77
_QmiSwiAvmsEventReportHTTPStatusResp, 77
    TlvPresent, 78
    wHTTPStatus, 78
_QmiSwiAvmsEventReportNotificationResp, 79
    bNotification, 79
    TlvPresent, 79
    wSessionStatus, 80
_QmiSwiAvmsEventReportPackageIDResp, 80
    bPackageID, 80
    TlvPresent, 80
_QmiSwiAvmsEventReportRegStatusResp, 80
    bRegStatus, 80
    TlvPresent, 80
_QmiSwiAvmsEventReportSessionTypeResp, 80
    bType, 81
    TlvPresent, 81
_QmiSwiAvmsEventReportWAMSParamChangeResp,
        81
    TlvPresent, 81
    wWamsChangeMask, 81
_QmiSwiAvmsSetSettingsAPNInfo, 81
    bAPNLength, 82
    bPWDLength, 82
    bUnameLength, 82
    szAPN, 82
    szPWD, 82
    szUname, 82
_QmiSwiAvmsSetSettingsConnectionRetryTimers, 82
    Timers, 82
_QmiSwiAvmsSetSettingsPeriodInfo, 82
    ulMax, 82
    ulMin, 82
_SLQSAVMSSetSettingsReqPara, 85
    pAPNInfo, 87
    pAutoConnect, 87
    pAutoReboot, 87
    pConnectionRetryTimers, 87
    pFwAutoSDM, 87
    pFwPromptUpdate, 87
    pFwPromptdownload, 87
    pNotifStore, 87
    pPeroidInfo, 87
    pPollingTimer, 87
_SLQSAVMSSettings, 87
    pAPNInfo, 87
    pAutoConnect, 87
    pAutoReboot, 87
    pConnectionRetryTimers, 87
    pFwAutoSDM, 87
    pFwPromptUpdate, 87
    pFwPromptdownload, 87
    pNotificationStore, 87
    pOMADMEnabled, 88
    pPeroidsInfo, 88

pPollingTimer, 88
_SLQSAVMSSettingsAPNInfo, 88
bAPNLength, 88
bPWDLength, 88
bUnameLength, 88
pAPN, 88
pPWD, 88
pUname, 88
_SLQSAVMSSettingsConnectionRetryTimers, 88
Timers, 88
_SLQSAVMSSettingsPeriodsInfo, 88
max, 89
min, 89
_SLQSOMADMSessionInfo, 91
pDate, 93
pDateLength, 93
pPkgDescLength, 93
pPkgDescription, 93
pPkgName, 93
pPkgNameLength, 93
pRetryCount, 93
pSessionState, 93
pSessionType, 93
pSeverity, 93
pSource, 93
pSourceLength, 93
pStatus, 93
pTime, 93
pTimeLength, 93
pUpdateCompleteStatus, 93
_SLQSOMADMSettings, 93
pAutosdm, 95
pFOTAUpdate, 95
pFOTAdownload, 95
pFwAutoCheck, 95
pOMADMEnabled, 95
_SLQSOMADMSettingsReqParams, 95
FOTAUpdate, 96
FOTAdownload, 95
pAutosdm, 96
_SLQSOMADMSettingsReqParams3, 96
FOTAUpdate, 97
FOTAdownload, 96
pAutosdm, 97
pFwAutoCheck, 97
_SLQSSwiGetHostDevInfoParams, 97
bManSize, 98
bModelSize, 98
bPlasmaIDSize, 98
bSWVerSize, 98
pManString, 98
pModelString, 98
pPlasmaIDString, 98
pSWVerString, 98
_SLQSSwiGetOSInfoParams, 98
bNameSize, 98
bVersionSize, 98
pNameString, 98
pVersionString, 99
_SLQSSwiGetSerialNoExtParams, 99
meidLength, 99
pMeidString, 99
_SLQSSwiSetHostDevInfoParams, 99
bManSize, 100
bModelSize, 100
bPlasmaIDSize, 100
bSWVerSize, 100
pManString, 100
pModelString, 100
pPlasmaIDString, 100
pSWVerString, 100
_SLQSSwiSetOSInfoParams, 100
bNameSize, 101
bVersionSize, 101
pNameString, 101
pVersionString, 101
_SlqsNas3GppNetworkRAT_, 89
MCC, 89
MNC, 89
RAT, 89
_SwiAvmsEventInd, 101
BinaryUpdateSessionInfo, 101
Config, 101
ConnectionRequest, 101
DataSessionStatus, 101
HTTPStatus, 102
Notification, 102
PackageID, 102
RegStatus, 102
SessionType, 102
WAMSParaChanged, 102
_getIndicationRegResp, 65
pRegCallStatInfoEvt, 66
pRegTransLayerInfoEvt, 66
pRegTransNWRegInfoEvt, 66
_getResetInfoNotification, 67
source, 68
type, 68
_getTransLayerInfoResp, 68
pRegInd, 68
pTransLayerInfo, 68
_getTransNWRegInfoResp, 68
pRegStatus, 69
_modemTempNotification, 69
ModemTempState, 70
ModemTemperature, 70
_packetSrvStatus, 70
bearerID, 72
connStatus, 72
ipFamily, 72
pQmiInterfaceInfo, 72
reconfigReqd, 72
sessionEndReason, 72
techName, 72
verboseSessnEndReason, 72
verboseSessnEndReasonType, 72

_qaQmi3GPP2BroadcastCfgInfo, 72
activated_ind, 73
CDMABroadcastConfig, 73
num_instances, 73
_qaQmi3GPPBroadcastCfgInfo, 73
activated_ind, 73
broadcastConfig, 73
num_instances, 73
_setIndicationRegReq, 82
pRegCallStatInfoEvt, 83
pRegTransLayerInfoEvt, 83
pRegTransNWRegInfoEvt, 83
_slqs3GPPConfigItem, 83
LTEAttachProfileListLen, 85
p3gppRelease, 85
pDefaultPDNEnabled, 85
pLTEAttachProfile, 85
pLTEAttachProfileList, 85
pProfileList, 85
_slqsNetworkScanInfo, 89
pNetworkInfo, 90
pNetworkInfoInstances, 90
pPCSDigitInfo, 90
pPCSDigitInstances, 90
pRATInfo, 90
pRATInstances, 90
pScanResult, 90
_sysSelectPrefInfo, 102
pBandPref, 106
pEmerMode, 106
pGWAcqOrderPref, 106
pLTEBandPref, 106
pModePref, 106
pNetSelPref, 106
pPRLPref, 106
pRoamPref, 106
pSrvDomainPref, 106
_sysSelectPrefParams, 107
pAcqOrderPref, 112
pBandPref, 112
pCSGID, 112
pChgDuration, 112
pEmerMode, 112
pGWAcqOrderPref, 112
pLTEBandPref, 112
pMNCIncPCSDigStat, 112
pModePref, 112
pNetSelPref, 112
pPRLPref, 112
pRAT, 113
pRoamPref, 113
pSrvDomainPref, 113
pSrvRegRestriction, 113
pTdscdmaBandPref, 113
_transLayerInfoNotification, 113
pTransLayerInfo, 114
regInd, 114
_transLayerinfo, 113
TransCap, 113
TransType, 113
_transNWRegInfoNotification, 114
NWRegStat, 114

AAASPI
unpack_wds_GetMobileIPProfile_t, 1360
AAAState
unpack_wds_GetMobileIPProfile_t, 1360
ABSOLUTE_VALIDITY
qaGobiApiSms.h, 1971
ACT_CODE_MAX_SIZE
dms.h, 1560
ALS
getAllCallInformation, 269
voice_getAllCallInformation, 1430
AMSSString
unpack_dms_GetFirmwareRevision_t, 1115
unpack_dms_GetFirmwareRevisions_t, 1116
APNName
unpack_wds_SLQSGetRuntimeSettings_t, 1377
AVMS (SWIAVMS), 61
absoluteValidity
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 171
accelAcceptReady
qaGobiApiCbk.h, 1727
accelAcceptReady_s, 115
batchPerSec, 115
injectEnable, 115
samplesPerBatch, 115
accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 660
accelTempAcceptReady
qaGobiApiCbk.h, 1727
accelTempAcceptReady_s, 115
batchPerSec, 116
injectEnable, 116
samplesPerBatch, 116
acceleroData
pack_loc_SLQSLOCInjectSensorData_t, 660
acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 660
AccessMac
nas_protocolSubtypeElement, 520
protocolSubtypeElement, 823
accolc
pack_nas_SetACCOLC_t, 666
accuracy
pack_pds_SetPDSDefaults_t, 703
ackIndicator
SMSTransferRouteMTMessage, 986
sMSTransferRouteMTMessageInfo, 986
acqOrdeLen
acqOrderPref, 117
nas_acqOrderPref, 442
acqOrderPref, 116
acqOrdeLen, 117
pAcqOrder, 117

acroamsetting
pack_wds_SetAutoconnect_t, 765
slqsautoconnect, 942
acsetting
pack_wds_SetAutoconnect_t, 765
slqsautoconnect, 942
actCode
pack_dms_ActivateAutomatic_t, 630
ActPilotPNElement, 117
ActSetPilotPN, 117
ActSetPilotPNStrength, 117
ActSetCnt
nas_NetworkStat1x, 505
NetworkStat1x, 614
ActSetPilotPN
ActPilotPNElement, 117
nas_ActPilotPNElement, 442
ActSetPilotPNStrength
ActPilotPNElement, 117
nas_ActPilotPNElement, 442
action
pack_nas_InitiateDomainAttach_t, 665
slqsautoconnect, 942
ssdatasession_params, 991
ActivateAutomatic
qaGobiApiDms.h, 1823
activated_ind
_qaQmi3GPP2BroadcastCfgInfo, 73
_qaQmi3GPPBroadcastCfgInfo, 73
sms_qaQmi3GPP2BroadcastCfgInfo, 965
sms_qaQmi3GPPBroadcastCfgInfo, 965
activationState
pack_dms_SetActivationStatusCallback_t, 632
activationStatus
dms_ActivationStatusTlv, 231
ActivationStatusTlv
unpack_dms_SetEventReport_ind_t, 1129
activeBand
nas_RfBandInfoExtFormatElements, 527
RfBandInfoExtFormatElements, 880
activeBandClass
nas_RFBandInfoElements, 526
nas_RFInfoTlv, 531
RFBandInfoElements, 879
activeChannel
nas_RFBandInfoElements, 526
nas_RfBandInfoExtFormatElements, 527
nas_RFInfoTlv, 531
RFBandInfoElements, 879
RfBandInfoExtFormatElements, 880
activeInd
messageWaitingInfoContent, 436
sms_messageWaitingInfoContent, 963
ActiveStatus
CLIPResp, 185
CLIRResp, 186
CNAPResp, 188
COLPResp, 189
COLRResp, 189
voice_CLIPResp, 1420
voice_CLIRResp, 1420
voice_CNAPResp, 1421
voice_COLPResp, 1422
voice_COLRResp, 1423
ActiveTechPref
unpack_nas_GetNetworkPreference_t, 1190
ActiveTimer
unpack_dms_PSMCfgChange_ind_t, 1127
activeTimer
dms_PSMActiveTimerTlv, 232
PSMActiveTimerTlv, 824
ActiveTimerInd
dms_PSMActiveTimerIndTlv, 232
AddCDMASysInfo, 117
geoSysIdx, 118
regPrd, 118
AddSysInfo, 118
cellBroadcastCap, 118
geoSysIdx, 118
addr
IPv4Addr, 345
IPv6Addr, 347
unpack_qos_IPv4Addr_t, 1255
unpack_qos_IPv6Addr_t, 1256
address
IPv4Info, 346
IPv6Info, 349
loc_IPv4Info, 390
loc_IPv6Info, 391
loc_urlAddr, 398
unpack_wds_GetMobileIPProfile_t, 1360
urlAddr, 1389
addressSize
unpack_sms_GetSMSCAddress_t, 1277
aid
uim_refreshevent, 1061
uim_sessionInformation, 1063
uim_UIMSessionInformation, 1067
UIMRefreshEvent, 1081
UIMSessionInformation, 1084
aidLength
appStats, 126
appStatus, 129
uim_appStatus, 1046
uim_refreshevent, 1061
uim_sessionInformation, 1063
uim_UIMSessionInformation, 1067
UIMRefreshEvent, 1081
UIMSessionInformation, 1084
aidVal
appStats, 126
appStatus, 129
uim_appStatus, 1046
aidingIndicatorMask
loc_sensorDataUsage, 396
sensorDataUsage_s, 903

t_sensor, 1023
airTimer, 118
airTimerValue, 119
namID, 119
airTimerValue
airTimer, 119
voice_airTimer, 1397
alertPitch
signalInfo, 937
voice_signalInfo, 1442
alertingPattern
arrAlertingPattern, 130
voice_arrAlertingPattern, 1400
AlertingType
arrAlertingType, 131
voice_arrAlertingType, 1401
alertmsg
omaDmConfigTlv, 621
omaDmConfigTlvExt, 623
unpack_omaDmConfigTlv_t, 1239
alertmsglength
omaDmConfigTlv, 621
omaDmConfigTlvExt, 623
unpack_omaDmConfigTlv_t, 1239
allCallsAlphaIDInfo, 119
AlphaIDInfo, 119
callID, 119
allCallsAlphaIDInfoArr
arrAlphaID, 131
voice_arrAlphaID, 1402
allCallsDiagInfo, 119
callID, 120
DiagInfo, 120
AllCallsUUSInfo
arrUUSInfo, 137
voice_arrUUSInfo, 1407
allCallsUUSInfo, 120
callID, 120
uusInfo, 120
alphaDcs
alphaIDInfo, 121
voice_alphaIDInfo, 1400
AlphaID
CatAlPhaIdentifierTlv, 160
AlphaIDInfo
allCallsAlphaIDInfo, 119
voice_allCallsAlphaIDInfo, 1398
alphaIDInfo, 120
alphaDcs, 121
alphaLen, 121
alphaText, 121
alphaIDLen
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
AlphaIDLength
CatAlPhaIdentifierTlv, 160
alphaLen
alphaIDInfo, 121
voice_alphaIDInfo, 1400
alphaText
alphaIDInfo, 121
voice_alphaIDInfo, 1400
alphabet
wcdmaMsgEncodingParams, 1502
altSrcInfo_t, 122
coverage, 123
linkage, 123
source, 123
Altitude
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
altitudeAssumed
unpack_loc_GnssSvInfo_Ind_t, 1168
altitudeSrcInfo, 121
coverage, 122
linkage, 122
pack_loc_SLQSLOCInjectPosition_t, 658
source, 122
altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 658
altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 658
ambr_dl
sApnExtraParams, 896
unpack_qos_SLQSQosSwiReadApnExtraParams-_t, 1261
ambr_dl_ext
sApnExtraParams, 896
unpack_qos_SLQSQosSwiReadApnExtraParams-_t, 1261
ambr_dl_ext2
sApnExtraParams, 896
unpack_qos_SLQSQosSwiReadApnExtraParams-_t, 1261
ambr_ul
sApnExtraParams, 896
unpack_qos_SLQSQosSwiReadApnExtraParams-_t, 1261
ambr_ul_ext
sApnExtraParams, 896
unpack_qos_SLQSQosSwiReadApnExtraParams-_t, 1261
ambr_ul_ext2
sApnExtraParams, 896
unpack_qos_SLQSQosSwiReadApnExtraParams-_t, 1261
amssSize
unpack_dms_GetFirmwareRevision_t, 1115
unpack_dms_GetFirmwareRevisions_t, 1116
AnswerUSSD
qaGobiApiVoice.h, 2081
apdoxypages.c, 1547
apnId
pack_qos_SLQSQosSwiReadApnExtraParams_t, 713

pack_qos_SLQSQosSwiReadDataStats_t, 713
sApnExtraParams, 896
sQosStat, 988
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1261
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
apnName
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
apnname
unpack_wds_GetDefaultProfile_t, 1356
apnsize
unpack_wds_GetDefaultProfile_t, 1356
appNameLength
loc_LocApplicationInfo, 392
LocApplicationInfo, 400
appProviderLength
loc_LocApplicationInfo, 392
LocApplicationInfo, 400
appState
appStats, 126
appStatus, 129
uim_appStatus, 1046
appStats, 123
aidLength, 126
aidVal, 126
appState, 126
appType, 126
persoFeature, 126
persoRetries, 126
persoState, 126
persoUnblockRetries, 126
pin1Retries, 126
pin1State, 126
pin2Retries, 126
pin2State, 126
puk1Retries, 126
puk2Retries, 126
univPin, 126
AppStatus
slotInf, 939
slotInfo, 941
uim_slotInfo, 1066
appStatus, 126
aidLength, 129
aidVal, 129
appState, 129
appType, 129
persoFeature, 129
persoRetries, 129
persoState, 129
persoUnblockRetries, 129
pin1Retries, 129
pin1State, 129
pin2Retries, 129
pin2State, 129
puk1Retries, 129
puk2Retries, 129
univPin, 129
appType
appStats, 126
appStatus, 129
uim_appStatus, 1046
appVersionLength
loc_LocApplicationInfo, 392
LocApplicationInfo, 400
appVersionValid
loc_LocApplicationInfo, 392
LocApplicationInfo, 400
Application
unpack_nas_GetCDMANetworkParameters_t, 1186
appversion_str
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
arfcn
GERANInfo, 267
gsmCellInfo, 307
nas_GERANInfo, 468
nas_gsmCellInfo, 469
arrAlertingPattern, 129
alertingPattern, 130
callID, 130
numInstances, 130
arrAlertingType, 130
AlertingType, 131
callID, 131
numInstances, 131
arrAlphaID, 131
allCallsAlphaIDInfoArr, 131
numInstances, 131
arrCallEndReason, 132
callEndReason, 133
callID, 133
numInstances, 133
arrCallInfo, 133
getAllCallInfo, 133
numInstances, 133
arrCallInfomation
unpack_voice_allCallStatusCallback_ind_t, 1311
voiceSetAllCallStatusCbkInfo, 1482
arrCalledPartyNum, 131
CalledPartyNum, 132
numInstances, 132
arrConnectPartyNum, 133
ConnectedPartyNum, 134
numInstances, 134
arrDiagInfo, 134
DiagInfo, 134
numInstances, 134
arrRedirPartyNum, 134
numInstances, 135
RedirPartyNum, 135
arrRemotePartyName, 135
GetAllCallRmtPtyName, 135
numInstances, 135

arrRemotePartyNum, 135
numInstances, 136
RmtPtyNum, 136
arrSvcOption, 136
callID, 136
numInstances, 136
srvOption, 136
arrUUSInfo, 137
AllCallsUUSInfo, 137
numInstances, 137
arrfileInfo
registerRefresh, 875
uim_refreshevent, 1061
uim_registerRefresh, 1062
UIMRefreshEvent, 1081
AtCmdPort
DcsUsbPortNames, 223
Audio Service (AUDIO), 55
auth
unpack_wds_GetDefaultProfile_t, 1356
authData
pack_uim_SLQSUIMAuthenticate_t, 734
UIMAuthenticateReq, 1069
AuthProt
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
authenticateResult, 137
content, 138
contentLen, 138
Authentication
unpack_wds_SLQSGetRuntimeSettings_t, 1377
authentication
pack_wds_SetDefaultProfile_t, 766
authenticationData, 138
context, 138
data, 139
dataLen, 139
Autosdm
unpack_swioma_SLQSOMADMGetSettings_t, 1297
avgPeriod
LTESigRptCfg, 427
LTESigRptConfig, 428
nas_LTESigRptCfg, 494
nas_LTESigRptConfig, 495
azimuth
loc_satelliteInfo, 395
satelliteInfo, 898

bAPNLength
_QmiSwiAvmsSetSettingsAPNInfo, 82
_SLQSAVMSSettingsAPNInfo, 88
bAltitudeAssumed
gnssSvInfoNotification, 300
bAuto
pack_pds_SetPortAutomaticTracking_t, 704
pack_pds_SetServiceAutomaticTracking_t, 704
bBinaryType
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 75
bEnable
pack_dms_UIMSetPINProtection_t, 645
pack_nas_SLQSSetSignalStrengthsCallback_t, 685
bEnabled
pack_pds_SetXTRAAutomaticDownload_t, 705
bForceDownload
pack_fms_SetImagesPreference_t, 649
bICCID
slot_t, 938
UIMSlotStatus, 1086
bICCIDLength
slot_t, 938
UIMSlotStatus, 1086
bLogicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 739
slot_t, 938
UIMSlotStatus, 1086
UIMSwitchSlotReq, 1088
bManSize
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
bModelSize
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
bNameSize
_SLQSSwiGetOSInfoParams, 98
_SLQSSwiSetOSInfoParams, 101
bNotification
_QmiSwiAvmsEventReportNotificationResp, 79
bNumberOfPhySlots
UIMSlotStatusChangeInfo, 1087
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 1301
BOOL
SwiDataTypes.h, 2186
BPTlv
NASQmiCbkNasSystemSelPrefInd, 599
bPWDLength
_QmiSwiAvmsSetSettingsAPNInfo, 82
_SLQSAVMSSettingsAPNInfo, 88
bPackageID
_QmiSwiAvmsEventReportPackageIDResp, 80
bPlasmaIDSize
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
bRegStatus
_QmiSwiAvmsEventReportRegStatusResp, 80
bResetStatistics
rmTrasnferStaticsReq, 884
swiRMTrasnferStaticsReq, 1016
bSWVerSize
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
bSerity

_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
bState
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
_QmiSwiAvmsEventReportConfigResp, 76
bType
_QmiSwiAvmsEventReportDataSessionStatus-Resp, 77
_QmiSwiAvmsEventReportSessionTypeResp, 81
BUILD_ID_LEN
qaGobiApiFms.h, 1857
BUILD_ID_MAX_LEN
qaGobiApiFms.h, 1857
bUnameLength
_QmiSwiAvmsSetSettingsAPNInfo, 82
_SLQSAVMSSettingsAPNInfo, 88
bUserInputRequest
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
_QmiSwiAvmsEventReportConfigResp, 76
_QmiSwiAvmsEventReportConnectionRequest-Resp, 77
bVersionSize
_SLQSSwiGetOSInfoParams, 98
_SLQSSwiSetOSInfoParams, 101
BYT_STAT_STAT_MASK
wds.h, 2256
BYTE
SwiDataTypes.h, 2186
band
LTEInfo, 417
nas_LTEInfo, 485
band1900
gsmCellInfo, 307
nas_gsmCellInfo, 470
band_pref
NASBandPreferenceTlv, 566
BandCapability
unpack_dms_GetBandCapability_t, 1103
bandCapability
BandCapabilityResp, 142
unpack_dms_SLQSGetBandCapability_t, 1139
BandCapabilityResp, 139
bandCapability, 142
pLteBandCapability, 142
pTdsBandCapability, 142
bandPref
pack_nas_SLQSSetBandPreference_t, 684
bandwidth
LTEInfo, 417
nas_LTEInfo, 485
nas_RfBandwidthInfoElements, 529
RfBandwidthInfoElements, 882
baseId
CDMAInfo, 166
CDMASysInfo, 178
nas_CDMAInfo, 447
nas_CDMASysInfo, 451
baseLat
CDMAInfo, 166
CDMASysInfo, 178
nas_CDMAInfo, 447
nas_CDMASysInfo, 451
baseLong
CDMAInfo, 166
CDMASysInfo, 179
nas_CDMAInfo, 447
nas_CDMASysInfo, 451
BasestationID
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
BasestationLatitude
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
BasestationLongitude
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
batchPerSec
accelAcceptReady_s, 115
accelTempAcceptReady_s, 116
gyroAcceptReady_s, 312
gyroTempAcceptReady_s, 313
loc_accelAcceptReady, 381
loc_accelTempAcceptReady, 382
loc_gyroAcceptReady, 388
loc_gyroTempAcceptReady, 389
BdsSV, 142
id, 142
mask, 142
BdsSVInfo, 142
len, 143
pSV, 143
BearerID
unpack_qos_QosFlowInfo_t, 1258
bearerID
_packetSrvStatus, 72
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1381
bearerId
sQosFlowStat, 987
unpack_QosFlowStat_t, 1274
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1387
BinaryUpdateSessionInfo
_SwiAvmsEventInd, 101
bits_129_192
nas_nr5gBandPref, 509
nr5gBandPref, 619
bits_193_256
nas_nr5gBandPref, 509
nr5gBandPref, 619
bits_1_64
nas_nr5gBandPref, 509
nr5gBandPref, 619
bits_65_128

nas_nr5gBandPref, 509
nr5gBandPref, 619
bootSize
unpack_dms_GetFirmwareRevisions_t, 1116
BootString
unpack_dms_GetFirmwareRevisions_t, 1116
bootversion_str
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
Broadcast
unpack_nas_GetCDMANetworkParameters_t, 1186
broadcastActivate
pack_sms_SLQSSetSmsBroadcastActivation_t, 722
BroadcastConfig, 143
fromServiceId, 144
selected, 144
toServiceId, 144
broadcastConfig
_qaQmi3GPPBroadcastCfgInfo, 73
sms_qaQmi3GPPBroadcastCfgInfo, 965
bsInfoValid
CDMASysInfo, 179
nas_CDMASysInfo, 451
bsPRev
CDMASysInfo, 179
nas_CDMASysInfo, 451
bsPRevValid
CDMASysInfo, 179
nas_CDMASysInfo, 451
bsic
GERANInfo, 267
nas_GERANInfo, 468
bsicId
gsmCellInfo, 307
nas_gsmCellInfo, 470
bucketSz
tokenBucket, 1037
unpack_qos_tokenBucket_t, 1273
buildID
CurrImageInfo, 206
FMSImageIdElement, 263
image_info_t, 324
ImageIdElement, 326
buildIDLen
CurrImageInfo, 206
image_info_t, 324
buildIDLength
FMSImageIdElement, 263
ImageIdElement, 326
buildId
FMSImageElement, 262
ImageElement, 325
buildIdLength
FMSImageElement, 262
ImageElement, 325
BurstDTMFInfo
pack_voice_SLQSVoiceBurstDTMF_t, 745
voiceBurstDTMFInfo, 1448
burstDTMFInfo, 144
digitCnt, 144
pCallID, 144
pDigitBuff, 144
ByteLoopbackMode
unpack_wds_SLQSSGetLoopback_t, 1384
WDSGetLoopbackData, 1538
ByteLoopbackMultiplier
unpack_wds_SLQSSGetLoopback_t, 1384
WDSGetLoopbackData, 1538
ByteTotalsElmntsV4
WdsByteTotals, 1529
ByteTotalsElmntsV6
WdsByteTotals, 1529

CATEventDataType, 161
eventMask, 161
pErrorMask, 161
CATSendEnvelopeCommand
qaGobiApiCat.h, 1715
CATSendTerminalResponse
qaGobiApiCat.h, 1716
CBK_DISABLE_EVENT
qaGobiApiCbk.h, 1725
CBK_ENABLE_EVENT
qaGobiApiCbk.h, 1725
CBK_NOCHANGE
qaGobiApiCbk.h, 1725
CCETlv
QmiCbkCatEventStatusReportInd, 833
CDMA_P_Rev
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
CDMABroadcastConfig, 163
_qaQmi3GPP2BroadcastCfgInfo, 73
language, 164
selected, 164
serviceCategory, 164
sms_qaQmi3GPP2BroadcastCfgInfo, 965
CDMAChannel, 164
priChA, 164
priChB, 164
secChA, 164
secChB, 165
CDMAECIOThresh, 165
CDMAECIOThreshListLen, 165
pCDMAECIOThreshList, 165
CDMAECIOThreshListLen
CDMAECIOThresh, 165
nas_CDMAECIOThresh, 446
CDMAInfo, 165
baseId, 166
baseLat, 166
baseLong, 166
nid, 166
refpn, 166
sid, 166

CDMARSSIThresh, 174
CDMARSSIThreshListLen, 174
pCDMARSSIThreshList, 174
CDMARSSIThreshListLen
CDMARSSIThresh, 174
nas_CDMARSSIThresh, 448
CDMASSInfo, 174
ecio, 175
rssi, 175
unpack_nas_SLQSNasGetSigInfo_t, 1225
CDMASysInfo, 176
baseId, 178
baseLat, 178
baseLong, 179
bsInfoValid, 179
bsPRev, 179
bsPRevValid, 179
ccsSupported, 179
ccsSupportedValid, 179
cdmaSysIdValid, 179
isSysPrlMatch, 179
isSysPrlMatchValid, 179
MCC, 179
MNC, 179
networkID, 179
networkIdValid, 179
pRevInUse, 179
pRevInUseValid, 179
packetZone, 179
packetZoneValid, 179
sysInfoCDMA, 179
systemID, 179
CDMASysInfoExt, 179
imsi_11_12, 180
MCC, 180
CDMASystemInfoExt
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
CHAR
SwiDataTypes.h, 2186
CK_MAX_SIZE
dms.h, 1560
CLIPResp, 184
ActiveStatus, 185
ProvisionStatus, 185
CLIRResp, 185
ActiveStatus, 186
ProvisionStatus, 186
CNAPResp, 187
ActiveStatus, 188
ProvisionStatus, 188
COLPResp, 188
ActiveStatus, 189
ProvisionStatus, 189
COLRResp, 189
ActiveStatus, 189
ProvisionStatus, 189
CONFIG_LEN
qaGobiApiSms.h, 1971
CQIValueCW0
LteCQIParm, 414
unpack_nas_SLQSSwiGetLteCQI_t, 1233
CQIValueCW1
LteCQIParm, 414
unpack_nas_SLQSSwiGetLteCQI_t, 1233
CRTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 1368
CSDomain
unpack_nas_GetServingNetwork_t, 1193
CSGID, 199
id, 200
mcc, 200
mnc, 200
mncPcsDigits, 200
rat, 200
CSTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 1368
CUGIndex
CUGInfo, 201
voice_CUGInfo, 1425
CUGInfo, 200
CUGIndex, 201
SuppOA, 201
SuppPrefCUG, 201
CallBacK registration (CBK), 43
CallBarStatus
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
callBarStatus, 145
csBarStatus, 146
psBarStatus, 146
CallBarringSysInfo, 144
csBarStatus, 145
psBarStatus, 145
callDuration
unpack_wds_GetSessionDuration_t, 1365
CallEndReason
DUNCallInfoInd, 248
wds_LastMdmCallEndRsnTlv, 1521
callEndReason
arrCallEndReason, 133
unpack_wds_SLQSGetDUNCallInfo_t, 1374
voice_arrCallEndReason, 1403
CallFWExtInfo
getCallFWExtInfo, 276
voice_getCallFWExtInfo, 1432
callFWExtInfo, 150
noReplyTimer, 152
numLen, 152
numPlan, 152
numType, 152
number, 152
PI, 152
SI, 152

SvcClass, 152
SvcStatus, 152
CallFWInfo
getCallFWInfo, 277
voice_getCallFWInfo, 1432
callFWInfo, 152
noReplyTimer, 153
numLen, 153
number, 153
SvcClass, 153
SvcStatus, 153
callFwdTypeAndPlan, 149
numberPlan, 150
numberType, 150
callID
allCallsAlphaIDInfo, 119
allCallsDiagInfo, 120
allCallsUUSInfo, 120
arrAlertingPattern, 130
arrAlertingType, 131
arrCallEndReason, 133
arrSvcOption, 136
callInfo, 154
DTMFInfo, 247
getAllCallRmtPtyName, 270
getAllCallRmtPtyNum, 270
pack_voice_SLQSVoiceGetCallInfo_t, 749
pack_voice_SLQSVoiceStopContDTMF_t, 760
peerNumberInfo, 793
unpack_voice_OTASPStatusCallback_ind_t, 1313
unpack_voice_SLQSVoiceStopContDTMF_t, 1344
unpack_voice_SUPSNotificationCallback_ind_t, 1348
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
unpack_voice_voicePrivacyChangeCallback_ind_t, 1351
voice_allCallsAlphaIDInfo, 1398
voice_allCallsDiagInfo, 1398
voice_allCallsUUSInfo, 1399
voice_arrAlertingPattern, 1400
voice_arrAlertingType, 1401
voice_arrCallEndReason, 1403
voice_arrSvcOption, 1407
voice_callInfo, 1416
voice_DTMFInfo, 1427
voice_getAllCallRmtPtyName, 1430
voice_getAllCallRmtPtyNum, 1431
voice_peerNumberInfo, 1436
voiceCallInfoReq, 1448
voiceInfoRec, 1476
voiceOTASPStatusInfo, 1479
voicePrivacyInfo, 1480
voiceStopContDTMFinfo, 1492
voiceSUPSNotification, 1496
callInfo, 153
callID, 154
callState, 154
callType, 155
direction, 155
mode, 155
callNumber
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
callState
callInfo, 154
voice_callInfo, 1416
callType
callInfo, 155
voice_callInfo, 1417
calledPartyInfo, 146
numLen, 147
numPlan, 147
numType, 147
number, 147
PI, 148
SI, 148
CalledPartyNum
arrCalledPartyNum, 132
voice_arrCalledPartyNum, 1402
calledPartySubAdd, 148
extBit, 148
oddEvenInd, 148
subAddr, 148
subAddrLen, 148
subAddrType, 148
callerID
callerIDInfo, 149
connectNumInfo, 194
voice_callerIDInfo, 1411
voice_connectNumInfo, 1424
callerIDInfo, 149
callerID, 149
callerIDLen, 149
PI, 149
callerIDLen
callerIDInfo, 149
connectNumInfo, 194
voice_callerIDInfo, 1411
voice_connectNumInfo, 1424
callerName
remotePartyName, 876
voice_remotePartyName, 1440
Callinfo
getAllCallInformation, 269
voice_getAllCallInformation, 1430
callingPartyInfo, 155
numLen, 156
numPlan, 156
numType, 156
number, 156
PI, 156
SI, 156
CancelUSSD
qaGobiApiVoice.h, 2081
Card Application Toolkit (CAT), 46
cardResult, 156

sw1, 157
sw2, 157
cardState
slotInf, 939
slotInfo, 941
uim_slotInfo, 1066
cardStatus, 157
index1xPri, 158
index1xSec, 158
indexGwPri, 158
indexGwSec, 158
numSlot, 158
SlotInfo, 158
Carrier
fwinfo_s, 266
carrier
CurrentImgList, 204
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1144
carrier_str
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
CarrierImage_t, 158
m_FwBuildId, 159
m_FwImageId, 159
m_PriBuildId, 159
m_PriImageId, 159
m_nCarrierId, 159
m_nFolderId, 159
m_nStorage, 159
CatAlPhaIdentifierTlv, 159
AlphaID, 160
AlphaIDLength, 160
ReferenceID, 160
CatAlphaIdtfr
currentCatEvent, 203
CatCommonEventTlv, 160
CatEvent, 160
EventID, 160
EventLength, 160
TlvPresent, 160
CatEndPS
currentCatEvent, 203
CatEndProactiveSessionTlv, 160
EndProactiveSession, 161
CatEvIDData
currentCatEvent, 203
CatEvent
CatCommonEventTlv, 160
CatEventIDDataTlv, 161
Data, 161
DataLength, 161
ReferenceID, 161
CatEventListTlv, 161
SetupEventList, 162
CatEventLst
currentCatEvent, 203
CatRefresh
currentCatEvent, 203
CatRefreshTlv, 162
RefreshMode, 162
RefreshStage, 162
causeCode
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-
_ind_t, 1288
ccSUPSType, 162
reason, 163
svcType, 163
ccsSupported
CDMASysInfo, 179
nas_CDMASysInfo, 451
ccsSupportedValid
CDMASysInfo, 179
nas_CDMASysInfo, 451
cdmaMsgDecodingParams, 166
absoluteValidity, 168
mcTimeStamp, 168
messageLength, 168
pAlertPriority, 168
pCallbkAddr, 168
pCallbkAddrLength, 168
pDisplayMode, 168
pLanguage, 168
pMessage, 169
pMessageID, 169
pPriority, 169
pPrivacy, 169
pReadAcknowledgementReq, 169
pRelativeValidity, 169
pSenderAddr, 169
pSenderAddrLength, 169
pTextMsg, 169
pTextMsgLength, 169
pUserAcknowledgementReq, 169
cdmaMsgDecodingParamsExt, 169
absoluteValidity, 171
mcTimeStamp, 171
messageLength, 171
pAlertPriority, 172
pCallbkAddr, 172
pCallbkAddrLength, 172
pDisplayMode, 172
pIsUDHPresent, 172
pLanguage, 172
pMessage, 172
pMessageID, 172
pPartNum, 172
pPriority, 172
pPrivacy, 172
pReadAcknowledgementReq, 172
pReferenceNum, 172
pRelativeValidity, 172
pSenderAddr, 172
pSenderAddrLength, 172
pTextMsg, 172
pTextMsgLength, 172

pTotalNum, 172
pUserAcknowledgementReq, 172
cdmaMsgEncodingParams, 172
messageId, 174
pCallbackAddr, 174
pDestAddr, 174
pEncodingAlphabet, 174
pMessage, 174
pMessageSize, 174
pPriority, 174
pRelValidity, 174
pTextMsg, 174
textMsgLength, 174
cdmaSSInfo, 175
ecio, 175
rssi, 175
cdmaSysIdValid
CDMASysInfo, 179
nas_CDMASysInfo, 451
cell_resel_priority
infoInterFreq, 344
nas_infoInterFreq, 481
cellBroadcastCap
AddSysInfo, 118
nas_AddSysInfo, 443
CellDb, 180
mask, 180
CellID
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
cellID
GERANInfo, 267
nas_GERANInfo, 468
nas_UMTSInfo, 556
UMTSInfo, 1091
cellId
GSMSysInfo, 311
LTESysInfo, 433
nas_GSMSysInfo, 473
nas_LTESysInfo, 498
nas_WCDMASysInfo, 564
WCDMASysInfo, 1506
cellIdValid
gsmCellInfo, 307
GSMSysInfo, 311
LTESysInfo, 433
nas_gsmCellInfo, 470
nas_GSMSysInfo, 473
nas_LTESysInfo, 499
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
cellInterFreqParams
infoInterFreq, 344
nas_infoInterFreq, 481
cellIsTDD
nas_umtsLTENbrCell, 558
umtsLTENbrCell, 1093
CellParams
LTEInfoIntrafreq, 420
nas_LTEInfoIntrafreq, 488
cellParams, 180
pci, 181
rsrp, 181
rsrq, 181
rssi, 181
srxlev, 181
cellReselPriority
lteGsmCellInfo, 415
LTEInfoIntrafreq, 420
lteWcdmaCellInfo, 434
nas_lteGsmCellInfo, 483
nas_LTEInfoIntrafreq, 488
nas_lteWcdmaCellInfo, 500
cells_len
infoInterFreq, 344
lteGsmCellInfo, 415
nas_infoInterFreq, 481
nas_lteGsmCellInfo, 483
cellsLen
LTEInfoIntrafreq, 420
lteWcdmaCellInfo, 434
nas_LTEInfoIntrafreq, 488
nas_lteWcdmaCellInfo, 500
chaddr
wds_DHCPv4HWConfig, 1515
WdsDHCPv4HWConfig, 1534
wdsDhcpv4HwConfig, 1534
chaddrLen
wds_DHCPv4HWConfig, 1515
WdsDHCPv4HWConfig, 1534
wdsDhcpv4HwConfig, 1534
changePIN
pack_uim_ChangePin_t, 731
UIMChangePinReq, 1070
changeUIMPIN, 181
oldPINLen, 182
oldPINVal, 182
pinID, 182
pinLen, 182
pinVal, 182
ChannelRate, 182
CurrChanRxRate, 183
CurrChanTxRate, 183
DUNCallInfoInd, 248
MaxChanRxRate, 183
MaxChanTxRate, 183
wds_ChannelRateTlv, 1508
channelRate, 183
CurrChanRxRate, 183
CurrChanTxRate, 183
unpack_wds_SLQSGetDUNCallInfo_t, 1374
Chipset
DeviceConfigDetail, 227
nas_DeviceConfigDetail, 463
ciotAcqOrderLen
ciotAcqOrderPref, 184

nas_ciotAcqOrderPref, 454
ciotAcqOrderPref, 183
ciotAcqOrderLen, 184
pCiotAcqOrder, 184
ckLen
depersonalizationInformation, 225
uim_depersonalizationInformation, 1052
ckVal
depersonalizationInformation, 225
uim_depersonalizationInformation, 1052
clear
pack_dms_SLQSSwiGetCrashInfo_t, 640
ClkInfo, 186
mask, 187
codingScheme
nas_PLMNNetworkNameData, 519
PLMNNetworkNameData, 804
remotePartyName, 876
voice_remotePartyName, 1440
CommInfo, 189
imsRegState, 191
modemMode, 191
psState, 191
systemMode, 191
temperature, 191
common.h
eAUDIO, 1550
eCAT, 1550
eCTL, 1550
eDMS, 1550
eIMS, 1550
eIMSA, 1550
eIND, 1551
eLOC, 1550
eLOG_DEBUG, 1550
eLOG_FATAL, 1550
eLOG_INFO, 1550
eLOG_WARN, 1550
eNAS, 1550
ePDS, 1550
eQOS, 1550
eREQ, 1551
eRMS, 1550
eRSP, 1551
eSAR, 1550
eSMS, 1550
eSWIAUDIO, 1550
eSWIDMS, 1550
eSWILOC, 1550
eSWIM2MCMD, 1550
eSWIM2MCMD_AVC2, 1550
eSWIOMA, 1550
eSWIOMAEXT, 1550
eTIMEOUT_10_S, 1551
eTIMEOUT_20_S, 1551
eTIMEOUT_2_S, 1551
eTIMEOUT_300_S, 1551
eTIMEOUT_30_S, 1551
eTIMEOUT_5_S, 1551
eTIMEOUT_60_S, 1551
eTIMEOUT_8_S, 1551
eTIMEOUT_DEFAULT, 1551
eTMD, 1550
eUIM, 1550
eVOICE, 1550
eWDS, 1550
common.h, 1547
eLOG_LEVEL, 1550
eQMI_SVC, 1550
eTimeout, 1550
fill_pack_ctx, 1551
fill_sdu_hdr, 1551
get_version, 1551
glog, 1553
gloglvl, 1553
helper_get_error_code, 1551
helper_get_error_reason, 1551
helper_get_req_str, 1551
helper_get_resp_ctx, 1552
helper_get_xid, 1552
helper_set_log_func, 1552
helper_set_log_lvl, 1552
liteqmi_GetVersion, 1552
liteqmi_helper_decode7bitAsciiEncString, 1552
liteqmi_log, 1552
logger, 1550
MINREQBKLEN, 1549
MSGID_AND_LEN, 1549
MSGID_DONT_CARE, 1549
msgtype, 1551
SDU_HDR_LEN, 1549
UNUSEDPARAM, 1550
unpack_result_code_only, 1552
commonInfo
swiModemStatusResp, 1004
unpack_nas_SLQSNasSwiModemStatus_t, 1229
ConcSvcInfo
unpack_nas_SLQSGetServingSystem_t, 1208
concSvcInfo
qaQmiServingSystemParam, 832
confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 662
Config
_SwiAvmsEventInd, 101
conn_status
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1381
ConnRateElmntsV4
WdsConnectionRate, 1531
ConnRateElmntsV6
WdsConnectionRate, 1531
connStatus
_packetSrvStatus, 72
connectNumInfo, 193
callerID, 194
callerIDLen, 194

numPlan, 194
numPresInd, 194
numType, 194
screeningInd, 194
ConnectedPartyNum
arrConnectPartyNum, 134
voice_arrConnectPartyNum, 1404
ConnectionRequest
_SwiAvmsEventInd, 101
ConnectionStatus, 191
MDMCallDuration, 192
MDMConnStatus, 192
connectionStatus, 192
MDMCallDuration, 192
MDMConnStatus, 193
unpack_wds_GetSessionState_t, 1365
unpack_wds_SLQSGetDUNCallInfo_t, 1374
connetionState
imsaPdpStatusInfo, 329
content
authenticateResult, 138
readResult, 872
uim_authenticateResult, 1047
uim_readResult, 1059
contentLen
authenticateResult, 138
readResult, 872
uim_authenticateResult, 1047
uim_readResult, 1059
context
authenticationData, 138
uim_authenticationData, 1048
contextId
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 785
swiPDPRuntimeSettingsReq, 1005
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
contextType
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 785
swiPDPRuntimeSettingsReq, 1005
ControlMac
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
count0
nas_RankIndicatorTlv, 525
Count1
RankIndicatorInd, 871
count1
nas_RankIndicatorTlv, 525
Count2
RankIndicatorInd, 871
countryInitials
nas_PLMNNetworkNameData, 519
PLMNNetworkNameData, 804
coverage
altitudeSrcInfo, 122
altSrcInfo_t, 123
cpich_ecno
nas_wcdmaCellInfo, 560
wcdmaCellInfo, 1496
cpich_rscp
nas_wcdmaCellInfo, 560
wcdmaCellInfo, 1496
cradleMountConfigStatus
QmiCbkLocCradleMountInd, 839
unpack_loc_CradleMountCallback_Ind_t, 1163
crashAction
pack_dms_SetCrashAction_t, 632
crashData
CrashInfo, 195
crashInformation, 197
crashId
CrashInfo, 195
crashInformation, 197
CrashInfo, 194
crashData, 195
crashId, 195
crashStrLen, 195
gcDumpStrLen, 195
numCrashes, 195
pCrashString, 195
pGCDumpString, 195
crashInfo
crashInfoParams, 196
crashInfoParam
unpack_dms_SLQSSwiGetCrashInfo_t, 1142
CrashInfoParams, 195
pCrashInfo, 196
pDevCrashStatus, 196
crashInfoParams, 196
crashInfo, 196
crashStatus, 196
crashInformation, 196
crashData, 197
crashId, 197
crashString, 197
crashStrlen, 197
gcdumpString, 197
gcdumpStrlen, 197
numCrashes, 197
crashStatus
crashInfoParams, 196
crashStrLen
CrashInfo, 195
crashString
crashInformation, 197
crashStrlen
crashInformation, 197
CreateProfileIn, 197
curProfile, 198
pProfileID, 198
pProfileType, 198
CreateProfileOut, 198
pExtErrorCode, 199

pProfileIndex, 199
pProfileType, 199
csAttachState
nas_servSystem, 541
NASServingSystemInfo, 601
ServingSystemInfo, 906
servSystem, 908
csBarStatus
CallBarringSysInfo, 145
callBarStatus, 146
nas_CallBarringSysInfo, 444
nas_callBarStatus, 445
CsgId, 199
csgId, 199
TlvPresent, 199
csgId
CsgId, 199
cur_carr_name
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
cur_carr_rev
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
curAMRConfig, 201
gsmAmrStat, 202
wcdmaAmrStat, 202
curDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 1372
curProfile
_GetProfileSettingOut, 67
CreateProfileIn, 198
ModifyProfileIn, 439
pack_wds_SLQSModifyProfile_t, 777
UnPackGetProfileSettingOut, 1388
CurrChanRxRate
ChannelRate, 183
channelRate, 183
dunchannelRate, 249
wds_channelRate, 1507
CurrChanTxRate
ChannelRate, 183
channelRate, 183
dunchannelRate, 249
wds_channelRate, 1507
currDBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
CurrDataSysStat, 202
pCurrNetworkInfo, 203
pNetworkInfoLen, 203
pPrefNetwork, 203
CurrImageInfo, 205
buildID, 206
buildIDLen, 206
imageType, 206
uniqueID, 206
currNWInfo
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
CurrNetworkInfo, 206
NetworkType, 207
RATMask, 208
SOMask, 208
currNetworkInfo, 208
NetworkType, 209
RATMask, 209
SOMask, 209
unpack_wds_SLQSGetCurrDataSystemStat_t, 1370
current_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1371
WDSSWICurrentChannelRates, 1546
current_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1371
WDSSWICurrentChannelRates, 1546
currentCatEvent, 203
CatAlphaIdtfr, 203
CatEndPS, 203
CatEvIDData, 203
CatEventLst, 203
CatRefresh, 203
currentChannelRXRate
unpack_wds_GetConnectionRate_t, 1353
currentChannelTXRate
unpack_wds_GetConnectionRate_t, 1353
currentDataBearer
pack_wds_SLQSSetWdsEventCallback_t, 780
CurrentImgList, 203
carrier, 204
fwvers, 204
numEntries, 204
pCurrImgInfo, 204
pkgver, 204
priver, 204
currentMitigationLvl
QmiCbkTmdMitiLvlRptInd, 854
currentNetwork
dataBearerTechnology, 219
qmiWDSDataBearerTechnology, 860
CurrentPLMN
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
currentPLMN, 204
MCC, 205
MNC, 205
netDescr, 205
netDescrLength, 205
cust_attr
custSettingInfo, 214
DMScustSettingInfo, 239
cust_id
custSettingInfo, 214
DMScustSettingInfo, 239

DMSgetCustomInput, 241
getCustomInput, 278
pack_dms_GetCustFeaturesV2_t, 631
pack_dms_SetCustFeaturesV2_t, 635
setCustomSettingV2, 915
cust_value
custSettingInfo, 214
DMScustSettingInfo, 239
pack_dms_SetCustFeaturesV2_t, 635
setCustomSettingV2, 915
custFeaturesInfo, 209
GpsEnable, 211
pDHCPRelayEnabled, 211
pDisableIMSI, 211
pGPSLPM, 211
pGPSSel, 211
pIPFamSupport, 211
pIsVoiceEnabled, 211
pRMAutoConnect, 211
pSMSSupport, 211
qaGobiApiDms.h, 1816
custFeaturesSetting, 212
pDHCPRelayEnabled, 213
pGPSEnable, 213
pGPSLPM, 213
pGPSSel, 213
pIsVoiceEnabled, 213
qaGobiApiDms.h, 1818
custSetting
custSettingList, 214
DMScustSettingList, 240
custSettingInfo, 213
cust_attr, 214
cust_id, 214
cust_value, 214
id_length, 214
value_length, 214
custSettingList, 214
custSetting, 214
list_type, 214
num_instances, 215
CustomSCP
unpack_nas_GetCDMANetworkParameters_t, 1186
CwtMute
GetM2MAudioProfileResp, 291
GetM2MAVMuteResp, 293

DEVICE_STATE_BOOT
qaGobiApiCbk.h, 1766
DEVICE_STATE_DISCONNECTED
qaGobiApiCbk.h, 1766
DEVICE_STATE_READY
qaGobiApiCbk.h, 1766
DEVICE_STATE_SERIAL_ONLY
qaGobiApiCbk.h, 1766
DBTTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 1368
dBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
dBTechExtRatValue
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
dBTechExtSoMask
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
dBTechnology
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
dBtechExtAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
dBtechnologyExt
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
DEFAULTBYTEVALUE
qaGobiApiPds.h, 1945
DEFAULTLONGVALUE
qaGobiApiPds.h, 1945
DEFAULTWORDVALUE
qaGobiApiPds.h, 1945
DEREGISTER_EVENT
qaGobiApiCbk.h, 1725
DEREGISTER_SRV
qaGobiApiCbk.h, 1725
DEVICE_OFFLINE
qaGobiApiFms.h, 1857
DEVICE_RESET
qaGobiApiFms.h, 1857
DEVICE_SHUTDOWN
qaGobiApiFms.h, 1857
DHCPOption, 228
optCode, 228
optValLen, 228
pOptVal, 228
DHCPOptionList, 228
numOpt, 229
pOptions, 229
DHCPRelayEnabled
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1106
DHCPv4LeaseOptTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1352
DHCPv4LeaseStateTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1352
dLatitude
unpack_pds_SetEventReport_Ind_t, 1248
dLongitude
unpack_pds_SetEventReport_Ind_t, 1248
DMS_IMGDETAILS_LEN
dms.h, 1560
DMS_PM_FACTORY
dms.h, 1560
DMS_PM_LOW
dms.h, 1560

DMS_PM_OFFLINE
dms.h, 1560
DMS_PM_ONLINE
dms.h, 1560
DMS_PM_RESET
dms.h, 1560
DMS_PM_SHUT_DOWN
dms.h, 1560
DMS_VALID_FSN_LEN
dms.h, 1561
DMScustSettingInfo, 238
cust_attr, 239
cust_id, 239
cust_value, 239
id_length, 239
value_length, 239
DMScustSettingList, 239
custSetting, 240
list_type, 240
num_instances, 240
DMSgetCustomFeatureV2, 240
pCustSettingInfo, 240
pCustSettingList, 240
pGetCustomInput, 240
DMSgetCustomInput, 241
cust_id, 241
list_type, 241
DRCCover
DRCParams, 246
nas_DRCParams, 464
DRCParams, 245
DRCCover, 246
DRCValue, 246
DRCValue
DRCParams, 246
nas_DRCParams, 464
DSTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 1368
DTMFEvent
DTMFInfo, 247
voice_DTMFInfo, 1427
DTMFInfo, 246
callID, 247
DTMFEvent, 247
digitBuff, 247
digitCnt, 247
DTMFInformation
unpack_voice_DTMFEventCallback_ind_t, 1312
voiceDTMFEventInfo, 1455
DTMFInterdigitInterval
DTMFLengths, 247
voice_DTMFLengths, 1428
DTMFLengths, 247
DTMFInterdigitInterval, 247
DTMFPulseWidth, 248
DTMFPulseWidth
DTMFLengths, 248
voice_DTMFLengths, 1428
DTMFdigit
pack_voice_SLQSVoiceStartContDTMF_t, 760
voiceContDTMFinfo, 1454
DTMInd
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
DUNCallInfoInd, 248
CallEndReason, 248
ChannelRate, 248
DataBearerTech, 248
DormancyStatus, 248
MdmConnStatus, 248
RXOKBytesCount, 248
TXOKBytesCount, 248
Data
CatEventIDDataTlv, 161
data
authenticationData, 139
SMSCAddress, 973
sMSCAddressInfo, 973
SMSEtwsMessage, 974
sMSEtwsMessageInfo, 975
SMSTransferRouteMTMessage, 986
sMSTransferRouteMTMessageInfo, 986
SwiOTAMsg_s, 1004
uim_authenticationData, 1048
data_buf
NASOTAMessageTlv, 590
data_len
NASOTAMessageTlv, 590
SwiOTAMsg_s, 1004
dataBearer
pack_wds_SLQSSetWdsEventCallback_t, 780
dataBearerMask
dataBearers, 215
unpack_wds_SLQSGetDataBearerTechnology_t, 1372
DataBearerTech, 215
DUNCallInfoInd, 248
ratValue, 217
soMask, 217
techType, 217
wds_DataBearTechTlv, 1511
dataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1374
DataBearerTechExt, 217
pBearerTech, 218
pLastBearerTech, 218
dataBearerTechExt
pack_wds_SLQSSetWdsEventCallback_t, 780
dataBearerTechnology, 218
currentNetwork, 219
ratMask, 219
soMask, 219
dataBearers, 215
dataBearerMask, 215
pCurDataBearerTechnology, 215

pLastCallDataBearerTechnology, 215
dataCapabilities
dataSrvCapabilities, 220
nas_dataSrvCapabilities, 458
dataCapabilitiesLen
dataSrvCapabilities, 220
nas_dataSrvCapabilities, 458
DataCaps
unpack_nas_GetServingNetwork_t, 1193
unpack_nas_GetServingNetworkCapabilities_t, 1194
dataCaps
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1196
DataCapsLen
unpack_nas_GetServingNetwork_t, 1193
unpack_nas_GetServingNetworkCapabilities_t, 1194
dataCapsSize
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1196
dataLen
authenticationData, 139
uim_authenticationData, 1048
DataLength
CatEventIDDataTlv, 161
DataRate
unpack_qos_swiQosFlow_t, 1271
dataRate, 219
dataRateMax, 220
guaranteedRate, 220
dataRateMax
dataRate, 220
unpack_qos_dataRate_t, 1255
dataServiceCaCapability
unpack_dms_GetDeviceCapabilities_t, 1109
DataServiceCapability
unpack_dms_GetDeviceCap_t, 1108
DataSessionStatus
_SwiAvmsEventInd, 101
DataSrvCapabilities
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
dataSrvCapabilities, 220
dataCapabilities, 220
dataCapabilitiesLen, 220
DataStatusDetail, 221
IPAddress, 222
LastErrCode, 222
dataSysStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
dataSystemStatus
pack_wds_SLQSSetWdsEventCallback_t, 780
DataULongLongTlv, 223
TlvPresent, 223
ullData, 223
DataULongTlv, 223
TlvPresent, 223
ulData, 223
Date
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
wcdmaLongMsgDecodingParams, 1499
wcdmaMsgDecodingParams, 1501
DateLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
day
nas_timeInfo, 553
nas_UniversalTime, 559
timeInfo, 1033
UniversalTime, 1101
dayLtSavingAdj
nas_timeInfo, 553
timeInfo, 1033
dayOfWeek
nas_timeInfo, 553
nas_UniversalTime, 559
timeInfo, 1033
UniversalTime, 1101
daylightSavings
nas_qaQmi3Gpp2TimeZone, 521
qaQmi3Gpp2TimeZone, 828
DcsUsbPortNames, 223
AtCmdPort, 223
DmPort, 223
NmeaPort, 223
dedicatedBand
nas_RfDedicatedBandInfoElements, 530
RfDedicatedBandInfoElements, 884
defaultPDNEnabled
unpack_wds_SLQSGet3GPPConfigItem_t, 1369
DefaultRoamInd
unpack_nas_SLQSGetServingSystem_t, 1208
defaultRoamInd
qaQmiServingSystemParam, 832
delAssistDataStatus, 223
status, 224
delayClass
GPRSQoS, 301
GPRSRequestedQoS, 302
LibPackGPRSRequestedQoS, 351
wds_GPRSQoS, 1519
DeleteStoredImage
qaGobiApiFms.h, 1860
deliveryErrSDU
LibPackUMTSQoS, 379
UMTSMinQoS, 1095
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
depersonalizationInformation, 224
ckLen, 225
ckVal, 225
feature, 225
operation, 225

depersonilisationInfo
pack_uim_SLQSUIMDepersonalization_t, 734
UIMDepersonalizationReq, 1071
Description
SlqsNas3GppNetworkInfo, 945
description
omaDmFotaTlv, 624
omaDmFotaTlvExt, 627
unpack_omaDmFotaTlv_t, 1241
descriptionlength
omaDmFotaTlv, 624
omaDmFotaTlvExt, 627
unpack_omaDmFotaTlv_t, 1241
Desription
nas_QmiNas3GppNetworkInfo, 523
destPortRangeEnd
LibPackTFTIDParams, 376
TFTIDParams, 1031
destPortRangeStart
LibPackTFTIDParams, 376
TFTIDParams, 1031
detailSvcInfo, 225
hdrHybrid, 226
hdrSrvStatus, 226
isSysForbidden, 226
srvCapability, 226
srvStatus, 227
DetailedSvcInfo
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
dev
qmifwinfo_s, 856
DevCrashState
unpack_dms_GetCrashAction_t, 1104
Device
SetM2MAudioAVCFGReq, 921
Device Connectivity Service (DCS), 39
Device Management Service (DMS), 41
device_state_enum
qaGobiApiCbk.h, 1766
DeviceConfigDetail, 227
Chipset, 227
HWVersion, 227
QLIC, 228
Technology, 228
deviceID
_MitigationDevInfo, 69
deviceIdLen
_MitigationDevInfo, 69
DiagInfo
allCallsDiagInfo, 120
arrDiagInfo, 134
voice_allCallsDiagInfo, 1398
voice_arrDiagInfo, 1404
diagInfo, 229
diagInfoLen, 229
diagnosticInfo, 229
diagInfoLen
diagInfo, 229
voice_diagInfo, 1426
diagnosticInfo
diagInfo, 229
voice_diagInfo, 1426
digitBuff
DTMFInfo, 247
voice_DTMFInfo, 1427
digitCnt
burstDTMFInfo, 144
DTMFInfo, 247
voice_burstDTMFInfo, 1408
voice_DTMFInfo, 1427
dirNum, 229
dirNum, 230
dirNumLen, 230
dirNum, 230
nas_dirNum, 463
dirNumLen
dirNum, 230
nas_dirNum, 463
direction
callInfo, 155
voice_callInfo, 1417
DisableIMSI
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1106
dispType
extDispRecInfo, 255
voice_extDispRecInfo, 1429
displayCondition
nas_serviceProviderName, 539
serviceProviderName, 905
dl_bw_value
nas_PhyCaAggPcellInfo, 511
nas_PhyCaAggScellDlBw, 512
nas_PhyCaAggScellInfo, 516
NASPhyCaAggPcellInfo, 590
NASPhyCaAggScellDlBw, 591
NASPhyCaAggScellInfo, 593
PhyCaAggPcellInfo, 795
PhyCaAggScellDlBw, 796
PhyCaAggScellInfo, 800
DmPort
DcsUsbPortNames, 223
dms.h, 1553
ACT_CODE_MAX_SIZE, 1560
CK_MAX_SIZE, 1560
DMS_IMGDETAILS_LEN, 1560
DMS_PM_FACTORY, 1560
DMS_PM_LOW, 1560
DMS_PM_OFFLINE, 1560
DMS_PM_ONLINE, 1560
DMS_PM_RESET, 1560
DMS_PM_SHUT_DOWN, 1560
DMS_VALID_FSN_LEN, 1561
ERI_DATA_MAX_SIZE, 1561
MAX_BUILD_ID_LEN, 1561

MEID_MAX_SIZE, 1561
pack_dms_ActivateAutomatic, 1561
pack_dms_GetActivationState, 1562
pack_dms_GetBandCapability, 1562
pack_dms_GetCrashAction, 1562
pack_dms_GetCustFeature, 1563
pack_dms_GetCustFeaturesV2, 1563
pack_dms_GetDeviceCap, 1564
pack_dms_GetDeviceCapabilities, 1564
pack_dms_GetDeviceHardwareRev, 1564
pack_dms_GetDeviceMfr, 1565
pack_dms_GetDeviceSerialNumbers, 1565
pack_dms_GetFSN, 1566
pack_dms_GetFirmwareInfo, 1565
pack_dms_GetFirmwareRevision, 1566
pack_dms_GetFirmwareRevisions, 1566
pack_dms_GetHardwareRevision, 1567
pack_dms_GetIMSI, 1567
pack_dms_GetManufacturer, 1567
pack_dms_GetModelID, 1568
pack_dms_GetNetworkTime, 1568
pack_dms_GetOfflineReason, 1568
pack_dms_GetPRLVersion, 1569
pack_dms_GetPower, 1569
pack_dms_GetSerialNumbers, 1569
pack_dms_GetUSBComp, 1570
pack_dms_GetVoiceNumber, 1570
pack_dms_ResetToFactoryDefaults, 1570
pack_dms_SLQSDmsSwiGetPCInfo, 1574
pack_dms_SLQSDmsSwiGetResetInfo, 1575
pack_dms_SLQSDmsSwiIndicationRegister, 1575
pack_dms_SLQSGetBandCapability, 1575
pack_dms_SLQSGetERIFile, 1576
pack_dms_SLQSGetPowerSaveModeConfig, 1576
pack_dms_SLQSSetPowerSaveModeConfig, 1576
pack_dms_SLQSSwiClearDyingGaspStatistics, 1577
pack_dms_SLQSSwiGetCrashInfo, 1577
pack_dms_SLQSSwiGetDyingGaspCfg, 1578
pack_dms_SLQSSwiGetDyingGaspStatistics, 1578
pack_dms_SLQSSwiGetFirmwareCurr, 1578
pack_dms_SLQSSwiGetFwUpdateStatus, 1579
pack_dms_SLQSSwiGetHostDevInfo, 1579
pack_dms_SLQSSwiGetOSInfo, 1579
pack_dms_SLQSSwiGetSerialNoExt, 1580
pack_dms_SLQSSwiSetDyingGaspCfg, 1580
pack_dms_SLQSSwiSetHostDevInfo, 1580
pack_dms_SLQSSwiSetOSInfo, 1581
pack_dms_SLQSUIMGetState, 1581
pack_dms_SetActivationStatusCallback, 1571
pack_dms_SetCrashAction, 1571
pack_dms_SetCustFeature, 1572
pack_dms_SetCustFeaturesV2, 1572
pack_dms_SetEventReport, 1572
pack_dms_SetFirmwarePreference, 1573
pack_dms_SetIndicationRegister, 1573
pack_dms_SetPower, 1573
pack_dms_SetUSBComp, 1574
pack_dms_SwiSetEventReport, 1581
pack_dms_UIMChangePIN, 1582
pack_dms_UIMGetControlKeyStatus, 1582
pack_dms_UIMGetICCID, 1582
pack_dms_UIMGetPINStatus, 1583
pack_dms_UIMSetControlKeyProtection, 1583
pack_dms_UIMSetPINProtection, 1584
pack_dms_UIMUnblockControlKey, 1584
pack_dms_UIMUnblockPIN, 1584
pack_dms_UIMVerifyPIN, 1585
pack_dms_ValidateSPC, 1585
SPC_SIZE, 1561
UNIQUE_ID_LEN, 1561
unpack_dms_ActivateAutomatic, 1585
unpack_dms_GetActivationState, 1586
unpack_dms_GetBandCapability, 1586
unpack_dms_GetCrashAction, 1587
unpack_dms_GetCustFeature, 1587
unpack_dms_GetCustFeaturesV2, 1587
unpack_dms_GetDeviceCap, 1588
unpack_dms_GetDeviceCapabilities, 1588
unpack_dms_GetDeviceHardwareRev, 1588
unpack_dms_GetDeviceMfr, 1589
unpack_dms_GetDeviceSerialNumbers, 1589
unpack_dms_GetFSN, 1590
unpack_dms_GetFirmwareInfo, 1589
unpack_dms_GetFirmwareRevision, 1590
unpack_dms_GetFirmwareRevisions, 1590
unpack_dms_GetHardwareRevision, 1591
unpack_dms_GetIMSI, 1591
unpack_dms_GetManufacturer, 1591
unpack_dms_GetModelID, 1592
unpack_dms_GetNetworkTime, 1592
unpack_dms_GetOfflineReason, 1592
unpack_dms_GetPRLVersion, 1593
unpack_dms_GetPower, 1593
unpack_dms_GetSerialNumbers, 1593
unpack_dms_GetUSBComp, 1594
unpack_dms_GetVoiceNumber, 1594
unpack_dms_PSMCfgChange_ind, 1595
unpack_dms_ResetToFactoryDefaults, 1595
unpack_dms_SLQSDmsSwiGetPCInfo, 1599
unpack_dms_SLQSDmsSwiGetResetInfo, 1599
unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 1600
unpack_dms_SLQSDmsSwiIndicationRegister, 1600
unpack_dms_SLQSGetBandCapability, 1600
unpack_dms_SLQSGetERIFile, 1601
unpack_dms_SLQSGetPowerSaveModeConfig, 1601
unpack_dms_SLQSSetPowerSaveModeConfig, 1601
unpack_dms_SLQSSwiClearDyingGaspStatistics, 1602
unpack_dms_SLQSSwiGetCrashInfo, 1602
unpack_dms_SLQSSwiGetDyingGaspCfg, 1603

unpack_dms_SLQSSwiGetDyingGaspStatistics, 1603
unpack_dms_SLQSSwiGetFirmwareCurr, 1603
unpack_dms_SLQSSwiGetFwUpdateStatus, 1604
unpack_dms_SLQSSwiGetHostDevInfo, 1604
unpack_dms_SLQSSwiGetOSInfo, 1604
unpack_dms_SLQSSwiGetSerialNoExt, 1605
unpack_dms_SLQSSwiSetDyingGaspCfg, 1605
unpack_dms_SLQSSwiSetHostDevInfo, 1605
unpack_dms_SLQSSwiSetOSInfo, 1606
unpack_dms_SLQSUIMGetState, 1606
unpack_dms_SetActivationStatusCallback, 1595
unpack_dms_SetCrashAction, 1596
unpack_dms_SetCustFeature, 1596
unpack_dms_SetCustFeaturesV2, 1596
unpack_dms_SetEventReport, 1597
unpack_dms_SetEventReport_ind, 1597
unpack_dms_SetFirmwarePreference, 1597
unpack_dms_SetIndicationRegister, 1598
unpack_dms_SetPower, 1598
unpack_dms_SetUSBComp, 1598
unpack_dms_SwiEventReportCallBack_ind, 1606
unpack_dms_SwiSetEventReport, 1607
unpack_dms_UIMChangePIN, 1607
unpack_dms_UIMGetControlKeyStatus, 1607
unpack_dms_UIMGetICCID, 1608
unpack_dms_UIMGetPINStatus, 1608
unpack_dms_UIMSetControlKeyProtection, 1608
unpack_dms_UIMSetPINProtection, 1609
unpack_dms_UIMUnblockControlKey, 1609
unpack_dms_UIMUnblockPIN, 1610
unpack_dms_UIMVerifyPIN, 1610
unpack_dms_ValidateSPC, 1610
dms_ActivationStatusTlv, 230
activationStatus, 231
TlvPresent, 231
dms_OperatingModeTlv, 231
operatingMode, 231
TlvPresent, 231
dms_PSMActiveTimerIndTlv, 231
ActiveTimerInd, 232
TlvPresent, 232
dms_PSMActiveTimerTlv, 232
activeTimer, 232
TlvPresent, 232
dms_PSMDurationDueToOOSTlv, 232
durationDueToOOS, 233
TlvPresent, 233
dms_PSMDurationThresholdTlv, 233
durationThreshold, 233
TlvPresent, 233
dms_PSMEarlyWakeupTimeTlv, 233
earlyWakeupTime, 234
TlvPresent, 234
dms_PSMEnableStateIndTlv, 234
EnableStateInd, 234
TlvPresent, 234
dms_PSMEnableStateTlv, 234
enableState, 235
TlvPresent, 235
dms_PSMPeriodicUpdateTimerIndTlv, 235
PeriodicUpdateTimerInd, 235
TlvPresent, 235
dms_PSMPeriodicUpdateTimerTlv, 235
periodicUpdateTimer, 236
TlvPresent, 236
dms_PSMRandomizationWindowTlv, 236
randomizationWindow, 236
TlvPresent, 236
dms_TemperatureTlv, 236
TempStat, 237
Temperature, 237
TlvPresent, 237
dms_VoltageTlv, 237
TlvPresent, 238
VoltStat, 238
Voltage, 238
dmsCurrentPRLInfo, 238
pPRLPreference, 238
pPRLVersion, 238
qaGobiApiDms.h, 1819
dmsIndicationRegisterReq, 241
pSwiGetResetInd, 241
dmsPsmCfgParamChange, 242
pActiveTimer, 242
pEnableState, 242
pPeriodicUpdateTimer, 242
dmsSwiGetPCInfo, 242
opMode, 243
pLpmFlag, 244
pPersistentLpm, 244
pPowerOffMode, 244
pWdisable, 244
dmsSwiGetResetInfo, 244
source, 244
type, 244
Domain, 244
domainLen, 245
domainName, 245
domain
DomainNameList, 245
wds_DomainNameList, 1518
domainLen
Domain, 245
wds_Domain, 1517
DomainList
unpack_wds_SLQSGetRuntimeSettings_t, 1377
domainName
Domain, 245
wds_Domain, 1517
DomainNameList, 245
domain, 245
numInstances, 245
DormancyStat
wds_DormStatTlv, 1518
dormancyStatAvail

unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
dormancyState
unpack_wds_GetDormancyState_t, 1358
DormancyStatus
DUNCallInfoInd, 248
dormancyStatus
pack_wds_SLQSSetWdsEventCallback_t, 780
unpack_wds_SLQSGetDUNCallInfo_t, 1375
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
downLink
NSSAudioCtrl, 620
voice_NSSAudioCtrl, 1434
DownloadToSlot
qaGobiApiFms.h, 1860
dscp
QosMap, 871
dtmSupp
GSMSysInfo, 311
nas_GSMSysInfo, 473
dtmSuppValid
GSMSysInfo, 311
nas_GSMSysInfo, 473
dunchannelRate, 248
CurrChanRxRate, 249
CurrChanTxRate, 249
MaxChanRxRate, 249
MaxChanTxRate, 249
Duration
pack_nas_SetNetworkPreference_t, 668
unpack_nas_GetNetworkPreference_t, 1190
durationDueToOOS
dms_PSMDurationDueToOOSTlv, 233
PSMDurationDueToOOSTlv, 825
durationThreshold
dms_PSMDurationThresholdTlv, 233
PSMDurationThresholdTlv, 825

eAUDIO
common.h, 1550
eCAT
common.h, 1550
eCTL
common.h, 1550
eDMS
common.h, 1550
eGOBI_DEV_SERIES_9X07_9X50
qaGobiApiFms.h, 1858
eGOBI_DEV_SERIES_9X15
qaGobiApiFms.h, 1858
eGOBI_DEV_SERIES_9X30
qaGobiApiFms.h, 1858
eGOBI_DEV_SERIES_G3K
qaGobiApiFms.h, 1858
eGOBI_DEV_SERIES_NON_GOBI
qaGobiApiFms.h, 1858
eGOBI_DEV_SERIES_SIERRA_GOBI
qaGobiApiFms.h, 1858
eGOBI_DEV_SERIES_UNKNOWN
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_3
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_AERIS
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_ALLTEL
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_AMX_TELCEL
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_ATT
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_BELL
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_BHARTI
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_BRASIL_VIVO
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_CHINA_MOBILE
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_CHINA_TELECOM
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_CHINA_UNICOM
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_EMOBILE
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_FACTORY
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_GENERIC
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_GENERIC_CDMA
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_IUSACELL
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_KDDI
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_KT_FREETEL
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_LEAP
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_METROPCS
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_NETCOM
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_NORF
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_NTT_DOCOMO
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_O2
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_OMH
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_ORANGE
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_RELIANCE1
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_RELIANCE2
qaGobiApiFms.h, 1859

eGOBI_IMG_CAR_ROGERS
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_SFR
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_SINGTEL_OPTUS
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_SK_TELCOM1
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_SK_TELCOM2
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_SOFTBANK
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_SPRINT
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_SWISSCOM
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_TATA
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_TELCOM_ITALIA
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_TELCOM_NZ
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_TELEFONICA
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_TELENOR
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_TELIASONERA
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_TELSTRA1
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_TELSTRA2
qaGobiApiFms.h, 1859
eGOBI_IMG_CAR_TELUS
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_TMOBILE
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_US
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_VERIZON
qaGobiApiFms.h, 1858
eGOBI_IMG_CAR_VODAFONE
qaGobiApiFms.h, 1858
eGOBI_IMG_GPS_ASSISTED
qaGobiApiFms.h, 1859
eGOBI_IMG_GPS_NO_XTRA
qaGobiApiFms.h, 1859
eGOBI_IMG_GPS_NONE
qaGobiApiFms.h, 1859
eGOBI_IMG_GPS_STAND_ALONE
qaGobiApiFms.h, 1859
eGOBI_IMG_REG_ASIA
qaGobiApiFms.h, 1859
eGOBI_IMG_REG_AUS
qaGobiApiFms.h, 1859
eGOBI_IMG_REG_EU
qaGobiApiFms.h, 1859
eGOBI_IMG_REG_GLOBAL
qaGobiApiFms.h, 1859
eGOBI_IMG_REG_LA
qaGobiApiFms.h, 1859
eGOBI_IMG_REG_NA
qaGobiApiFms.h, 1859
eGOBI_IMG_TECH_CDMA
qaGobiApiFms.h, 1860
eGOBI_IMG_TECH_UMTS
qaGobiApiFms.h, 1860
eGobi_DEV_SERIES_MC83
qaGobiApiFms.h, 1858
eIMS
common.h, 1550
eIMSA
common.h, 1550
eIND
common.h, 1551
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
nas.h, 1644
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
nas.h, 1644
eLOC
common.h, 1550
eLOG_DEBUG
common.h, 1550
eLOG_FATAL
common.h, 1550
eLOG_INFO
common.h, 1550
eLOG_WARN
common.h, 1550
eNAS
common.h, 1550
eNAS_LTE_CPHY_CA_BW_NRB_100
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_CA_BW_NRB_15
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_CA_BW_NRB_25
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_CA_BW_NRB_50
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_CA_BW_NRB_6

qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_CA_BW_NRB_75
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
nas.h, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
nas.h, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
nas.h, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
nas.h, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
nas.h, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
nas.h, 1644
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE
nas.h, 1645
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
nas.h, 1645
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
qaGobiApiNas.h, 1909
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
nas.h, 1645
eNAS_RADIO_IF_GSM
qaGobiApiNas.h, 1908
eNAS_RADIO_IF_LTE
qaGobiApiNas.h, 1908
eNAS_RADIO_IF_TDSCDMA
qaGobiApiNas.h, 1908
eNAS_RADIO_IF_UMTS
qaGobiApiNas.h, 1908
ePDS
common.h, 1550
eQA_QMI_SVC_NA
qaGobiApiCbk.h, 1766
eQA_QMI_SVC_NAS
qaGobiApiCbk.h, 1766
eQA_QMI_SVC_WDS
qaGobiApiCbk.h, 1766
eQCWWAN_ERR_API_MUTEX_TIMEOUT
qmerrno.h, 2146
eQCWWAN_ERR_BUFFER_SZ
qmerrno.h, 2145
eQCWWAN_ERR_CANCEL_OP
qmerrno.h, 2146
eQCWWAN_ERR_DRIVER
qmerrno.h, 2146
eQCWWAN_ERR_ENUM_BEGIN
qmerrno.h, 2145
eQCWWAN_ERR_ENUM_END
qmerrno.h, 2146
eQCWWAN_ERR_FILE_COPY
qmerrno.h, 2145
eQCWWAN_ERR_FILE_OPEN
qmerrno.h, 2145
eQCWWAN_ERR_GENERAL
qmerrno.h, 2145
eQCWWAN_ERR_INTERNAL
qmerrno.h, 2145
eQCWWAN_ERR_INVALID_ARG
qmerrno.h, 2145
eQCWWAN_ERR_INVALID_DEVID
qmerrno.h, 2145
eQCWWAN_ERR_INVALID_FILE
qmerrno.h, 2145
eQCWWAN_ERR_INVALID_QMI_RSP
qmerrno.h, 2145
eQCWWAN_ERR_INVALID_XID
qmerrno.h, 2146
eQCWWAN_ERR_MALFORMED_QMI_RSP
qmerrno.h, 2145
eQCWWAN_ERR_MEMORY
qmerrno.h, 2145
eQCWWAN_ERR_MULTIPLE_DEVICES
qmerrno.h, 2146
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
qmerrno.h, 2146
eQCWWAN_ERR_NO_CANCELABLE_OP
qmerrno.h, 2146
eQCWWAN_ERR_NO_CONNECTION
qmerrno.h, 2145
eQCWWAN_ERR_NO_DEVICE
qmerrno.h, 2145
eQCWWAN_ERR_NO_SIGNAL
qmerrno.h, 2146
eQCWWAN_ERR_NONE
qmerrno.h, 2145
eQCWWAN_ERR_NULL_TLV
qmerrno.h, 2149
eQCWWAN_ERR_OFFLINE
qmerrno.h, 2146
eQCWWAN_ERR_PDU_GENERATION
qmerrno.h, 2146
eQCWWAN_ERR_QMI_ABORTED
qmerrno.h, 2146
eQCWWAN_ERR_QMI_ACCESS_DENIED
qmerrno.h, 2148
eQCWWAN_ERR_QMI_ACK_NOT_SENT
qmerrno.h, 2148
eQCWWAN_ERR_QMI_ARG_TOO_LONG
qmerrno.h, 2146
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
qmerrno.h, 2147
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
qmerrno.h, 2147
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED

qmerrno.h, 2148
eQCWWAN_ERR_QMI_CALL_FAILED
qmerrno.h, 2146
eQCWWAN_ERR_QMI_CARD_BUSY_RSP
qmerrno.h, 2149
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
qmerrno.h, 2148
eQCWWAN_ERR_QMI_CAT_END
qmerrno.h, 2149
eQCWWAN_ERR_QMI_CAT_START
qmerrno.h, 2149
eQCWWAN_ERR_QMI_CAUSE_CODE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
qmerrno.h, 2146
eQCWWAN_ERR_QMI_CONNECT
qmerrno.h, 2145
eQCWWAN_ERR_QMI_DEVICE_IN_USE
qmerrno.h, 2146
eQCWWAN_ERR_QMI_DEVICE_NOT_READY
qmerrno.h, 2147
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
qmerrno.h, 2147
eQCWWAN_ERR_QMI_DISABLED
qmerrno.h, 2147
eQCWWAN_ERR_QMI_ENCODING
qmerrno.h, 2147
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
qmerrno.h, 2149
eQCWWAN_ERR_QMI_EVENT_REG_FAILED
qmerrno.h, 2149
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
qmerrno.h, 2148
eQCWWAN_ERR_QMI_FDN_RESTRICT
qmerrno.h, 2148
eQCWWAN_ERR_QMI_FLOW_SUSPENDED
qmerrno.h, 2147
eQCWWAN_ERR_QMI_GENERAL
qmerrno.h, 2147
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
qmerrno.h, 2148
eQCWWAN_ERR_QMI_IFACE
qmerrno.h, 2145
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
qmerrno.h, 2148
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INCORRECT_PIN
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
qmerrno.h, 2148
eQCWWAN_ERR_QMI_INJECT_TIMEOUT
qmerrno.h, 2148
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INTERNAL
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_ARG
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
qmerrno.h, 2149
eQCWWAN_ERR_QMI_INVALID_HANDLE
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_ID
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_INDEX
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_OPERATION
qmerrno.h, 2148
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_PINID
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_PROFILE
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_QMI_CMD
qmerrno.h, 2148
eQCWWAN_ERR_QMI_INVALID_QOS_ID
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_TECH_PREF
qmerrno.h, 2146
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
qmerrno.h, 2149
eQCWWAN_ERR_QMI_INVALID_TRANSITION
qmerrno.h, 2147
eQCWWAN_ERR_QMI_INVALID_TX_ID
qmerrno.h, 2146
eQCWWAN_ERR_QMI_MALFORMED_MSG
qmerrno.h, 2146
eQCWWAN_ERR_QMI_MAX
qmerrno.h, 2148

eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
qmerrno.h, 2147
eQCWWAN_ERR_QMI_MISSING_ARG
qmerrno.h, 2146
eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, 2148
eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, 2148
eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, 2147
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, 2146
eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, 2147
eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, 2146
eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, 2146
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
qmerrno.h, 2146
eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, 2148
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, 2148
eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, 2146
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, 2146
eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, 2148
eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, 2146
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED
qmerrno.h, 2146
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED
qmerrno.h, 2146
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, 2148
eQCWWAN_ERR_QMI_OUT_OF_CALL
qmerrno.h, 2146
eQCWWAN_ERR_QMI_PIN_BLOCKED
qmerrno.h, 2147
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
qmerrno.h, 2147
eQCWWAN_ERR_QMI_POLICY_MISMATCH
qmerrno.h, 2148
eQCWWAN_ERR_QMI_REQ
qmerrno.h, 2145
eQCWWAN_ERR_QMI_REQ_SCH
qmerrno.h, 2145
eQCWWAN_ERR_QMI_REQ_TO
qmerrno.h, 2145
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
qmerrno.h, 2147
eQCWWAN_ERR_QMI_RSP
qmerrno.h, 2145
eQCWWAN_ERR_QMI_RSP_TO
qmerrno.h, 2145
eQCWWAN_ERR_QMI_SEGMENT_ORDER
qmerrno.h, 2148
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
qmerrno.h, 2148
eQCWWAN_ERR_QMI_SESSION_INACTIVE
qmerrno.h, 2147
eQCWWAN_ERR_QMI_SESSION_INVALID
qmerrno.h, 2147
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
qmerrno.h, 2147
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
qmerrno.h, 2148
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
qmerrno.h, 2147
eQCWWAN_ERR_QMI_SMSC_ADDR
qmerrno.h, 2148
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
qmerrno.h, 2148
eQCWWAN_ERR_QMI_TPDU_TYPE
qmerrno.h, 2148
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
qmerrno.h, 2146
eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, 2147
eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, 2149
eQCWWAN_ERR_RESET
qmerrno.h, 2146
eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_END
qmerrno.h, 2149
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID

qmerrno.h, 2148
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_START
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, 2148
eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, 2148
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
qmerrno.h, 2149
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
qmerrno.h, 2149
eQCWWAN_ERR_SWIDCS_END
qmerrno.h, 2149
eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, 2149
eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, 2149
eQCWWAN_ERR_SWIDCS_START
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_END
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_SECBOOT_IMAGE_NOT_SIGNED
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, 2149
eQCWWAN_ERR_SWIIM_START
qmerrno.h, 2149
eQCWWAN_ERR_SWISM_END
qmerrno.h, 2149
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND
qmerrno.h, 2149
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, 2149
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, 2149
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED
qmerrno.h, 2149
eQCWWAN_ERR_SWISMS_START
qmerrno.h, 2149
eQMI_LOC_SESS_STATUS_FAILURE
loc.h, 1620
eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, 1620
eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, 1620
eQMI_LOC_SESS_STATUS_TIMEOUT
loc.h, 1620
eQOS
common.h, 1550
eREQ
common.h, 1551
eRMS
common.h, 1550
eRSP
common.h, 1551
eSAR
common.h, 1550

eSMS
common.h, 1550
eSWIAUDIO
common.h, 1550
eSWIDMS
common.h, 1550
eSWILOC
common.h, 1550
eSWIM2MCMD
common.h, 1550
eSWIM2MCMD_AVC2
common.h, 1550
eSWIOMA
common.h, 1550
eSWIOMAEXT
common.h, 1550
eSYS_SRV_DOMAIN_CAMPED
qaGobiApiNas.h, 1908
eSYS_SRV_DOMAIN_CS_ONLY
qaGobiApiNas.h, 1908
eSYS_SRV_DOMAIN_CS_PS
qaGobiApiNas.h, 1908
eSYS_SRV_DOMAIN_NO_SRV
qaGobiApiNas.h, 1908
eSYS_SRV_DOMAIN_PS_ONLY
qaGobiApiNas.h, 1908
eSYS_SRV_DOMAIN_UNKNOWN
qaGobiApiNas.h, 1908
eSetServiceAutomaticTrackingDisable
qaGobiApiPds.h, 1945
eSetServiceAutomaticTrackingEnable
qaGobiApiPds.h, 1945
eTIMEOUT_10_S
common.h, 1551
eTIMEOUT_20_S
common.h, 1551
eTIMEOUT_2_S
common.h, 1551
eTIMEOUT_300_S
common.h, 1551
eTIMEOUT_30_S
common.h, 1551
eTIMEOUT_5_S
common.h, 1551
eTIMEOUT_60_S
common.h, 1551
eTIMEOUT_8_S
common.h, 1551
eTIMEOUT_DEFAULT
common.h, 1551
eTLV_3GPP_NETWORK_INFO
qaNasPerformNetworkScan.h, 2143
eTLV_CBK_ALPHA_IDENTIFIER
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_DISPLAY_TEXT
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_END_PROACTIVE_SESSION
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_GET_IN_KEY
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_GET_INPUT
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_LANGUAGE_NOTIFICATION
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_REFRESH
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_SELECT_ITEM
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_SETUP_EVENT_LIST
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_SETUP_IDLE_MODE_TEXT
qaCbkCatEventReportInd.h, 1710
eTLV_CBK_SETUP_MENU
qaCbkCatEventReportInd.h, 1710
eTLV_END_PROACTIVE_SESSION_LENGTH
qaCbkCatEventReportInd.h, 1710
eTLV_IND_OMA_DM_CONFIG
qaCbkSwiOmaDmEventReportInd.h, 1711
eTLV_IND_OMA_DM_FOTA
qaCbkSwiOmaDmEventReportInd.h, 1711
eTLV_IND_OMA_DM_NOT
qaCbkSwiOmaDmEventReportInd.h, 1711
eTLV_LTE_OPERATIONAL_MODE
qaNasGetRFBandInfo.h, 2142
eTLV_REFRESH_LENGTH
qaCbkCatEventReportInd.h, 1710
eTLV_RF_BAND_INFO
qaNasGetRFBandInfo.h, 2142
eTLV_RF_BANDWIDTH_INFO
qaNasGetRFBandInfo.h, 2142
eTLV_RF_DEDICATED_BAND_INFO
qaNasGetRFBandInfo.h, 2142
eTLV_RF_EXT_FORMAT_BAND_INFO
qaNasGetRFBandInfo.h, 2142
eTLV_SETUP_EVENT_LIST_LENGTH
qaCbkCatEventReportInd.h, 1710
eTMD
common.h, 1550
eUIM
common.h, 1550
eVOICE
common.h, 1550
eWDS
common.h, 1550
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID-
_IDENT_FOR_PROFILE
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT-
_DEFINED
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_-
PROFILES
qmerrno.h, 2150

eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_END
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_FAIL
qmerrno.h, 2150
eWDS_ERR_PROFILE_REG_RESULT_LIST_END
qmerrno.h, 2150
ECIOThresListLen
ECIOThresh, 250
nas_ECIOThresh, 465
ECIOThresh, 250
ECIOThresListLen, 250
pECIOThresList, 250
ECTCallState
ECTNum, 251
voice_ECTNum, 1429
ECTNum, 250
ECTCallState, 251
number, 251
presentationInd, 251
eDevState
qaGobiApiCbk.h, 1727
eDevice
sGetDeviceSeriesResult, 934
eGetDeviceSeries
qaGobiApiFms.h, 1861
eGobiDeviceSeries
qaGobiApiFms.h, 1858
eGobiImageCarrier
qaGobiApiFms.h, 1858
eGobiImageGPS
qaGobiApiFms.h, 1859
eGobiImageRegion
qaGobiApiFms.h, 1859
eGobiImageTech
qaGobiApiFms.h, 1859
eLOG_LEVEL
common.h, 1550
EMTlv
NASQmiCbkNasSystemSelPrefInd, 599
eQCWWANError
qmerrno.h, 2145
eQMI_SVC
common.h, 1550
eQMISARRFState
qaGobiApiSar.h, 1967
eQaQMIService
qaGobiApiCbk.h, 1766
ERI_DATA_MAX_SIZE
dms.h, 1561
ERIFileparams, 253
pFile, 253
pFileSize, 253
qaGobiApiDms.h, 1819
eSMSEventType
qaGobiApiCbk.h, 1728
ESNString
unpack_dms_GetDeviceSerialNumbers_t, 1112
eSYS_SRV_DOMAIN
qaGobiApiNas.h, 1908
ETWSPLMNInfo
eTWSPLMNInfoTlv, 255
eTWSPLMNInfoTlv, 254
ETWSPLMNInfo, 255
TlvPresent, 255
ETWSPLMNTlv
unpack_sms_SetNewSMSCallback_ind_t, 1279
ETWSTlv
unpack_sms_SetNewSMSCallback_ind_t, 1279
eTimeout
common.h, 1550
EVENT_MASK_CARD
qaGobiApiCbk.h, 1725
eValid
LibPackTFTIDParams, 376
TFTIDParams, 1031
EarMute
GetAudioProfileResp, 274
GetM2MAudioProfileResp, 291

GetM2MAVMuteResp, 293
SetAudioProfileReq, 913
SetM2MAVMuteReq, 924
earfcn
infoInterFreq, 344
LTEInfoIntrafreq, 420
ltePCI, 423
nas_infoInterFreq, 481
nas_LTEInfoIntrafreq, 488
nas_ltePCI, 491
nas_umtsLTENbrCell, 558
umtsLTENbrCell, 1093
earfcn0
lteEARFCN, 414
nas_lteEARFCN, 482
earfcn1
lteEARFCN, 414
nas_lteEARFCN, 482
earlyWakeupTime
dms_PSMEarlyWakeupTimeTlv, 234
PSMEarlyWakeupTimeTlv, 826
ecio
CDMASSInfo, 175
cdmaSSInfo, 175
ecioListElement, 249
HDRSSInfo, 319
hdrSSInfo, 320
nas_ecioListElement, 464
nas_rxInfo, 536
nas_UMTSInfo, 556
rxInfo, 891
TDSCDMASigInfoExt, 1025
tdscdmaSigInfoExt, 1026
UMTSInfo, 1091
ecioDelta
nas_SLQSSignalStrengthsIndReq, 543
SLQSSignalStrengthsIndReq, 954
ecioInfo
nas_SLQSSignalStrengthsInformation, 544
SLQSSignalStrengthsInformation, 955
ecioList
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
ecioListElement, 249
ecio, 249
radioIf, 250
ecioListLen
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
ecioThresholdList
nas_SLQSSignalStrengthsIndReq, 543
SLQSSignalStrengthsIndReq, 954
ecioThresholdListLen
nas_SLQSSignalStrengthsIndReq, 543
SLQSSignalStrengthsIndReq, 954
egprsSupp
GSMSysInfo, 311
nas_GSMSysInfo, 474
egprsSuppValid
GSMSysInfo, 311
nas_GSMSysInfo, 474
ehrpdMTUSize
ehrpdMTUSizeTlv, 251
swidms_ehrpdMTUSizeTlv, 997
ehrpdMTUSizeTlv, 251
ehrpdMTUSize, 251
TlvPresent, 251
elevation
loc_satelliteInfo, 395
satelliteInfo, 898
EmerMode
NASEmergencyModeTlv, 568
emmConnState
LTEInfo, 418
nas_LTEInfo, 486
emmState
LTEInfo, 418
nas_LTEInfo, 486
emmSubState
LTEInfo, 418
nas_LTEInfo, 486
Enable
SetM2MAudioLPBKReq, 922
enable
pack_pds_SetPDSState_t, 703
pack_qos_SLQSSetQosEventCallback_t, 714
EnableState
unpack_dms_PSMCfgChange_ind_t, 1127
enableState
dms_PSMEnableStateTlv, 235
PSMEnableStateTlv, 826
EnableStateInd
dms_PSMEnableStateIndTlv, 234
enabled
pack_rms_SetSMSWake_t, 714
unpack_rms_GetSMSWake_t, 1275
unpack_wds_GetMobileIPProfile_t, 1360
EnabledStatus
unpack_pds_SetPdsState_Ind_t, 1249
EncryptProt
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
EncryptedPIN1
pack_uim_ChangePin_t, 731
pack_uim_SetPinProtection_t, 733
pack_uim_UnblockPin_t, 740
encryptedPIN1, 252
pin1Len, 252
pin1Val, 252
EndPointType
qos_BindDataPortPeripheralEndPointID_t, 866
EndProactiveSession
CatEndProactiveSessionTlv, 161
EngineState
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253

engineState
QmiCbkLocEngineStateInd, 840
unpack_loc_EngineState_Ind_t, 1165
entries
t_Sv, 1023
eqmiCbkSetStatus
sms.h, 2167
eriData
eriDataparams, 253
eriDataLen
eriDataparams, 253
eriDataparams, 252
eriData, 253
eriDataLen, 253
eriFile
unpack_dms_SLQSGetERIFile_t, 1139
error
unpack_wds_GetLastMobileIPError_t, 1358
errorClass
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
errorRate
errorRateListElement, 254
nas_errorRateListElement, 466
errorRateInfo
nas_SLQSSignalStrengthsInformation, 544
SLQSSignalStrengthsInformation, 956
errorRateList
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
errorRateListElement, 253
errorRate, 254
radioIf, 254
errorRateListLen
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
errorState
slotInf, 939
slotInfo, 941
uim_slotInfo, 1066
esn
unpack_dms_GetSerialNumbers_t, 1123
esnSize
serialNumbersInfo, 904
unpack_dms_GetDeviceSerialNumbers_t, 1112
EspSpi
unpack_qos_swiQosFilter_t, 1267
EtwsMessageInfo
sMSEtwsMessageTlv, 976
event
unpack_qos_SLQSSetQosPriEventCallback_ind_-t, 1264
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1265
event_Index
QmiCbkCatEventStatusReportInd, 833
EventID
CatCommonEventTlv, 160
EventLength
CatCommonEventTlv, 160
eventMask
CATEventDataType, 161
pack_uim_SLQSUIMEventRegister_t, 734
UIMEventRegisterReqResp, 1072
unpack_uim_SLQSUIMEventRegister_t, 1303
eventRegister
LOCEventRegisterReqResp, 403
pack_loc_EventRegister_t, 652
eventType
unpack_swioma_SLQSOMADMAlertCallback_ind-_t, 1293
evrcCapability
prefVoiceSO, 810
voice_prefVoiceSO, 1438
executingImage
FMSImageIDEntries, 264
ImageIDEntries, 327
exponent
pktErrRate, 801
unpack_qos_pktErrRate_t, 1257
extBit
calledPartySubAdd, 148
voice_calledPartySubAdd, 1410
extDispInfo
extDispRecInfo, 255
voice_extDispRecInfo, 1429
extDispInfoLen
extDispRecInfo, 255
voice_extDispRecInfo, 1429
extDispRecInfo, 255
dispType, 255
extDispInfo, 255
extDispInfoLen, 255
ExtErrorCode
PackCreateProfileOut, 786
extPowerState
LOCExtPowerStateReqResp, 403
pack_loc_SetExtPowerState_t, 652
extendedErrorCode
unpack_wds_SLQSDeleteProfile_t, 1367

FIRST_INSTANCE
qaGobiApiCbk.h, 1725
FLOAT
SwiDataTypes.h, 2186
FMSImageElement, 262
buildId, 262
buildIdLength, 262
imageId, 262
imageType, 262
FMSImageIDEntries, 263
executingImage, 264
imageIDElement, 264
imageIDSize, 264
imageType, 264
maxImages, 264

FMSImageIdElement, 262
buildID, 263
buildIDLength, 263
failureCount, 263
imageID, 263
storageIndex, 263
FMSImageList, 264
imageIDEntries, 265
listSize, 265
FMSPrefImageList, 265
listEntries, 265
listSize, 265
FORBIDDEN_INDEX
qaNasPerformNetworkScan.h, 2143
FOTAUpdate
_SLQSOMADMSettingsReqParams, 96
_SLQSOMADMSettingsReqParams3, 97
pack_swioma_SLQSOMADMSetSettings_t, 729
unpack_swioma_SLQSOMADMGetSettings_t, 1297
FOTAdownload
_SLQSOMADMSettingsReqParams, 95
_SLQSOMADMSettingsReqParams3, 96
pack_swioma_SLQSOMADMSetSettings_t, 729
unpack_swioma_SLQSOMADMGetSettings_t, 1297
FSNumber
FactorySequenceNumber, 256
facility
pack_dms_UIMGetControlKeyStatus_t, 643
pack_dms_UIMSetControlKeyProtection_t, 644
pack_dms_UIMUnblockControlKey_t, 646
facilityCk
pack_dms_UIMSetControlKeyProtection_t, 644
pack_dms_UIMUnblockControlKey_t, 646
facilityState
pack_dms_UIMSetControlKeyProtection_t, 644
unpack_dms_UIMGetControlKeyStatus_t, 1151
FactorySequenceNumber, 255
FSNumber, 256
failureCount
FMSImageIdElement, 263
ImageIdElement, 326
failureReason
ssdatasession_params, 991
failureReasonv4
ssdatasession_params, 992
failureReasonv6
ssdatasession_params, 992
family
pack_wds_GetDefaultProfileNum_t, 761
pack_wds_SetDefaultProfileNum_t, 767
feature
depersonalizationInformation, 225
personalizationStatus, 794
uim_depersonalizationInformation, 1052
uim_personalizationStatus, 1058
fileAttributes, 256
fileID, 259
fileSize, 259
fileType, 259
rawLen, 259
rawValue, 259
recordCount, 259
recordSize, 259
secActivate, 259
secActivateMask, 259
secDeactivate, 259
secDeactivateMask, 259
secIncrease, 259
secIncreaseMask, 259
secRead, 259
secReadMask, 259
secWrite, 259
secWriteMask, 259
fileID
fileAttributes, 259
fileInfo, 260
uim_fileAttributes, 1056
uim_fileInfo, 1057
fileIndex
pack_uim_ReadTransparent_t, 732
pack_uim_SLQSUIMGetFileAttributes_t, 736
UIMGetFileAttributesReq, 1074
UIMReadTransparentReq, 1078
fileInfo, 259
fileID, 260
path, 260
pathLen, 260
fileSize
fileAttributes, 259
uim_fileAttributes, 1056
fileType
fileAttributes, 259
uim_fileAttributes, 1056
fill_pack_ctx
common.h, 1551
fill_sdu_hdr
common.h, 1551
filterId
LibPackTFTIDParams, 377
TFTIDParams, 1031
Firmware Management Service (FMS), 48
FirmwareID
fwinfo_s, 266
FirmwareUpdatStat, 260
pImgType, 261
pLogString, 261
pLogStringLen, 262
pRefData, 262
pRefString, 262
pRefStringLen, 262
ResCode, 262
fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocGetAutoStartResp, 1001

SwiLocSetAutoStartReq, 1003
unpack_swiloc_SwiLocGetAutoStart_t, 1292
fix_rate_reported
SwiLocGetAutoStartResp, 1001
unpack_swiloc_SwiLocGetAutoStart_t, 1292
fix_type
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocGetAutoStartResp, 1001
SwiLocSetAutoStartReq, 1003
unpack_swiloc_SwiLocGetAutoStart_t, 1292
fix_type_reported
SwiLocGetAutoStartResp, 1001
unpack_swiloc_SwiLocGetAutoStart_t, 1292
fixAccuracy
pack_pds_StartPDSTrackingSessionExt_t, 711
fixCount
pack_pds_StartPDSTrackingSessionExt_t, 711
fixInterval
pack_pds_StartPDSTrackingSessionExt_t, 711
fixTimeout
pack_pds_StartPDSTrackingSessionExt_t, 711
flags
sensorData, 901
sensorData_t, 902
flowLabel
LibPackTFTIDParams, 377
TFTIDParams, 1031
fms.h, 1611
GetValidFwPriCombinations, 1612
pack_fms_GetImagesPreference, 1612
pack_fms_GetStoredImages, 1613
pack_fms_SetImagesPreference, 1613
unpack_fms_GetImagesPreference, 1613
unpack_fms_GetStoredImages, 1613
unpack_fms_SetImagesPreference, 1613
Forbidden
nas_QmiNas3GppNetworkInfo, 523
SlqsNas3GppNetworkInfo, 945
forceFlag
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 706
ForceRev0
unpack_nas_GetCDMANetworkParameters_t, 1186
ForceXTRADownload
qaGobiApiPds.h, 1945
format
SMSTransferRouteMTMessage, 986
sMSTransferRouteMTMessageInfo, 986
ForwardMac
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
fqdnAddr
PCSCFFQDNAddress, 788
wds_PCSCFFQDNAddress, 1522
fqdnLen
PCSCFFQDNAddress, 788
wds_PCSCFFQDNAddress, 1522
freeSlots
sms_maxStorageSizeResp, 963
smsMaxStorageSizeResp, 979
freq
nas_PhyCaAggPcellInfo, 511
nas_PhyCaAggScellIndType, 513
nas_PhyCaAggScellInfo, 516
NASPhyCaAggPcellInfo, 590
NASPhyCaAggScellIndType, 592
NASPhyCaAggScellInfo, 593
PhyCaAggPcellInfo, 795
PhyCaAggScellIndType, 797
PhyCaAggScellInfo, 800
freqsLen
LTEInfoInterfreq, 418
LTEInfoNeighboringGSM, 421
LTEInfoNeighboringWCDMA, 422
nas_LTEInfoInterfreq, 486
nas_LTEInfoNeighboringGSM, 489
nas_LTEInfoNeighboringWCDMA, 490
fromServiceId
BroadcastConfig, 144
sms_BroadcastConfig, 958
fumoResultCode
omaDmFotaTlvExt, 627
function
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocGetAutoStartResp, 1001
SwiLocSetAutoStartReq, 1003
unpack_swiloc_SwiLocGetAutoStart_t, 1292
function_reported
SwiLocGetAutoStartResp, 1001
unpack_swiloc_SwiLocGetAutoStart_t, 1292
FwAutoCheck
unpack_swioma_SLQSOMADMGetSettings_t, 1297
FwAvailability
unpack_swioma_SLQSOMADMStartSession_t, 1298
fwdloadsize
omaDmFotaTlv, 624
unpack_omaDmFotaTlv_t, 1241
fwinfo_s, 265
Carrier, 266
FirmwareID, 266
GPSCapability, 266
Region, 266
Technology, 266
fwloadComplete
omaDmFotaTlv, 625
unpack_omaDmFotaTlv_t, 1241
fwvers
CurrentImgList, 204
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1144

g
qmifwinfo_s, 856
gDlBitRate
LibPackQosClassID, 375

QosClassID, 867
GERANInfo, 266
arfcn, 267
bsic, 267
cellID, 267
insNmrCellInfo, 267
lac, 268
nmrInst, 268
plmn, 268
rxLev, 268
timingAdvance, 268
GPRSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1377
GPRSQoS, 300
delayClass, 301
meanThroughputClass, 301
peakThroughputClass, 301
precedenceClass, 301
reliabilityClass, 301
GPRSRequestedQoS, 301
delayClass, 302
meanThroughputClass, 302
peakThroughputClass, 302
precedenceClass, 302
reliabilityClass, 302
GPSCapability
fwinfo_s, 266
GPSLPM
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1106
GPSSel
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1106
GPSStateInfo, 302
Altitude, 305
EngineState, 305
glo_almanac_sv_msk, 305
glo_ephemeris_sv_msk, 305
glo_health_sv_msk, 305
glo_visible_sv_msk, 305
gps_almanac_sv_msk, 305
gps_ephemeris_sv_msk, 305
gps_health_sv_msk, 305
gps_visible_sv_msk, 305
HorizontalUncertainty, 305
Iono_valid, 305
Latitude, 305
Longitude, 305
sbas_almanac_sv_msk, 305
sbas_ephemeris_sv_msk, 305
sbas_health_sv_msk, 305
sbas_visible_sv_msk, 306
Time_uncert_ms, 306
TimeStmp_gps_week, 306
TimeStmp_tow_ms, 306
ValidMask, 306
VerticalUncertainty, 306
xtra_start_gps_minutes, 306
xtra_start_gps_week, 306
xtra_valid_duration_hours, 306
GSMRSSIThresh, 308
GSMRSSIThreshListLen, 308
pGSMRSSIThreshList, 308
GSMRSSIThreshListLen
GSMRSSIThresh, 308
nas_GSMRSSIThresh, 470
GSMSSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1225
GSMSrvStatusInfo, 308
isPrefDataPath, 309
srvStatus, 309
trueSrvStatus, 309
GSMSysInfo, 309
cellId, 311
cellIdValid, 311
dtmSupp, 311
dtmSuppValid, 311
egprsSupp, 311
egprsSuppValid, 311
lac, 311
lacValid, 312
MCC, 312
MNC, 312
networkIdValid, 312
regRejectInfoValid, 312
rejCause, 312
rejectSrvDomain, 312
sysInfoGSM, 312
gULBitRate
LibPackQosClassID, 375
QosClassID, 867
GWAOPTlv
NASQmiCbkNasSystemSelPrefInd, 599
GWAcqOrderPref
NASGWAcqOrderPrefTlv, 578
GWAddressV4
unpack_wds_SLQSGetRuntimeSettings_t, 1378
gcDumpStrLen
CrashInfo, 195
gcdumpString
crashInformation, 197
gcdumpStrlen
crashInformation, 197
Generator
GetAudioProfileReq, 273
GetAudioVolTLBConfigReq, 275
GetM2MAudioProfileResp, 291
GetM2MAudioVolumeReq, 292
SetAudioProfileReq, 913
SetAudioVolTLBConfigReq, 914
SetM2MAudioVolumeReq, 924
geoSysIdx
AddCDMASysInfo, 118
AddSysInfo, 118
nas_AddCDMASysInfo, 443
nas_AddSysInfo, 443

geranArfcn
geranInstInfo, 268
nas_geranInstInfo, 468
geranBsicBcc
geranInstInfo, 268
nas_geranInstInfo, 468
geranBsicNcc
geranInstInfo, 268
nas_geranInstInfo, 469
geranInst
nas_UMTSInfo, 556
UMTSInfo, 1091
GeranInstInfo
nas_UMTSInfo, 556
UMTSInfo, 1091
geranInstInfo, 268
geranArfcn, 268
geranBsicBcc, 268
geranBsicNcc, 268
geranRssi, 268
geranRssi
geranInstInfo, 268
nas_geranInstInfo, 469
get_version
common.h, 1551
GetACCOLC
qaGobiApiNas.h, 1909
GetANAAAAuthenticationStatus
qaGobiApiNas.h, 1909
GetActivationState
qaGobiApiDms.h, 1823
getAllCallInfo
arrCallInfo, 133
voice_arrCallInfo, 1403
getAllCallInformation, 269
ALS, 269
Callinfo, 269
isEmpty, 269
GetAllCallRmtPtyName
arrRemotePartyName, 135
voice_arrRemotePartyName, 1405
getAllCallRmtPtyName, 269
callID, 270
RemotePartyName, 270
getAllCallRmtPtyNum, 270
callID, 270
RemotePartyNum, 270
GetAudioPathConfigReq, 270
Item, 271
Profile, 271
GetAudioPathConfigResp, 271
pCodecSTGain, 272
pDTMFTXGain, 273
pECMode, 273
pMICGainSelect, 273
pNSEnable, 273
pRXAGCList, 273
pRXAVCAGCSwitch, 273
pRXAVCList, 273
pRXPCMIIRFltr, 273
pTXAGCList, 273
pTXAVCSwitch, 273
pTXGain, 273
pTXPCMIIRFltr, 273
GetAudioProfileReq, 273
Generator, 273
GetAudioProfileResp, 273
EarMute, 274
MicMute, 274
Profile, 274
Volume, 274
GetAudioVolTLBConfigReq, 275
Generator, 275
Item, 275
Profile, 275
Volume, 275
GetAudioVolTLBConfigResp, 275
ResCode, 276
GetAutoconnect
qaGobiApiWds.h, 2105
GetByteTotals
qaGobiApiWds.h, 2105
GetCDMANetworkParameters
qaGobiApiNas.h, 1910
getCallFWExtInfo, 276
CallFWExtInfo, 276
numInstances, 276
getCallFWInfo, 276
CallFWInfo, 277
numInstances, 277
GetConnectionRate
qaGobiApiWds.h, 2106
GetCustomFeatureV2
unpack_dms_GetCustFeaturesV2_t, 1107
getCustomFeatureV2, 277
pCustSettingInfo, 277
pCustSettingList, 277
pGetCustomInput, 277
getCustomInput, 278
cust_id, 278
list_type, 278
getDUNCallInfoReq, 278
Mask, 279
pReportChannelRate, 279
pReportConnStatus, 279
pReportDataBearerTech, 279
pReportDormStatus, 279
pTransferStatInd, 279
getDUNCallInfoResp, 280
pCallEndReason, 282
pChannelRate, 282
pConnectionStatus, 282
pDataBearerTech, 282
pDormancyStatus, 282
pLastCallDataBearerTech, 282
pLastCallRXOKBytesCnt, 282

pLastCallTXOKBytesCnt, 282
pMdmCallDurationActive, 282
pRXOKBytesCount, 282
pTXOKBytesCount, 282
GetDataBearerTechnology
qaGobiApiWds.h, 2107
GetDefaultProfile
qaGobiApiWds.h, 2107
GetDefaultProfileLTE
qaGobiApiWds.h, 2109
GetDefaultProfileNum
qaGobiApiWds.h, 2111
GetDeviceCapabilities
qaGobiApiDms.h, 1824
GetDormancyState
qaGobiApiWds.h, 2111
getDyingGaspCfg, 282
pDestSMSContent, 283
pDestSMSNum, 283
getDyingGaspStatistics, 283
pSMSAttemptedFlag, 283
pTimeStamp, 284
GetErrRateResp, 284
pCDMAFrameErrRate, 284
pGSMBER, 284
pHDRPackErrRate, 284
pWCDMABER, 284
GetFirmwareRevision
qaGobiApiDms.h, 1825
GetFirmwareRevisions
qaGobiApiDms.h, 1825
GetHRPDStatsResp, 285
pDRCParams, 285
pPilotSetData, 285
pUATI, 285
GetHardwareRevision
qaGobiApiDms.h, 1826
GetHomeNetwork
qaGobiApiNas.h, 1911
GetHomeNetwork3GPP2
qaGobiApiNas.h, 1912
GetIMSI
qaGobiApiDms.h, 1827
GetIMSSMSConfigParams, 285
pPhoneCtxtURI, 286
pPhoneCtxtURILen, 286
pSMSFormat, 286
pSMSOverIPNwInd, 286
pSettingResp, 286
GetIMSUserConfigParams, 286
pIMSDomain, 287
pIMSDomainLen, 287
pSettingResp, 287
GetIMSVoIPConfigResp, 287
pAmrMode, 289
pAmrOctetAligned, 289
pAmrWBMode, 289
pAmrWBOctetAligned, 289
pAmrWbEnable, 289
pMinSessionExpiryTimer, 289
pRTPRTCPInactTimer, 289
pRingBackTimer, 289
pRingingTimer, 289
pScrAmrEnable, 289
pScrAmrWbEnable, 289
pSessionExpiryTimer, 289
pSettingResp, 289
GetIPAddressLTE
qaGobiApiWds.h, 2112
GetImageStore
qaGobiApiFms.h, 1862
GetImagesPreference
qaGobiApiFms.h, 1861
getIndicationRegResp
qaGobiApiSms.h, 1971
GetInstIDResp, 289
pIPFamily, 289
pInstanceID, 289
GetLastMobileIPError
qaGobiApiWds.h, 2112
GetM2MAVMuteReq, 292
Profile, 292
GetM2MAVMuteResp, 293
CwtMute, 293
EarMute, 293
MicMute, 293
GetM2MAudioProfileReq, 290
pGenerator, 290
GetM2MAudioProfileResp, 290
CwtMute, 291
EarMute, 291
Generator, 291
MicMute, 291
Profile, 291
Volume, 291
GetM2MAudioVolumeReq, 291
Generator, 292
Profile, 292
GetM2MAudioVolumeResp, 292
Level, 292
GetM2MSpkrGainReq, 293
Profile, 294
GetM2MSpkrGainResp, 294
Value, 294
GetManufacturer
qaGobiApiDms.h, 1827
GetMobileIP
qaGobiApiWds.h, 2113
GetMobileIPProfile
qaGobiApiWds.h, 2113
GetModelID
qaGobiApiDms.h, 1828
getMsgWaitingInfo, 294
msgWaitInfo, 295
numInstances, 295
GetNetworkPreference

qaGobiApiNas.h, 1914
GetNetworkTime
qaGobiApiDms.h, 1828
GetNetworkTimeResp, 295
p3GPP2TimeInfo, 295
p3GPPTimeInfo, 295
GetOfflineReason
qaGobiApiDms.h, 1829
GetPDSDefaults
qaGobiApiPds.h, 1945
GetPDSState
qaGobiApiPds.h, 1946
GetPRLVersion
qaGobiApiDms.h, 1830
GetPacketStatistics
qaGobiApiWds.h, 2115
GetPacketStatus
qaGobiApiWds.h, 2115
GetPortAutomaticTracking
qaGobiApiPds.h, 1946
GetPower
qaGobiApiDms.h, 1829
GetProfileSettingIn
qaGobiApiWds.h, 2101
GetProfileSettingOut
qaGobiApiWds.h, 2102
GetQmapIsSupported
qaGobiApiDcs.h, 1806
GetRFInfo
qaGobiApiNas.h, 1915
GetRegMgrConfigParams, 295
pIMSTestMode, 296
pPCSCFPort, 296
pPriCSCFPortName, 296
pPriCSCFPortNameLen, 296
pSettingResp, 296
GetSIPConfigResp, 296
pSIPLocalPort, 297
pSettingResp, 297
pSigCompEnabled, 297
pSubscribeTimer, 297
pTimerSIPReg, 297
pTimerT1, 297
pTimerT2, 297
pTimerTf, 297
GetSMSCAddress
qaGobiApiSms.h, 1974
GetSMSWake
qaGobiApiRms.h, 1965
GetSerialNumbers
qaGobiApiDms.h, 1830
GetServiceAutomaticTracking
qaGobiApiPds.h, 1947
GetServingNetwork
qaGobiApiNas.h, 1915
GetServingNetworkCapabilities
qaGobiApiNas.h, 1916
GetSessionDuration
qaGobiApiWds.h, 2116
GetSessionIDResp, 296
pSessionIDv4, 296
pSessionIDv6, 296
GetSessionState
qaGobiApiWds.h, 2116
GetSignalStrengths
qaGobiApiNas.h, 1917
GetStoredImages
qaGobiApiFms.h, 1862
getTransLayerInfoResp
qaGobiApiSms.h, 1972
getTransNWRegInfoResp
qaGobiApiSms.h, 1972
GetValidFwPriCombinations
fms.h, 1612
GetVoiceNumber
qaGobiApiDms.h, 1831
GetXTRAAutomaticDownload
qaGobiApiPds.h, 1947
GetXTRANetwork
qaGobiApiPds.h, 1948
GetXTRAValidity
qaGobiApiPds.h, 1948
glo_almanac_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
glo_ephemeris_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
glo_health_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
glo_visible_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
globalCellId
LTEInfoIntrafreq, 420
nas_LTEInfoIntrafreq, 488
glog
common.h, 1553
gloglvl
common.h, 1553
GnssData, 298
mask, 299
gnssSvId
loc_satelliteInfo, 395
satelliteInfo, 898
gnssSvInfoNotification, 299
bAltitudeAssumed, 300
pSatelliteInfo, 300
gnssSvUsedList
loc_svUsedforFix, 398
svUsedforFix_s, 995
gnssSvUsedList_len
loc_svUsedforFix, 398
svUsedforFix_s, 995
Gpp2TimeZone

qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
GppNetworkDSTAdjustment
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
GppTimeZone
qaQmiServingSystemParam, 832
unpack_nas_SLQSGetServingSystem_t, 1208
gps_almanac_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
gps_ephemeris_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
gps_health_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
gps_visible_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
GpsEnable
custFeaturesInfo, 211
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1106
gpsTime
qaGobiApiCbk.h, 1728
gpsTime_s, 306
gpsTimeOfWeekMs, 306
gpsWeek, 306
gpsTimeOfWeekMs
gpsTime_s, 306
loc_gpsTime, 387
t_gpsTime, 1022
gpsWeek
gpsTime_s, 306
loc_gpsTime, 387
t_gpsTime, 1022
grntDownlinkBitrate
LibPackUMTSQoS, 379
UMTSMinQoS, 1095
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
grntUplinkBitrate
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
gsmAmrStat
curAMRConfig, 202
voice_curAMRConfig, 1426
GsmCellInfo
lteGsmCellInfo, 415
nas_lteGsmCellInfo, 484
gsmCellInfo, 306
arfcn, 307
band1900, 307
bsicId, 307
cellIdValid, 307
rssi, 307
srxlev, 307
gsmUmtsDl
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
gsmUmtsUl
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
guaranteedRate
dataRate, 220
unpack_qos_dataRate_t, 1255
gwAddressV6
IPV6GWAddressInfo, 349
wds_IPV6GWAddressInfo, 1521
gwV6PrefixLen
IPV6GWAddressInfo, 349
wds_IPV6GWAddressInfo, 1521
gyroAcceptReady
qaGobiApiCbk.h, 1728
gyroAcceptReady_s, 312
batchPerSec, 312
injectEnable, 312
samplesPerBatch, 313
gyroData
pack_loc_SLQSLOCInjectSensorData_t, 660
gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 660
gyroTempAcceptReady
qaGobiApiCbk.h, 1728
gyroTempAcceptReady_s, 313
batchPerSec, 313
injectEnable, 313
samplesPerBatch, 313
gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 661

HASPI
unpack_wds_GetMobileIPProfile_t, 1360
HAState
unpack_wds_GetMobileIPProfile_t, 1360
HDOP
loc_precisionDilution, 393
precisionDilution_s, 807
HDRECIOThresh, 313
HDRECIOThreshListLen, 314
pHDRECIOThreshList, 314
HDRECIOThreshListLen
HDRECIOThresh, 314
nas_HDRECIOThresh, 474
HDRIOThresh, 314
HDRIOThreshListLen, 314
pHDRIOThreshList, 314
HDRIOThreshListLen
HDRIOThresh, 314
nas_HDRIOThresh, 475
HDRPersonalityInd, 314
pCurrentPersonality, 315
pPersonalityListLength, 315
pProtocolSubtypeElement, 315

HDRPersonalityResp, 315
pCurrentPersonality, 315
pPersonalityListLength, 315
pProtocolSubtypeElement, 315
HDRProtSubtypResp, 315
pAppSubType, 316
pCurrentPrsnlty, 316
pPersonalityListLength, 316
pProtoSubTypElmnt, 316
HDRRSSIThresh, 316
HDRRSSIThreshListLen, 317
pHDRRSSIThreshList, 317
HDRRSSIThreshListLen
HDRRSSIThresh, 317
nas_HDRRSSIThresh, 476
HDRSINRThresListLen
HDRSINRThresh, 317
nas_HDRSINRThresh, 477
HDRSINRThresh, 317
HDRSINRThresListLen, 317
pHDRSINRThresList, 317
HDRSINRThreshListLen
HDRSINRThreshold, 318
nas_HDRSINRThreshold, 478
HDRSINRThreshold, 317
HDRSINRThreshListLen, 318
pHDRSINRThreshList, 318
HDRSSInfo, 318
ecio, 319
io, 319
rssi, 319
sinr, 319
unpack_nas_SLQSNasGetSigInfo_t, 1225
HDRSysInfo, 320
hdrActiveProt, 322
hdrActiveProtValid, 322
hdrPersonality, 322
hdrPersonalityValid, 322
is856SysId, 322
is856SysIdValid, 322
isSysPrlMatch, 322
isSysPrlMatchValid, 322
sysInfoHDR, 322
HTTPStatus
_SwiAvmsEventInd, 102
HWVersion
DeviceConfigDetail, 227
nas_DeviceConfigDetail, 463
HardwareControlledMode
unpack_dms_GetPower_t, 1121
has_LpmFlag
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
has_PersistentLpm
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
has_PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 1248
has_PowerOffMode
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
has_SessionStatus
unpack_pds_SetEventReport_Ind_t, 1248
has_Wdisable
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
has_accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 661
has_acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 661
has_accleroData
pack_loc_SLQSLOCInjectSensorData_t, 661
has_altitudeSrcInfo
pack_loc_SLQSLOCInjectPosition_t, 658
has_altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 658
has_altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 658
has_confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 662
has_dLatitude
unpack_pds_SetEventReport_Ind_t, 1248
has_dLongitude
unpack_pds_SetEventReport_Ind_t, 1248
has_gyroData
pack_loc_SLQSLOCInjectSensorData_t, 661
has_gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 661
has_gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 661
has_horConfidence
pack_loc_SLQSLOCInjectPosition_t, 658
has_horReliability
pack_loc_SLQSLOCInjectPosition_t, 658
has_horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 658
has_latitude
pack_loc_SLQSLOCInjectPosition_t, 658
has_longitude
pack_loc_SLQSLOCInjectPosition_t, 658
has_opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 661
has_posSrc
unpack_pds_SetEventReport_Ind_t, 1248
has_positionSrc
pack_loc_SLQSLOCInjectPosition_t, 658
has_rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 658
has_rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 658
has_serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 654
has_timestampAge
pack_loc_SLQSLOCInjectPosition_t, 658
has_timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 658
has_vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 658
has_vertRelicability
pack_loc_SLQSLOCInjectPosition_t, 658

has_vertUnc
pack_loc_SLQSLOCInjectPosition_t, 659
hdrActiveProt
HDRSysInfo, 322
nas_HDRSysInfo, 479
hdrActiveProtValid
HDRSysInfo, 322
nas_HDRSysInfo, 479
hdrHybrid
detailSvcInfo, 226
nas_detailSvcInfo, 462
HdrPersonality
unpack_nas_SLQSGetServingSystem_t, 1208
hdrPersonality
HDRSysInfo, 322
nas_HDRSysInfo, 479
NASServingSystemInfo, 601
qaQmiServingSystemParam, 832
ServingSystemInfo, 906
hdrPersonalityValid
HDRSysInfo, 322
nas_HDRSysInfo, 479
hdrSSInfo, 319
ecio, 320
io, 320
rssi, 320
sinr, 320
hdrSrvStatus
detailSvcInfo, 226
nas_detailSvcInfo, 462
healthStatus
loc_satelliteInfo, 395
satelliteInfo, 898
helper_get_error_code
common.h, 1551
helper_get_error_reason
common.h, 1551
helper_get_req_str
common.h, 1551
helper_get_resp_ctx
common.h, 1552
helper_get_xid
common.h, 1552
helper_set_log_func
common.h, 1552
helper_set_log_lvl
common.h, 1552
homeOrigVoiceSO
prefVoiceSO, 810
voice_prefVoiceSO, 1438
homePageVoiceSO
prefVoiceSO, 810
voice_prefVoiceSO, 1438
homeSIDNID, 322
numInstances, 323
SidNid, 323
horConfidence
pack_loc_SLQSLOCInjectPosition_t, 659
horReliability
pack_loc_SLQSLOCInjectPosition_t, 659
horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 659
HorizontalUncertainty
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
hotSwap
hotSwapStatus, 323
uim_hotSwapStatus, 1057
hotSwapLength
hotSwapStatus, 323
uim_hotSwapStatus, 1057
hotSwapStatus, 323
hotSwap, 323
hotSwapLength, 323
hour
nas_timeInfo, 553
nas_UniversalTime, 559
timeInfo, 1033
UniversalTime, 1101
hrpdMTUSize
hrpdMTUSizeTlv, 324
swidms_hrpdMTUSizeTlv, 997
hrpdMTUSizeTlv, 323
hrpdMTUSize, 324
TlvPresent, 324
hsCallStatus
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
hsCallStatusValid
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
hsInd
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
hsIndValid
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
hwType
wds_DHCPv4HWConfig, 1515
WdsDHCPv4HWConfig, 1534
wdsDhcpv4HwConfig, 1534
hwVer
unpack_dms_GetHardwareRevision_t, 1117

iFaceTab
PCMparams, 787
iFaceTabLen
PCMparams, 787
iGetByteTotals
qaGobiApiWds.h, 2117
iGetConnectionRate
qaGobiApiWds.h, 2117
iGetPacketStatistics
qaGobiApiWds.h, 2117
iLTEbandValue
nas_PhyCaAggPcellInfo, 511
nas_PhyCaAggScellInfo, 516

NASPhyCaAggPcellInfo, 590
NASPhyCaAggScellInfo, 593
PhyCaAggPcellInfo, 795
PhyCaAggScellInfo, 800
IMCNflag
unpack_wds_SLQSGetRuntimeSettings_t, 1378
IMEIString
unpack_dms_GetDeviceSerialNumbers_t, 1112
IMG_ID_LEN
qaGobiApiFms.h, 1857
IMGDETAILS_LEN
qaGobiApiFms.h, 1857
IMS Service (IMS), 57
IMSAIndRegisterInfo, 328
pPdpStatusConfig, 328
pRatHandoverStatusConfig, 328
pRegStatusConfig, 329
pServiceStatusConfig, 329
IMSARegistrationStatus, 330
pImsRegErrCode, 331
pImsRegStatus, 331
pNewImsRegStatus, 331
IMSAServiceStatus, 331
pSmsServiceRat, 333
pSmsServiceStatus, 333
pUtServiceRat, 333
pUtServiceStatus, 333
pVoipServiceRat, 333
pVoipServiceStatus, 333
pVsServiceRat, 333
pVsServiceStatus, 333
pVtServiceRat, 333
pVtServiceStatus, 334
IMSASupportedFieldsResp, 334
pIndFieldsList, 334
pReqFieldsList, 334
pRespFieldsList, 334
IMSASupportedMsgInfo, 334
pSupportedMsgList, 335
IMSI_M_S1_LENGTH
qaGobiApiNas.h, 1896
IMSI_M_S2_LENGTH
qaGobiApiNas.h, 1896
IMSInfo
sMSOnIMSTlv, 983
IMSTlv
unpack_sms_SetNewSMSCallback_ind_t, 1279
INDEX_ZERO
qaNasPerformNetworkScan.h, 2143
INT32
SwiDataTypes.h, 2186
INT8
SwiDataTypes.h, 2186
INVALID_INSTACNE
qaGobiApiCbk.h, 1725
IOThresListLen
IOThresh, 345
nas_IOThresh, 482
IOThresh, 344
IOThresListLen, 345
pIOThresList, 345
IPAddress
DataStatusDetail, 222
nas_DataStatusDetail, 460
IPAddressV6
IPV6AddressInfo, 347
ipv6AddressInfo, 348
wds_IPV6AddressInfo, 1520
IPFamSupport
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1106
IPFamilyPreference
pack_wds_SLQSSetIPFamilyPreference_t, 779
unpack_wds_SLQSGetRuntimeSettings_t, 1378
IPSECSPI
LibPackTFTIDParams, 377
TFTIDParams, 1031
IPV4
qaGobiApiCbk.h, 1725
IPV4V6
qaGobiApiCbk.h, 1725
IPV6
qaGobiApiCbk.h, 1725
IPV6AddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1378
IPV6AddressInfo, 347
IPAddressV6, 347
IPV6PrefixLen, 347
IPV6GWAddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1378
IPV6GWAddressInfo, 348
gwAddressV6, 349
gwV6PrefixLen, 349
IPV6PrefixLen
IPV6AddressInfo, 347
ipv6AddressInfo, 348
wds_IPV6AddressInfo, 1520
IPv4
unpack_wds_SLQSGetRuntimeSettings_t, 1378
IPv4Addr, 345
addr, 345
IPv4Config, 346
loc_IPv4Config, 389
subnetMask, 345
wds_IPv4AdTlv, 1519
IPv4AddrTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1352
IPv4Config, 345
IPv4Addr, 346
IPv4Port, 346
IPv4DstAddr
unpack_qos_swiQosFilter_t, 1267
IPv4Info, 346
address, 346
port, 346
TlvPresent, 346

IPv4Port
IPv4Config, 346
loc_IPv4Config, 389
IPv4SrcAddr
unpack_qos_swiQosFilter_t, 1267
IPv4Tos
unpack_qos_swiQosFilter_t, 1267
IPv6Addr, 346
addr, 347
IPv6Config, 348
loc_IPv6Config, 390
prefixLen, 347
IPv6Config, 348
IPv6Addr, 348
IPv6Port, 348
IPv6DstAddr
unpack_qos_swiQosFilter_t, 1267
IPv6Info, 349
address, 349
port, 349
TlvPresent, 349
IPv6Label
unpack_qos_swiQosFilter_t, 1268
IPv6Port
IPv6Config, 348
loc_IPv6Config, 390
IPv6SrcAddr
unpack_qos_swiQosFilter_t, 1268
IPv6TrafCls, 349
mask, 350
unpack_qos_swiQosFilter_t, 1268
val, 350
iSLQSMISetIPFamilyPreference
qaGobiApiWds.h, 2117
iSLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1767
iSLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1767
iSLQSSetWdsFirstInstEventCallback
qaGobiApiCbk.h, 1767
iSLQSSetWdsSecondInstEventCallback
qaGobiApiCbk.h, 1767
iSLQSSetWdsThirdInstEventCallback
qaGobiApiCbk.h, 1767
iSLQSSetWdsXferStatsFirstInstCallback
qaGobiApiCbk.h, 1767
iSLQSSetWdsXferStatsSecondInstCallback
qaGobiApiCbk.h, 1767
iSetCATEventCallback
qaGobiApiCbk.h, 1767
iSetSignalStrengthCallback
qaGobiApiCbk.h, 1767
Id
unpack_qos_swiQosFilter_t, 1267
id
BdsSV, 142
CSGID, 200
loc_BdsSV, 383
loc_SV, 397
nas_CSGID, 456
pack_dms_UIMChangePIN_t, 643
pack_dms_UIMSetPINProtection_t, 645
pack_dms_UIMUnblockPIN_t, 646
pack_dms_UIMVerifyPIN_t, 647
QosFlowInfoState, 870
SV, 994
swiQosModifyReq, 1014
unpack_qos_QosFlowInfoState_t, 1259
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1265
id_length
custSettingInfo, 214
DMScustSettingInfo, 239
IdleState
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
IfaceID
qos_BindDataPortPeripheralEndPointID_t, 866
image_info_t, 324
buildID, 324
buildIDLen, 324
imageType, 324
uniqueID, 324
ImageElement, 325
buildId, 325
buildIdLength, 325
imageId, 325
imageType, 325
imageID
FMSImageIdElement, 263
ImageIdElement, 326
imageIDElement
FMSImageIDEntries, 264
ImageIDEntries, 327
ImageIDEntries, 326
executingImage, 327
imageIDElement, 327
imageIDSize, 327
imageType, 327
maxImages, 327
imageIDEntries
FMSImageList, 265
ImageList, 327
imageIDSize
FMSImageIDEntries, 264
ImageIDEntries, 327
imageId
FMSImageElement, 262
ImageElement, 325
ImageIdElement, 325
buildID, 326
buildIDLength, 326
failureCount, 326
imageID, 326
storageIndex, 326
ImageList, 327

imageIDEntries, 327
listSize, 328
imageList
unpack_fms_GetStoredImages_t, 1156
ImageListSize
unpack_fms_GetImagesPreference_t, 1156
imageListSize
pack_fms_SetImagesPreference_t, 649
imageType
CurrImageInfo, 206
FMSImageElement, 262
FMSImageIDEntries, 264
image_info_t, 324
ImageElement, 325
ImageIDEntries, 327
ImageTypes
unpack_fms_SetImagesPreference_t, 1157
ImageTypesSize
unpack_fms_SetImagesPreference_t, 1157
imagelistSize
unpack_fms_GetStoredImages_t, 1156
imei_no
unpack_dms_GetSerialNumbers_t, 1123
imeiSize
serialNumbersInfo, 904
unpack_dms_GetDeviceSerialNumbers_t, 1112
imeiSvnSize
serialNumbersInfo, 904
unpack_dms_GetDeviceSerialNumbers_t, 1112
ImeiSvnString
unpack_dms_GetDeviceSerialNumbers_t, 1112
imeisv_svn
unpack_dms_GetSerialNumbers_t, 1123
imgType
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1145
imsCfgIndRegisterInfo, 336
pRegMgrConfigEvents, 337
pSIPConfigEvents, 337
pSMSConfigEvents, 337
pUserConfigEvents, 337
pVoIPConfigEvents, 337
imsRegMgrConfigInfo, 337
pCSCFPortName, 338
pIMSTestMode, 338
pPriCSCFPort, 338
imsRegState
CommInfo, 191
nas_CommInfo, 456
imsSIPConfigInfo, 338
pSIPLocalPort, 339
pSigCompEnabled, 339
pSubscribeTimer, 339
pTimerSIPReg, 339
pTimerT1, 339
pTimerT2, 339
pTimerTf, 339
imsSMSConfigInfo, 339
pPhoneCtxtURI, 339
pSMSFormat, 339
pSMSOverIPNwInd, 339
imsUserConfigInfo, 340
pIMSDomain, 340
imsVoIPConfigInfo, 340
pAmrMode, 342
pAmrOctetAligned, 342
pAmrWBMode, 342
pAmrWBOctetAligned, 342
pAmrWbEnable, 342
pMinSessionExpiryTimer, 342
pRTPRTCPInactTimer, 342
pRingBackTimer, 342
pRingingTimer, 342
pScrAmrEnable, 342
pScrAmrWbEnable, 342
pSessionExpiryTimer, 342
imsaPdpStatusInfo, 329
connetionState, 329
pFailErrorCode, 329
imsaRatStatusInfo, 329
pErrorCodeStr, 330
pRATStatus, 330
pSrcRAT, 330
pTgtRAT, 330
imsaRegStatusInfo, 331
pImsRegStatus, 331
pRegStatusErrorCode, 331
pbIMSRegistered, 331
imsaSvcStatusInfo, 335
pSMSSvcRAT, 335
pSMSSvcStatus, 335
pUTSvcRAT, 335
pUTSvcStatus, 335
pVOIPSvcRAT, 335
pVOIPSvcStatus, 336
pVTSvcRAT, 336
pVTSvcStatus, 336
imsi
unpack_dms_GetIMSI_t, 1117
imsi_11_12
CDMASysInfoExt, 180
nas_CDMASysInfoExt, 452
imsiM1112
minBasedIMSI, 437
nas_minBasedIMSI, 501
imsiMS1
minBasedIMSI, 437
nas_minBasedIMSI, 501
imsiMS2
minBasedIMSI, 437
nas_minBasedIMSI, 501
imsiT1112
nas_trueIMSI, 554
trueIMSI, 1040
imsiTS1
nas_trueIMSI, 554

trueIMSI, 1040
imsiTS2
nas_trueIMSI, 554
trueIMSI, 1040
imsiTaddrNum
nas_trueIMSI, 554
trueIMSI, 1040
InUse
nas_QmiNas3GppNetworkInfo, 523
SlqsNas3GppNetworkInfo, 945
includes_pcs_digit
nas_QmisNasPcsDigit, 524
SlqsNasPcsDigit, 946
IndFieldsList, 342
indicationFields, 343
indicationFieldsLen, 343
index
pack_wds_GetMobileIPProfile_t, 763
pack_wds_SetDefaultProfileNum_t, 767
pack_wds_SetMobileIPProfile_t, 770
swiQosFilter, 1009
swiQosFlow, 1012
swiQosReq, 1015
unpack_qos_swiQosFilter_t, 1267
unpack_qos_swiQosFlow_t, 1271
unpack_wds_GetDefaultProfileNum_t, 1357
index1xPri
cardStatus, 158
uim_cardStatus, 1050
index1xSec
cardStatus, 158
uim_cardStatus, 1050
indexGwPri
cardStatus, 158
uim_cardStatus, 1050
indexGwSec
cardStatus, 158
uim_cardStatus, 1050
indicationFields
IndFieldsList, 343
indicationFieldsLen
IndFieldsList, 343
Info
unpack_nas_SLQSNasSwiOTAMessageCallback-_ind_t, 1229
unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind_t, 1230
infoInterFreq, 343
cell_resel_priority, 344
cellInterFreqParams, 344
cells_len, 344
earfcn, 344
threshXHigh, 344
threshXLow, 344
InfoInterfreq
LTEInfoInterfreq, 418
nas_LTEInfoInterfreq, 486
InitiateDomainAttach
qaGobiApiNas.h, 1918
InitiateNetworkRegistration
qaGobiApiNas.h, 1918
injectEnable
accelAcceptReady_s, 115
accelTempAcceptReady_s, 116
gyroAcceptReady_s, 312
gyroTempAcceptReady_s, 313
loc_accelAcceptReady, 381
loc_accelTempAcceptReady, 382
loc_gyroAcceptReady, 388
loc_gyroTempAcceptReady, 389
injectSensorDataStatus
QmiCbkLocInjectSensorDataInd, 843
unpack_loc_InjectSensorDataCallback_Ind_t, 1170
injectTimeSyncStatus
QmiCbkLocInjectTimeInd, 844
unpack_loc_InjectTimeSyncDataCallback_Ind_t, 1171
insNmrCellInfo
GERANInfo, 267
nas_GERANInfo, 468
instanceId
ssdatasession_params, 992
instancesSize
unpack_nas_GetRFInfo_t, 1191
interval
pack_pds_SetPDSDefaults_t, 703
pack_pds_SetXTRAAutomaticDownload_t, 705
pack_wds_SLQSSetWdsEventCallback_t, 780
TransferStatsDataType, 1038
Io
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
io
HDRSSInfo, 319
hdrSSInfo, 320
nas_SLQSSignalStrengthsInformation, 544
SLQSSignalStrengthsInformation, 956
ioDelta
nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
Iono_valid
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
ip
WdsIpAddressInfoReq, 1539
ipAddress
pack_wds_SetDefaultProfile_t, 767
ipFamily
_packetSrvStatus, 72
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1381
ipVersion
LibPackTFTIDParams, 377
TFTIDParams, 1031
ipaddr

unpack_wds_GetDefaultProfile_t, 1356
ipaddrv6
unpack_wds_GetDefaultProfile_t, 1356
ipfamily
ssdatasession_params, 992
ipv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1387
ipv4GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1387
ipv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1387
ipv6AddressInfo, 347
IPAddressV6, 348
IPV6PrefixLen, 348
ipv6GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1387
is856SysId
HDRSysInfo, 322
nas_HDRSysInfo, 479
is856SysIdValid
HDRSysInfo, 322
nas_HDRSysInfo, 479
is_DataRate_Available
unpack_qos_swiQosFlow_t, 1271
is_EspSpi_Available
unpack_qos_swiQosFilter_t, 1268
is_IPv4DstAddr_Available
unpack_qos_swiQosFilter_t, 1268
is_IPv4SrcAddr_Available
unpack_qos_swiQosFilter_t, 1268
is_IPv4Tos_Available
unpack_qos_swiQosFilter_t, 1268
is_IPv6DstAddr_Available
unpack_qos_swiQosFilter_t, 1268
is_IPv6Label_Available
unpack_qos_swiQosFilter_t, 1268
is_IPv6SrcAddr_Available
unpack_qos_swiQosFilter_t, 1268
is_IPv6TrafCls_Available
unpack_qos_swiQosFilter_t, 1268
is_Id_Available
unpack_qos_swiQosFilter_t, 1268
is_Jitter_Available
unpack_qos_swiQosFlow_t, 1272
is_Latency_Available
unpack_qos_swiQosFlow_t, 1272
is_LteBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1139
is_LteQci_Available
unpack_qos_swiQosFlow_t, 1272
is_MaxAllowedPktSz_Available
unpack_qos_swiQosFlow_t, 1272
is_MinPolicedPktSz_Available
unpack_qos_swiQosFlow_t, 1272
is_NxtHdrProto_Available
unpack_qos_swiQosFilter_t, 1268
is_PktErrRate_Available
unpack_qos_swiQosFlow_t, 1272
is_Precedence_Available
unpack_qos_swiQosFilter_t, 1268
is_ProfileId3GPP2_Available
unpack_qos_swiQosFlow_t, 1272
is_RxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1258
is_TCPDstPort_Available
unpack_qos_swiQosFilter_t, 1268
is_TCPSrcPort_Available
unpack_qos_swiQosFilter_t, 1268
is_TdsBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1139
is_TokenBucket_Available
unpack_qos_swiQosFlow_t, 1272
is_TrafficClass_Available
unpack_qos_swiQosFlow_t, 1272
is_TranDstPort_Available
unpack_qos_swiQosFilter_t, 1268
is_TranSrcPort_Available
unpack_qos_swiQosFilter_t, 1268
is_TxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1258
is_UDPDstPort_Available
unpack_qos_swiQosFilter_t, 1268
is_UDPSrcPort_Available
unpack_qos_swiQosFilter_t, 1268
is_val_3GPP2Pri_Available
unpack_qos_swiQosFlow_t, 1272
is_val_3GPPImCn_Available
unpack_qos_swiQosFlow_t, 1272
is_val_3GPPResResidualBER_Available
unpack_qos_swiQosFlow_t, 1272
is_val_3GPPSigInd_Available
unpack_qos_swiQosFlow_t, 1272
is_val_3GPPTraHdlPri_Available
unpack_qos_swiQosFlow_t, 1272
isEmpty
getAllCallInformation, 269
voice_getAllCallInformation, 1430
isInTraffic
nas_txInfo, 554
txInfo, 1042
isModByCC
SUPSInfo, 993
voice_SUPSInfo, 1443
isNewFlow
QosFlowInfoState, 870
unpack_qos_QosFlowInfoState_t, 1259
isPrefDataPath
GSMSrvStatusInfo, 309
nas_GSMSrvStatusInfo, 471
nas_SrvStatusInfo, 546
SrvStatusInfo, 989
isRadioTuned

nas_rxInfo, 536
nas_RxSigInfo, 537
rxInfo, 891
RxSigInfo, 894
isSysForbidden
detailSvcInfo, 226
nas_detailSvcInfo, 462
nas_sysInfoCommon, 548
sysInfoCommon, 1018
isSysForbiddenValid
nas_sysInfoCommon, 548
sysInfoCommon, 1018
isSysPrlMatch
CDMASysInfo, 179
HDRSysInfo, 322
nas_CDMASysInfo, 451
nas_HDRSysInfo, 479
isSysPrlMatchValid
CDMASysInfo, 179
HDRSysInfo, 322
nas_CDMASysInfo, 451
nas_HDRSysInfo, 479
IsVoiceEnabled
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1106
Item
GetAudioPathConfigReq, 271
GetAudioVolTLBConfigReq, 275
SetAudioVolTLBConfigReq, 914

Jitter
unpack_qos_swiQosFlow_t, 1272

KeyExchange
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823

LITE_VOICE_SUPS_SRV_CLASS_DATA
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITAS-
YNC
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSY-
NC
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_FAX
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_NONE
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_PADACCESS
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_SMS
voice.h, 2225
LITE_VOICE_SUPS_SRV_CLASS_VOICE
voice.h, 2225
LITEQMI_QMI_CBK_PARAM_NOCHANGE
sms.h, 2167
LITEQMI_QMI_CBK_PARAM_RESET
sms.h, 2167
LITEQMI_QMI_CBK_PARAM_SET
sms.h, 2167
LBPTlv
NASQmiCbkNasSystemSelPrefInd, 599
LEN
qaGobiApiDcs.h, 1805
LMCERTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t,
1368
LOCEVENTMASKGNSSSVINFO
loc.h, 1618
LOCEVENTMASKNMEA
loc.h, 1619
LOCEVENTMASKWIFIREQ
loc.h, 1619
LOCEventRegisterReqResp, 401
eventRegister, 403
LOCExtPowerStateReqResp, 403
extPowerState, 403
LOCStartReq, 411
pApplicationInfo, 412
pConfigAltitudeAssumed, 412
pHorizontalAccuracyLvl, 412
pIntermediateReportState, 412
pMinIntervalTime, 412
pRecurrenceType, 412
SessionId, 412
LOCStopReq, 413
sessionId, 413
LPCSTR
SwiDataTypes.h, 2186
LTEAttachProfile
unpack_wds_SLQSGet3GPPConfigItem_t, 1369
LTEAttachProfileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1369
LTEAttachProfileListLen
_slqs3GPPConfigItem, 85
pack_wds_SLQSSet3GPPConfigItem_t, 778
unpack_wds_SLQSGet3GPPConfigItem_t, 1369
LTEBandPref
NASLTEBandPreferenceTlv, 583
LTECphyCaInfo
unpack_nas_SlqsGetLTECphyCAInfo_t, 1201
LTEInfo, 416
band, 417
bandwidth, 417
emmConnState, 418
emmState, 418
emmSubState, 418
RXChan, 418
TXChan, 418
LTEInfoInterfreq, 418
freqsLen, 418
InfoInterfreq, 418
ueInIdle, 418
LTEInfoIntrafreq, 418

CellParams, 420
cellReselPriority, 420
cellsLen, 420
earfcn, 420
globalCellId, 420
plmn, 420
sIntraSearch, 420
sNonIntraSearch, 420
servingCellId, 420
tac, 420
threshServingLow, 420
ueInIdle, 420
LTEInfoNeighboringGSM, 420
freqsLen, 421
LteGsmCellInfo, 421
ueInIdle, 421
LTEInfoNeighboringWCDMA, 421
freqsLen, 422
LTEWCDMACellInfo, 422
ueInIdle, 422
LTEOperationMode, 422
pLTEOperationMode, 423
TlvPresent, 423
LTERSRPThresh, 424
LTERSRPThreshListLen, 424
pLTERSRPThreshList, 424
LTERSRPThreshListLen
LTERSRPThresh, 424
nas_LTERSRPThresh, 492
LTERSRQThresh, 424
LTERSRQThreshListLen, 425
pLTERSRQThreshList, 425
LTERSRQThreshListLen
LTERSRQThresh, 425
nas_LTERSRQThresh, 492
LTERSSIThresh, 425
LTERSSIThreshListLen, 425
pLTERSSIThreshList, 425
LTERSSIThreshListLen
LTERSSIThresh, 425
nas_LTERSSIThresh, 493
LTESNRThresListLen
LTESNRThresh, 429
nas_LTESNRThresh, 496
LTESNRThresh, 428
LTESNRThresListLen, 429
pLTESNRThresList, 429
LTESNRThreshListLen
LTESNRThreshold, 429
nas_LTESNRThreshold, 496
LTESNRThreshold, 429
LTESNRThreshListLen, 429
pLTESNRThreshList, 429
LTESSInfo, 430
rsrp, 430
rsrq, 430
rssi, 430
snr, 430
unpack_nas_SLQSNasGetSigInfo_t, 1225
LTESigRptCfg, 426
avgPeriod, 427
rptRate, 427
LTESigRptConfig, 427
avgPeriod, 428
rptRate, 428
LTESysInfo, 431
cellId, 433
cellIdValid, 433
lac, 433
lacValid, 433
MCC, 433
MNC, 433
networkIdValid, 433
regRejectInfoValid, 433
rejCause, 433
rejectSrvDomain, 433
sysInfoLTE, 433
tac, 433
tacValid, 434
LTEWCDMACellInfo
LTEInfoNeighboringWCDMA, 422
nas_LTEInfoNeighboringWCDMA, 490
Lac
qaQmiServingSystemParam, 833
unpack_nas_SLQSGetServingSystem_t, 1208
lac
GERANInfo, 268
GSMSysInfo, 311
LTESysInfo, 433
nas_GERANInfo, 468
nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_UMTSInfo, 556
nas_WCDMASysInfo, 565
UMTSInfo, 1091
WCDMASysInfo, 1506
lac1
nas_OperatorPLMNData, 510
OperatorPLMNData, 629
lac2
nas_OperatorPLMNData, 510
OperatorPLMNData, 629
lacValid
GSMSysInfo, 312
LTESysInfo, 433
nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
language
CDMABroadcastConfig, 164
sms_CDMABroadcastConfig, 959
lastCallDataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1375
lastCallDataBearerTechnology

unpack_wds_SLQSGetDataBearerTechnology_t, 1372
lastCallRXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1375
lastCallTXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1375
LastErrCode
DataStatusDetail, 222
nas_DataStatusDetail, 460
Latency
unpack_qos_swiQosFlow_t, 1272
Latitude
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
latitude
pack_loc_SLQSLOCInjectPosition_t, 659
leapSeconds
nas_qaQmi3Gpp2TimeZone, 522
qaQmi3Gpp2TimeZone, 828
leaseState
wds_DHCPLeaseStateTlv, 1513
len
BdsSVInfo, 143
loc_BdsSVInfo, 383
loc_SVInfo, 397
SVInfo, 995
t_Sv, 1023
unpack_nas_GetSignalStrengths_t, 1194
length
readTransparentInfo, 872
SMSCAddress, 973
sMSCAddressInfo, 973
SMSEtwsMessage, 974
sMSEtwsMessageInfo, 975
SMSTransferRouteMTMessage, 986
sMSTransferRouteMTMessageInfo, 986
uim_readTransparentInfo, 1059
Level
GetM2MAudioVolumeResp, 292
SetM2MAudioVolumeReq, 924
LibPackGPRSRequestedQoS, 350
delayClass, 351
meanThroughputClass, 351
peakThroughputClass, 351
precedenceClass, 351
reliabilityClass, 351
LibPackQosClassID, 374
gDlBitRate, 375
gUlBitRate, 375
maxDlBitRate, 375
maxUlBitRate, 375
QCI, 375
LibPackTFTIDParams, 375
destPortRangeEnd, 376
destPortRangeStart, 376
eValid, 376
filterId, 377
flowLabel, 377
IPSECSPI, 377
ipVersion, 377
nextHeader, 377
pSourceIP, 377
sourceIPMask, 377
srcPortRangeEnd, 377
srcPortRangeStart, 377
tosMask, 377
LibPackUMTSQoS, 377
deliveryErrSDU, 379
grntDownlinkBitrate, 379
grntUplinkBitrate, 379
maxDownlinkBitrate, 379
maxSDUSize, 379
maxUplinkBitrate, 379
qosDeliveryOrder, 379
resBerRatio, 379
sduErrorRatio, 379
trafficClass, 379
trafficPriority, 379
transferDelay, 379
LibPackUMTSReqQoSSigInd, 380
SigInd, 380
UMTSReqQoS, 380
LibPackprofile_3GPP, 363
pAPNClass, 368
pAPNDisabledFlag, 368
pAPNName, 368
pAPNnameSize, 368
pAddrAllocPref, 367
pAuthenticationPref, 368
pGPRSMinimumQoS, 368
pGPRSRequestedQos, 368
pIPv4AddrPref, 368
pIPv6AddPref, 368
pImCnFlag, 368
pPDNInactivTimeout, 368
pPDPtype, 368
pPassword, 368
pPasswordSize, 368
pPcscfAddrUsingDhcp, 368
pPcscfAddrUsingPCO, 368
pPdpAccessConFlag, 368
pPdpContext, 368
pPdpDataCompType, 368
pPdpHdrCompType, 368
pPriDNSIPv4AddPref, 368
pPriDNSIPv6addpref, 368
pPrimaryID, 368
pProfilename, 368
pProfilenameSize, 368
pQosClassID, 368
pSecDNSIPv4AddPref, 368
pSecDNSIPv6addpref, 368
pSecondaryFlag, 369
pSupportEmergencyCalls, 369
pTFTID1Params, 369
pTFTID2Params, 369

pUMTSMinQoS, 369
pUMTSMinQosSigInd, 369
pUMTSReqQoS, 369
pUMTSReqQoSSigInd, 369
pUsername, 369
pUsernameSize, 369
LibPackprofile_3GPP2, 369
pAPNClass3GPP2, 373
pAPNEnabled3GPP2, 373
pAllowLinger, 373
pApnString, 373
pApnStringSize, 373
pAppPriority, 373
pAppType, 373
pAuthPassword, 373
pAuthPassword_tSize, 373
pAuthProtocol, 373
pAuthRetryCount, 373
pAuthTimeout, 373
pDataMode, 373
pDataRate, 373
pIpcpAckTimeout, 373
pIpcpCreqRetryCount, 374
pIsPcscfAddressNedded, 374
pLcpAckTimeout, 374
pLcpCreqRetryCount, 374
pNegoDnsSrvrPref, 374
pPDNInactivTimeout3GPP2, 374
pPdnType, 374
pPppSessCloseTimer1x, 374
pPppSessCloseTimerDO, 374
pPriV6DnsAddress, 374
pPrimaryV4DnsAddress, 374
pRATType, 374
pSecV6DnsAddress, 374
pSecondaryV4DnsAddress, 374
pUserId, 374
pUserIdSize, 374
LibpackProfile3GPP, 351
pAPNClass, 356
pAPNDisabledFlag, 356
pAPNName, 356
pAPNnameSize, 356
pAddrAllocPref, 356
pAuthenticationPref, 356
pGPRSMinimumQoS, 356
pGPRSRequestedQos, 356
pIPv4AddrPref, 357
pIPv6AddPref, 357
pImCnFlag, 357
pPDNInactivTimeout, 357
pPDPtype, 357
pPassword, 357
pPasswordSize, 357
pPcscfAddrUsingDhcp, 357
pPcscfAddrUsingPCO, 357
pPdpAccessConFlag, 357
pPdpContext, 357
pPdpDataCompType, 357
pPdpHdrCompType, 357
pPriDNSIPv4AddPref, 357
pPriDNSIPv6addpref, 357
pPrimaryID, 357
pProfilename, 357
pProfilenameSize, 357
pQosClassID, 357
pSecDNSIPv4AddPref, 357
pSecDNSIPv6addpref, 357
pSecondaryFlag, 357
pSupportEmergencyCalls, 357
pTFTID1Params, 357
pTFTID2Params, 357
pUMTSMinQoS, 357
pUMTSMinQosSigInd, 357
pUMTSReqQoS, 357
pUMTSReqQoSSigInd, 358
pUsername, 358
pUsernameSize, 358
LibpackProfile3GPP2, 358
pAPNClass3GPP2, 362
pAPNEnabled3GPP2, 362
pAllowLinger, 362
pApnString, 362
pApnStringSize, 362
pAppPriority, 362
pAppType, 362
pAuthPassword, 362
pAuthPasswordSize, 362
pAuthProtocol, 362
pAuthRetryCount, 362
pAuthTimeout, 362
pDataMode, 362
pDataRate, 362
pIpcpAckTimeout, 362
pIpcpCreqRetryCount, 362
pIsPcscfAddressNedded, 362
pLcpAckTimeout, 362
pLcpCreqRetryCount, 362
pNegoDnsSrvrPref, 363
pPDNInactivTimeout3GPP2, 363
pPdnType, 363
pPppSessCloseTimer1x, 363
pPppSessCloseTimerDO, 363
pPriV6DnsAddress, 363
pPrimaryV4DnsAddress, 363
pRATType, 363
pSecV6DnsAddress, 363
pSecondaryV4DnsAddress, 363
pUserId, 363
pUserIdSize, 363
lineCtrlInfo, 380
polarityIncluded, 381
pwrDenialTime, 381
revPolarity, 381
toggleMode, 381
lineValue

pack_voice_SLQSVoiceALSSelectLine_t, 743
voiceALSSelectLineInfo, 1446
linkage
altitudeSrcInfo, 122
altSrcInfo_t, 123
list_type
custSettingList, 214
DMScustSettingList, 240
DMSgetCustomInput, 241
getCustomInput, 278
pack_dms_GetCustFeaturesV2_t, 631
listEntries
FMSPrefImageList, 265
PrefImageList, 808
listSize
FMSImageList, 265
FMSPrefImageList, 265
ImageList, 328
PrefImageList, 808
liteQmiDataBearerMasks
wds.h, 2257
liteServiceClassInformation
voice.h, 2225
liteqmi_GetVersion
common.h, 1552
liteqmi_helper_decode7bitAsciiEncString
common.h, 1552
liteqmi_log
common.h, 1552
loc.h
eQMI_LOC_SESS_STATUS_FAILURE, 1620
eQMI_LOC_SESS_STATUS_IN_PROGRESS, 1620
eQMI_LOC_SESS_STATUS_SUCCESS, 1620
eQMI_LOC_SESS_STATUS_TIMEOUT, 1620
loc.h, 1614
LOCEVENTMASKNMEA, 1619
LOCEVENTMASKWIFIREQ, 1619
MAX_TEMP_DATA_LEN, 1619
pack_loc_DeleteAssistData, 1620
pack_loc_EventRegister, 1620
pack_loc_SLQSLOCGetBestAvailPos, 1621
pack_loc_SLQSLOCGetServer, 1622
pack_loc_SLQSLOCInjectPosition, 1622
pack_loc_SLQSLOCInjectSensorData, 1622
pack_loc_SLQSLOCInjectUTCTime, 1623
pack_loc_SLQSLOCSetCradleMountConfig, 1623
pack_loc_SLQSLOCSetServer, 1624
pack_loc_SetExtPowerState, 1620
pack_loc_SetOperationMode, 1621
pack_loc_Start, 1624
pack_loc_Stop, 1624
unpack_loc_BestAvailPos_Ind, 1625
unpack_loc_CradleMountCallback_Ind, 1625
unpack_loc_DeleteAssistData, 1626
unpack_loc_DeleteAssistData_Ind, 1626
unpack_loc_EngineState_Ind, 1626
unpack_loc_EventNMEA_Ind, 1627
unpack_loc_EventRegister, 1627
unpack_loc_EventTimeSyncCallback_Ind, 1627
unpack_loc_GetServer_Ind, 1628
unpack_loc_GnssSvInfo_Ind, 1628
unpack_loc_InjectPositionCallback_Ind, 1628
unpack_loc_InjectSensorDataCallback_Ind, 1629
unpack_loc_InjectTimeSyncDataCallback_Ind, 1629
unpack_loc_InjectUTCTimeCallback_Ind, 1629
unpack_loc_PositionRpt_Ind, 1630
unpack_loc_SLQSLOCGetBestAvailPos, 1632
unpack_loc_SLQSLOCGetServer, 1632
unpack_loc_SLQSLOCInjectPosition, 1633
unpack_loc_SLQSLOCInjectSensorData, 1633
unpack_loc_SLQSLOCInjectUTCTime, 1633
unpack_loc_SLQSLOCSetCradleMountConfig, 1634
unpack_loc_SLQSLOCSetServer, 1634
unpack_loc_SensorStreamingCallback_Ind, 1630
unpack_loc_SetExtPowerConfig_Ind, 1630
unpack_loc_SetExtPowerState, 1631
unpack_loc_SetOperationMode, 1631
unpack_loc_SetOperationMode_Ind, 1631
unpack_loc_SetServer_Ind, 1632
unpack_loc_Start, 1634
unpack_loc_Stop, 1635
loc_BdsSV, 382
id, 383
mask, 383
loc_BdsSVInfo, 383
len, 383
pSV, 383
loc_CellDb, 383
mask, 384
loc_ClkInfo, 384
mask, 385
loc_GnssData, 385
mask, 386
loc_IPv4Config, 389
IPv4Addr, 389
IPv4Port, 389
loc_IPv4Info, 389
address, 390
port, 390
TlvPresent, 390
loc_IPv6Config, 390
IPv6Addr, 390
IPv6Port, 390
loc_IPv6Info, 390
address, 391
port, 391
TlvPresent, 391
loc_LocApplicationInfo, 391
appNameLength, 392
appProviderLength, 392
appVersionLength, 392
appVersionValid, 392
pAppName, 392

pAppProvider, 392
pAppVersion, 392
loc_SV, 396
id, 397
mask, 397
system, 397
loc_SVInfo, 397
len, 397
pSV, 397
loc_URLAddrInfo, 398
urlAddr, 399
loc_accelAcceptReady, 381
batchPerSec, 381
injectEnable, 381
samplesPerBatch, 382
loc_accelTempAcceptReady, 382
batchPerSec, 382
injectEnable, 382
samplesPerBatch, 382
loc_gpsTime, 387
gpsTimeOfWeekMs, 387
gpsWeek, 387
loc_gyroAcceptReady, 387
batchPerSec, 388
injectEnable, 388
samplesPerBatch, 388
loc_gyroTempAcceptReady, 388
batchPerSec, 389
injectEnable, 389
samplesPerBatch, 389
loc_precisionDilution, 392
HDOP, 393
PDOP, 393
VDOP, 393
loc_satelliteInfo, 393
azimuth, 395
elevation, 395
gnssSvId, 395
healthStatus, 395
snr, 395
svInfoMask, 395
svListLen, 395
svStatus, 395
system, 395
validMask, 395
loc_sensorDataUsage, 395
aidingIndicatorMask, 396
usageMask, 396
loc_svUsedforFix, 397
gnssSvUsedList, 398
gnssSvUsedList_len, 398
loc_urlAddr, 398
address, 398
TlvPresent, 398
LocApplicationInfo, 399
appNameLength, 400
appProviderLength, 400
appVersionLength, 400
appVersionValid, 400
pAppName, 400
pAppProvider, 400
pAppVersion, 400
LocDelAssDataReq, 400
pBdsSVInfo, 401
pCellDb, 401
pClkInfo, 401
pGnssData, 401
pSVInfo, 401
LocGetServerReq, 403
pServerAddrTypeMask, 404
serverType, 404
LocInjectPositionReq, 404
pAltitudeSrcInfo, 407
pAltitudeWrtEllipsoid, 407
pAltitudeWrtMeanSeaLevel, 407
pHorConfidence, 407
pHorReliability, 408
pHorUncCircular, 408
pLatitude, 408
pLongitude, 408
pPositionSrc, 408
pRawHorConfidence, 408
pRawHorUncCircular, 408
pTimestampAge, 408
pTimestampUtc, 408
pVertConfidence, 408
pVertReliability, 408
pVertUnc, 408
LocInjectSensorDataReq, 408
pAcceleroData, 409
pAcceleroTempData, 409
pAcceleroTimeSrc, 409
pGyroData, 409
pGyroTempData, 409
pGyroTimeSrc, 409
pOpaqueIdentifier, 409
LocSetCradleMountReq, 409
pConfidence, 410
state, 410
LocSetServerReq, 410
pIPv4Config, 411
pIPv6Config, 411
pURLAddr, 411
serverType, 411
localTimeOffset
nas_qaQmi3Gpp2TimeZone, 522
qaQmi3Gpp2TimeZone, 828
Location Service(LOC), 59
logString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1145
logger
common.h, 1550
longName
nas_PLMNNetworkNameData, 519
nasPLMNNameResp, 596

PLMNNetworkNameData, 804
unpack_nas_SLQSGetPLMNName_t, 1204
longNameCI
nasPLMNNameResp, 596
unpack_nas_SLQSGetPLMNName_t, 1204
longNameEn
nasPLMNNameResp, 596
unpack_nas_SLQSGetPLMNName_t, 1204
longNameLen
nas_PLMNNetworkNameData, 519
nasPLMNNameResp, 596
PLMNNetworkNameData, 804
unpack_nas_SLQSGetPLMNName_t, 1204
longNameSB
nasPLMNNameResp, 596
unpack_nas_SLQSGetPLMNName_t, 1204
longNameSpareBits
nas_PLMNNetworkNameData, 519
PLMNNetworkNameData, 804
Longitude
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
longitude
pack_loc_SLQSLOCInjectPosition_t, 659
loopbackMode
pack_wds_SLQSSSetLoopback_t, 781
loopbackMultiplier
pack_wds_SLQSSSetLoopback_t, 781
LpmFlag
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
LteBandCapability
unpack_dms_SLQSGetBandCapability_t, 1139
LteCQIParm, 413
CQIValueCW0, 414
CQIValueCW1, 414
ValidityCW0, 414
ValidityCW1, 414
lteEARFCN, 414
earfcn0, 414
earfcn1, 414
status, 414
lteEmmDl
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
lteEmmUl
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
lteEsmDl
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
lteEsmUl
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
LteGsmCellInfo
LTEInfoNeighboringGSM, 421
nas_LTEInfoNeighboringGSM, 489
lteGsmCellInfo, 414
cellReselPriority, 415
cells_len, 415
GsmCellInfo, 415
nccPermitted, 415
threshGsmHigh, 415
threshGsmLow, 416
LteNasReleaseInfo
qaGobiApiCbk.h, 1729
LteNasReleaseInfo_s, 422
nas_major, 422
nas_minor, 422
nas_release, 422
ltePCI, 423
earfcn, 423
pci, 423
status, 423
LteQci
unpack_qos_swiQosFlow_t, 1272
lteRsrpDelta
nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
lteRsrpinfo
nas_SLQSSignalStrengthsInformation, 544
SLQSSignalStrengthsInformation, 956
lteRsrpinformation, 424
rsrplevel, 424
lteSSInfo, 430
rsrp, 431
rsrq, 431
rssi, 431
snr, 431
LteSccRxInfoResp, 426
pSccRxInfo, 426
lteSnrDelta
nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
lteSnrinfo
nas_SLQSSignalStrengthsInformation, 545
SLQSSignalStrengthsInformation, 956
lteSnrinformation, 428
snrlevel, 428
lteWcdmaCellInfo, 434
cellReselPriority, 434
cellsLen, 434
threshXhigh, 434
threshXlow, 435
uarfcn, 435
WCDMACellInfo, 435
ltersrp
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
ltesnr
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211

m_FwBuildId
CarrierImage_t, 159
SWI_STRUCT_CarrierImage, 996
m_FwImageId
CarrierImage_t, 159

SWI_STRUCT_CarrierImage, 996
m_PriBuildId
CarrierImage_t, 159
SWI_STRUCT_CarrierImage, 996
m_PriImageId
CarrierImage_t, 159
SWI_STRUCT_CarrierImage, 997
m_nCarrierId
CarrierImage_t, 159
SWI_STRUCT_CarrierImage, 996
m_nFolderId
CarrierImage_t, 159
SWI_STRUCT_CarrierImage, 996
m_nStorage
CarrierImage_t, 159
SWI_STRUCT_CarrierImage, 996
MACIndex
nas_NetworkStatEVDO, 507
NetworkStatEVDO, 616
MAX_BUILD_ID_LEN
dms.h, 1561
MAX_CALL_NO_LEN
qaGobiApiVoice.h, 2080
MAX_CONTENT_LENGTH
qaGobiApiUim.h, 2067
MAX_CUST_ID_LEN
qaGobiApiDms.h, 1816
MAX_FSN_LENGTH
qaGobiApiDms.h, 1816
MAX_ICCID_LENGTH
qaGobiApiUim.h, 2067
uim.h, 2204
MAX_MSE_TWS_MSG
sms.h, 2167
MAX_NMEA_STR_LEN
qaGobiApiCbk.h, 1726
MAX_NO_OF_CALLS
qaGobiApiCbk.h, 1726
qaGobiApiVoice.h, 2080
MAX_NO_OF_FILES
qaGobiApiCbk.h, 1726
MAX_NO_OF_SLOTS
qaGobiApiCbk.h, 1726
qaGobiApiUim.h, 2067
uim.h, 2204
MAX_NO_OF_UUSINFO
qaGobiApiCbk.h, 1726
MAX_PATH_LENGTH
qaGobiApiCbk.h, 1726
qaGobiApiUim.h, 2067
MAX_PILOT_SETS
qaGobiApiNas.h, 1896
MAX_PUK_LENGTH
qaGobiApiUim.h, 2067
MAX_SENSOR_DATA_LEN
loc.h, 1619
MAX_SLOTS_STATUS
qaGobiApiUim.h, 2067
uim.h, 2204
MAX_SMS_LIST_SIZE
sms.h, 2167
MAX_SMS_ROUTES
qaGobiApiSms.h, 1971
MAX_TEMP_DATA_LEN
loc.h, 1619
qaGobiApiLoc.h, 1884
MAXUSSDLENGTH
qaGobiApiCbk.h, 1726
qaGobiApiVoice.h, 2080
MAXVOICEUSSDLENGTH
voice.h, 2225
MCC
_SlqsNas3GppNetworkRAT_, 89
CDMASysInfo, 179
CDMASysInfoExt, 180
currentPLMN, 205
GSMSysInfo, 312
LTESysInfo, 433
nas_CDMASysInfo, 451
nas_CDMASysInfoExt, 452
nas_currentPLMN, 457
nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_QmiNas3GppNetworkInfo, 523
nas_QmiNas3GppNetworkRAT, 523
nas_QmisNasPcsDigit, 524
nas_WCDMASysInfo, 565
SlqsNas3GppNetworkInfo, 945
SlqsNasPcsDigit, 946
unpack_nas_GetServingNetwork_t, 1193
WCDMASysInfo, 1506
MDMCallDuration
ConnectionStatus, 192
connectionStatus, 192
MDMConnStatus
ConnectionStatus, 192
connectionStatus, 193
wds_ConnStatusTlv, 1508
MEID_MAX_SIZE
dms.h, 1561
MEIDString
unpack_dms_GetDeviceSerialNumbers_t, 1112
MIN
unpack_dms_GetVoiceNumber_t, 1126
MINREQBKLEN
common.h, 1549
MMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1279
MNC
_SlqsNas3GppNetworkRAT_, 89
CDMASysInfo, 179
currentPLMN, 205
GSMSysInfo, 312
LTESysInfo, 433
nas_CDMASysInfo, 451
nas_currentPLMN, 457

nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_QmiNas3GppNetworkInfo, 523
nas_QmiNas3GppNetworkRAT, 524
nas_QmisNasPcsDigit, 524
nas_WCDMASysInfo, 565
SlqsNas3GppNetworkInfo, 945
SlqsNasPcsDigit, 946
unpack_nas_GetServingNetwork_t, 1193
WCDMASysInfo, 1506
MNRInfo, 437
mcc, 438
mnc, 438
rat, 438
MPTlv
NASQmiCbkNasSystemSelPrefInd, 599
MSGID_AND_LEN
common.h, 1549
MSGID_DONT_CARE
common.h, 1549
MTMessageInfo
newMTMessageTlv, 617
MTUSize
MTUSizeReq, 441
pack_swidms_SLQSSwiDmsSetMTU_t, 725
MTUSize3gpp
mtuSize3gppTlv, 440
swidms_mtuSize3gppTlv, 998
MTUSizeReq, 440
MTUSize, 441
manString
pack_dms_SLQSSwiSetHostDevInfo_t, 641
unpack_dms_SLQSSwiGetHostDevInfo_t, 1146
manufacturer
unpack_dms_GetManufacturer_t, 1118
Mask
getDUNCallInfoReq, 279
pack_wds_SLQSGetDUNCallInfo_t, 774
mask
BdsSV, 142
CellDb, 180
ClkInfo, 187
GnssData, 299
IPv6TrafCls, 350
loc_BdsSV, 383
loc_CellDb, 384
loc_ClkInfo, 385
loc_GnssData, 386
loc_SV, 397
SV, 994
Tos, 1037
unpack_qos_IPv6TrafCls_t, 1256
unpack_qos_Tos_t, 1274
max
_SLQSAVMSSettingsPeriodsInfo, 89
max_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1371
WDSSWICurrentChannelRates, 1546
max_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1371
WDSSWICurrentChannelRates, 1546
max_dist
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocGetAutoStartResp, 1001
SwiLocSetAutoStartReq, 1003
unpack_swiloc_SwiLocGetAutoStart_t, 1292
max_dist_reported
SwiLocGetAutoStartResp, 1001
unpack_swiloc_SwiLocGetAutoStart_t, 1292
max_time
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocGetAutoStartResp, 1001
SwiLocSetAutoStartReq, 1003
unpack_swiloc_SwiLocGetAutoStart_t, 1292
max_time_reported
SwiLocGetAutoStartResp, 1001
unpack_swiloc_SwiLocGetAutoStart_t, 1292
MaxAllowedPktSz
unpack_qos_swiQosFlow_t, 1272
MaxChanRxRate
ChannelRate, 183
dunchannelRate, 249
MaxChanTxRate
ChannelRate, 183
dunchannelRate, 249
maxChannelRXRate
unpack_wds_GetConnectionRate_t, 1353
maxChannelTXRate
unpack_wds_GetConnectionRate_t, 1353
maxDlBitRate
LibPackQosClassID, 375
QosClassID, 867
maxDownlinkBitrate
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
maxImages
FMSImageIDEntries, 264
ImageIDEntries, 327
maxMitigationLevel
mitigationDevList, 437
MaxRXChannelRate
unpack_dms_GetDeviceCap_t, 1108
maxRxChannelRate
unpack_dms_GetDeviceCapabilities_t, 1109
maxSDUSize
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
maxStorageSize
sms_maxStorageSizeResp, 963
smsMaxStorageSizeResp, 979

MaxTXChannelRate
unpack_dms_GetDeviceCap_t, 1108
maxTxChannelRate
unpack_dms_GetDeviceCapabilities_t, 1109
maxUlBitRate
LibPackQosClassID, 375
QosClassID, 867
maxUplinkBitrate
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
mcTimeStamp
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 171
mcc
CSGID, 200
MNRInfo, 438
nas_CSGID, 457
nas_MNRInfo, 501
nas_netSelectionPref, 503
nas_OperatorPLMNData, 510
nasPLMNNameReq, 594
netSelectionPref, 610
OperatorPLMNData, 629
pack_nas_SLQSGetPLMNName_t, 671
PlmnID, 802
unpack_nas_GetHomeNetwork_t, 1189
mccM
minBasedIMSI, 437
nas_minBasedIMSI, 501
mccT
nas_trueIMSI, 554
trueIMSI, 1040
mdmCallDurationActive
unpack_wds_SLQSGetDUNCallInfo_t, 1375
MdmConnStatus
DUNCallInfoInd, 248
meanThroughputClass
GPRSQoS, 301
GPRSRequestedQoS, 302
LibPackGPRSRequestedQoS, 351
wds_GPRSQoS, 1519
meid
unpack_dms_GetSerialNumbers_t, 1123
meidLength
_SLQSSwiGetSerialNoExtParams, 99
meidSize
serialNumbersInfo, 904
unpack_dms_GetDeviceSerialNumbers_t, 1112
meidString
unpack_dms_SLQSSwiGetSerialNoExt_t, 1148
message
unpack_sms_SLQSGetSMS_t, 1281
message_type
NASOTAMessageTlv, 590
messageClass
sms_routeEntry, 966
smsRouteEntry, 984
messageFailureCode
slqssendsmsparams_s, 949
unpack_sms_SendSMS_t, 1278
messageFormat
pack_sms_SaveSMS_t, 716
pack_sms_SendSMS_t, 716
slqssendasyncsmsparams_s, 948
slqssendsmsparams_s, 949
sms_sendAsyncsmsParams, 968
unpack_sms_SLQSGetSMS_t, 1281
messageID
slqssendsmsparams_s, 949
SMSAsyncRawSend_s, 972
unpack_sms_SendSMS_t, 1278
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
messageId
cdmaMsgEncodingParams, 174
messageIndex
pack_sms_SLQSGetSMS_t, 719
pack_sms_SLQSModifySMSStatus_t, 721
qmiSmsMessageList, 859
SMSMTMessage, 981
sMSMTMessageInfo, 982
messageLength
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 171
messageList
unpack_sms_SLQSGetSMSList_t, 1283
messageListSize
unpack_sms_SLQSGetSMSList_t, 1283
messageMode
SMSMemoryInfo, 980
SMSMessageMode, 980
sMSMessageModeInfo, 980
unpack_sms_SLQSWmsMemoryFullCallBack_ind-_t, 1288
MessageModeInfo
messageModeTlv, 435
messageModeTlv, 435
MessageModeInfo, 435
TlvPresent, 435
messageSize
pack_sms_SaveSMS_t, 716
pack_sms_SendSMS_t, 716
slqssendasyncsmsparams_s, 948
slqssendsmsparams_s, 949
sms_sendAsyncsmsParams, 968
unpack_sms_SLQSGetSMS_t, 1281
wcdmaMsgEncodingParams, 1502
messageTag
pack_sms_SLQSModifySMSStatus_t, 721
qmiSmsMessageList, 859
unpack_sms_SLQSGetSMS_t, 1281
messageType
sms_routeEntry, 966
smsRouteEntry, 984

messageWaitingInfoContent, 435
activeInd, 436
msgCount, 436
msgType, 436
MicMute
GetAudioProfileResp, 274
GetM2MAudioProfileResp, 291
GetM2MAVMuteResp, 293
SetAudioProfileReq, 913
SetM2MAVMuteReq, 924
min
_SLQSAVMSSettingsPeriodsInfo, 89
minBasedIMSI, 436
imsiM1112, 437
imsiMS1, 437
imsiMS2, 437
mccM, 437
MinPolicedPktSz
unpack_qos_swiQosFlow_t, 1272
minSize
unpack_dms_GetVoiceNumber_t, 1126
minute
nas_timeInfo, 553
nas_UniversalTime, 559
timeInfo, 1033
UniversalTime, 1101
mipMode
unpack_wds_GetMobileIP_t, 1358
mipStatus
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
mipstatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1382
mitigationDevID
TmdDeRegNotMitigationLvlReq, 1033
TmdGetMitigationLvlReq, 1034
TmdMitigationLvlIndReq, 1036
TmdRegNotMitigationLvlReq, 1036
mitigationDevIDLen
TmdDeRegNotMitigationLvlReq, 1033
TmdGetMitigationLvlReq, 1034
TmdMitigationLvlIndReq, 1036
TmdRegNotMitigationLvlReq, 1036
mitigationDevId
mitigationDevList, 437
mitigationDevIdLen
mitigationDevList, 437
MitigationDevInfo
QmiCbkTmdMitiLvlRptInd, 854
mitigationDevList, 437
maxMitigationLevel, 437
mitigationDevId, 437
mitigationDevIdLen, 437
mnc
CSGID, 200
MNRInfo, 438
nas_CSGID, 457
nas_MNRInfo, 501
nas_netSelectionPref, 503
nas_OperatorPLMNData, 510
nasPLMNNameReq, 594
netSelectionPref, 610
OperatorPLMNData, 629
pack_nas_SLQSGetPLMNName_t, 671
PlmnID, 802
unpack_nas_GetHomeNetwork_t, 1189
mncPcsDigits
CSGID, 200
nas_CSGID, 457
mobileCountryCode
SMSEtwsPlmn, 976
sMSEtwsPlmnInfo, 977
mobileIP
pack_wds_SLQSSetWdsEventCallback_t, 780
mobileNetworkCode
SMSEtwsPlmn, 976
sMSEtwsPlmnInfo, 977
mode
callInfo, 155
pack_dms_SetEventReport_t, 635
pack_dms_SetPower_t, 637
pack_loc_SetOperationMode_t, 653
pack_sms_SLQSGetSmsBroadcastConfig_t, 719
pack_sms_SLQSSetSmsBroadcastActivation_t, 722
pack_sms_SLQSSetSmsBroadcastConfig_t, 723
pack_wds_SetMobileIP_t, 768
uim_refreshevent, 1061
UIMRefreshEvent, 1081
voice_callInfo, 1417
ModePref
NASModePreferenceTlv, 584
modelString
pack_dms_SLQSSwiSetHostDevInfo_t, 641
unpack_dms_SLQSSwiGetHostDevInfo_t, 1146
modelid
unpack_dms_GetModelID_t, 1118
modelid_str
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
modemMode
CommInfo, 191
nas_CommInfo, 456
modemTempNotification
qaGobiApiCbk.h, 1729
ModemTempState
_modemTempNotification, 70
ModemTemperature
_modemTempNotification, 70
modemindex
pack_fms_SetImagesPreference_t, 649
ModifyProfileIn, 438
curProfile, 439
pProfileID, 439
pProfileType, 439

ModifyProfileOut, 439
pExtErrorCode, 439
month
nas_timeInfo, 553
nas_UniversalTime, 559
timeInfo, 1033
UniversalTime, 1101
msgCount
messageWaitingInfoContent, 436
sms_messageWaitingInfoContent, 963
msgDelFailureCause
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
msgDelFailureType
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
msgProtocol
sms_msgProtocolResp, 964
smsMsgprotocolResp, 981
msgType
messageWaitingInfoContent, 436
sms_messageWaitingInfoContent, 963
msgWaitInfo
getMsgWaitingInfo, 295
msgWaitingInfo, 440
sms_getMsgWaitingInfo, 960
unpack_sms_SLQSWmsMessageWaitingCall-Back_ind_t, 1289
msgWaitingInfo, 439
msgWaitInfo, 440
numInstances, 440
msgid
pack_qmi_t, 712
unpack_qmi_t, 1254
msgtype
common.h, 1551
Mtu
unpack_wds_SLQSGetRuntimeSettings_t, 1378
mtuSize3gppTlv, 440
MTUSize3gpp, 440
TlvPresent, 440
MultDisc
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
multiplier
pktErrRate, 801
unpack_qos_pktErrRate_t, 1257
MuxID
qos_BindDataPortMuxID_t, 865

NAI
unpack_wds_GetMobileIPProfile_t, 1360
NAM_NAME_LENGTH
qaGobiApiNas.h, 1896
NAS_NAM_NAME_LENGTH
nas.h, 1644
NAS_PLMN_LENGTH
nas.h, 1644
NAS_SRV
qaGobiApiCbk.h, 1726
NASBandPreferenceTlv, 566
band_pref, 566
TlvPresent, 566
NASEmergencyModeTlv, 567
EmerMode, 568
TlvPresent, 568
NASGWAcqOrderPrefTlv, 578
GWAcqOrderPref, 578
TlvPresent, 578
NASLTEBandPreferenceTlv, 582
LTEBandPref, 583
TlvPresent, 583
NASLteNasReleaseInfoTlv, 583
nas_major, 583
nas_minor, 583
nas_release, 583
TlvPresent, 584
NASModePreferenceTlv, 584
ModePref, 584
TlvPresent, 584
NASNetSelPreferenceTlv, 584
NetSelPref, 584
TlvPresent, 584
NASOTAMessageTlv, 589
data_buf, 590
data_len, 590
message_type, 590
TlvPresent, 590
NASPRLPreferenceTlv, 597
PRLPref, 597
TlvPresent, 597
NASPhyCaAggPcellInfo, 590
dl_bw_value, 590
freq, 590
iLTEbandValue, 590
pci, 590
TlvPresent, 591
NASPhyCaAggScellDlBw, 591
dl_bw_value, 591
TlvPresent, 591
NASPhyCaAggScellIndType, 592
freq, 592
pci, 592
scell_state, 592
TlvPresent, 592
NASPhyCaAggScellIndex, 591
scell_idx, 591
TlvPresent, 592
NASPhyCaAggScellInfo, 592
dl_bw_value, 593
freq, 593
iLTEbandValue, 593
pci, 593
scell_state, 593
TlvPresent, 593

NASQmiCbkNasSwiOTAMessageInd, 597
nasRelInfoTlv, 597
otaMsgTlv, 598
timeTlv, 598
NASQmiCbkNasSystemSelPrefInd, 598
BPTlv, 599
EMTlv, 599
GWAOPTlv, 599
LBPTlv, 599
MPTlv, 599
NSPTlv, 599
PRLPTlv, 599
RPTlv, 599
SDPTlv, 599
NASRoamPreferenceTlv, 599
RoamPref, 599
TlvPresent, 599
NASServDomainPrefTlv, 599
SrvDomainPref, 600
TlvPresent, 600
NASServingSystemInfo, 600
csAttachState, 601
hdrPersonality, 601
psAttachState, 601
radioInterfaceList, 601
radioInterfaceNo, 601
registrationState, 601
selectedNetwork, 601
NASTimeInfoTlv, 609
time, 609
TlvPresent, 609
NMEAData
unpack_loc_EventNMEA_Ind_t, 1166
NSPTlv
NASQmiCbkNasSystemSelPrefInd, 599
NSSAudioCtrl, 619
downLink, 620
upLink, 620
NUM_OF_SET
qaGobiApiCbk.h, 1726
qaGobiApiSms.h, 1971
nUsbNetNum
pack_swidms_SLQSSwiDmsSetUsbNetNum_t, 725
NWProfile, 620
pProfSz, 620
pProfValues, 620
tech, 620
NWQoSStatus
unpack_qos_SLQSQosGetNetworkStatus_t, 1260
NWRegStat
_transNWRegInfoNotification, 114
unpack_sms_SLQSNWRegInfoCallback_ind_t, 1284
naiSize
unpack_wds_GetMobileIPProfile_t, 1360
namID
airTimer, 119
nasGet3GPP2SubscriptionInfoReq, 568
pack_nas_SLQSNasGet3GPP2Subscription_t, 677
prefVoiceSO, 810
roamTimer, 886
voice_airTimer, 1397
voice_prefVoiceSO, 1438
voice_roamTimer, 1442
namName, 441
namName, 441
namNameLen, 441
namName, 441
nas_namName, 502
namNameLen
namName, 441
nas_namName, 502
Name
unpack_nas_GetServingNetwork_t, 1193
name
unpack_nas_GetHomeNetwork_t, 1189
unpack_wds_GetDefaultProfile_t, 1356
nameLen
remotePartyName, 876
voice_remotePartyName, 1440
namePI
remotePartyName, 876
voice_remotePartyName, 1440
nameSize
unpack_nas_GetHomeNetwork3GPP2_t, 1188
unpack_nas_GetServingNetwork_t, 1193
nameString
pack_dms_SLQSSwiSetOSInfo_t, 642
unpack_dms_SLQSSwiGetOSInfo_t, 1147
namelength
omaDmFotaTlv, 625
omaDmFotaTlvExt, 627
unpack_omaDmFotaTlv_t, 1241
namesize
unpack_wds_GetDefaultProfile_t, 1356
nas.h
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6, 1644
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75, 1644
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1644
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1644
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1644

eNAS_LTE_CPHY_CA_BW_NRB_LITE_100, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6, 1644
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75, 1644
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE, 1645
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE, 1645
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE, 1645
nas.h, 1635
NAS_PLMN_LENGTH, 1644
pack_nas_GetACCOLC, 1645
pack_nas_GetANAAAAuthenticationStatus, 1645
pack_nas_GetCDMANetworkParameters, 1645
pack_nas_GetHomeNetwork, 1646
pack_nas_GetHomeNetwork3GPP2, 1646
pack_nas_GetNetworkPreference, 1646
pack_nas_GetRFInfo, 1646
pack_nas_GetServingNetwork, 1647
pack_nas_GetServingNetworkCapabilities, 1647
pack_nas_GetSignalStrengths, 1647
pack_nas_InitiateDomainAttach, 1648
pack_nas_PerformNetworkScan, 1648
pack_nas_SLQSConfigSigInfo, 1650
pack_nas_SLQSGetErrorRate, 1651
pack_nas_SLQSGetNetworkTime, 1651
pack_nas_SLQSGetOperatorNameData, 1651
pack_nas_SLQSGetPLMNName, 1652
pack_nas_SLQSGetServingSystem, 1652
pack_nas_SLQSGetSignalStrength, 1652
pack_nas_SLQSGetSysInfo, 1653
pack_nas_SLQSGetSysSelectionPref, 1653
pack_nas_SLQSInitiateNetworkRegistration, 1654
pack_nas_SLQSNASGeteDRXParams, 1655
pack_nas_SLQSNASSeteDRXParams, 1657
pack_nas_SLQSNASSwiGetChannelLock, 1657
pack_nas_SLQSNASSwiSetChannelLock, 1658
pack_nas_SLQSNasConfigSigInfo2, 1654
pack_nas_SLQSNasGet3GPP2Subscription, 1654
pack_nas_SLQSNasGetCellLocationInfo, 1655
pack_nas_SLQSNasGetHDRColorCode, 1655
pack_nas_SLQSNasGetRFInfo, 1656
pack_nas_SLQSNasGetSigInfo, 1656
pack_nas_SLQSNasGetTxRxInfo, 1656
pack_nas_SLQSNasIndicationRegisterExt, 1657
pack_nas_SLQSNasSwiIndicationRegister, 1658
pack_nas_SLQSNasSwiModemStatus, 1658
pack_nas_SLQSSetBandPreference, 1659
pack_nas_SLQSSetSignalStrengthsCallback, 1659
pack_nas_SLQSSetSysSelectionPref, 1660
pack_nas_SLQSSetSysSelectionPrefExt, 1660
pack_nas_SLQSSwiGetHDRPersonality, 1660
pack_nas_SLQSSwiGetHDRProtSubtype, 1661
pack_nas_SLQSSwiGetHRPDStats, 1661
pack_nas_SLQSSwiGetLteCQI, 1661
pack_nas_SLQSSwiGetLteSccRxInfo, 1662
pack_nas_SLQSSwiNetworkDebug, 1662
pack_nas_SLQSSwiPSDetach, 1662
pack_nas_SetACCOLC, 1649
pack_nas_SetCDMANetworkParameters, 1649
pack_nas_SetLURejectCallback, 1649
pack_nas_SetNetworkPreference, 1650
pack_nas_SetRFInfoCallback, 1650
pack_nas_SlqsGetLTECphyCAInfo, 1651
unpack_nas_GetACCOLC, 1663
unpack_nas_GetANAAAAuthenticationStatus, 1663
unpack_nas_GetCDMANetworkParameters, 1664
unpack_nas_GetHomeNetwork, 1664
unpack_nas_GetHomeNetwork3GPP2, 1664
unpack_nas_GetNetworkPreference, 1665
unpack_nas_GetRFInfo, 1665
unpack_nas_GetServingNetwork, 1665
unpack_nas_GetServingNetworkCapabilities, 1666
unpack_nas_GetSignalStrengths, 1666
unpack_nas_InitiateDomainAttach, 1666
unpack_nas_PerformNetworkScan, 1667
unpack_nas_SLQSConfigSigInfo, 1670
unpack_nas_SLQSGetErrorRate, 1671
unpack_nas_SLQSGetNetworkTime, 1671
unpack_nas_SLQSGetOperatorNameData, 1672
unpack_nas_SLQSGetPLMNName, 1672
unpack_nas_SLQSGetServingSystem, 1673
unpack_nas_SLQSGetSignalStrength, 1673
unpack_nas_SLQSGetSysInfo, 1673
unpack_nas_SLQSGetSysSelectionPref, 1674
unpack_nas_SLQSInitiateNetworkRegistration, 1674
unpack_nas_SLQSNASGeteDRXParams, 1675
unpack_nas_SLQSNASSeteDRXParams, 1678
unpack_nas_SLQSNASSwiGetChannelLock, 1679
unpack_nas_SLQSNASSwiSetChannelLock, 1680
unpack_nas_SLQSNasConfigSigInfo2, 1674
unpack_nas_SLQSNasGet3GPP2Subscription, 1675
unpack_nas_SLQSNasGetCellLocationInfo, 1675
unpack_nas_SLQSNasGetHDRColorCode, 1676
unpack_nas_SLQSNasGetRFInfo, 1676
unpack_nas_SLQSNasGetSigInfo, 1676
unpack_nas_SLQSNasGetTxRxInfo, 1677
unpack_nas_SLQSNasIndicationRegisterExt, 1677
unpack_nas_SLQSNasNetworkTimeCallBack_ind, 1678
unpack_nas_SLQSNasSigInfoCallback_ind, 1678
unpack_nas_SLQSNasSwiIndicationRegister, 1679
unpack_nas_SLQSNasSwiModemStatus, 1679
unpack_nas_SLQSNasSwiOTAMessageCallback_ind, 1680

unpack_nas_SLQSNasSysInfoCallback_ind, 1680
unpack_nas_SLQSNasTimerCallback_ind, 1681
unpack_nas_SLQSSetBandPreference, 1681
unpack_nas_SLQSSetSignalStrengthsCallback, 1681
unpack_nas_SLQSSetSysSelectionPref, 1681
unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind, 1682
unpack_nas_SLQSSetSysSelectionPrefExt, 1682
unpack_nas_SLQSSwiGetHDRPersonality, 1682
unpack_nas_SLQSSwiGetHDRProtSubtype, 1683
unpack_nas_SLQSSwiGetHRPDStats, 1683
unpack_nas_SLQSSwiGetLteCQI, 1684
unpack_nas_SLQSSwiGetLteSccRxInfo, 1684
unpack_nas_SLQSSwiHDRPersonalityCallback_-Ind, 1684
unpack_nas_SLQSSwiNetworkDebug, 1685
unpack_nas_SLQSSwiPSDetach, 1685
unpack_nas_SLQSSwiRandIndicatorCallback_Ind, 1685
unpack_nas_SetACCOLC, 1667
unpack_nas_SetCDMANetworkParameters, 1667
unpack_nas_SetDataCapabilitiesCallback_ind, 1668
unpack_nas_SetEventReportInd, 1668
unpack_nas_SetLURejectCallback, 1668
unpack_nas_SetNasLTECphyCaIndCallback_ind, 1669
unpack_nas_SetNetworkPreference, 1669
unpack_nas_SetRFInfoCallback, 1669
unpack_nas_SetRoamingIndicatorCallback_ind, 1670
unpack_nas_SetServingSystemCallback_ind, 1670
unpack_nas_SlqsGetLTECphyCAInfo, 1671
unpack_valid_nas_GetCDMANetworkParameters, 1686
unpack_valid_nas_SLQSGetServingSystem, 1686
unpack_valid_nas_SLQSGetSignalStrength, 1687
unpack_valid_nas_SLQSNasGetSigInfo, 1688
nas_ActPilotPNElement, 442
ActSetPilotPN, 442
ActSetPilotPNStrength, 442
nas_AddCDMASysInfo, 442
geoSysIdx, 443
regPrd, 443
nas_AddSysInfo, 443
cellBroadcastCap, 443
geoSysIdx, 443
nas_CDMAChannel, 445
priChA, 446
priChB, 446
secChA, 446
secChB, 446
nas_CDMAECIOThresh, 446
CDMAECIOThreshListLen, 446
pCDMAECIOThreshList, 446
nas_CDMAInfo, 446
baseId, 447
baseLat, 447
baseLong, 447
nid, 447
refpn, 447
sid, 447
nas_CDMARSSIThresh, 447
CDMARSSIThreshListLen, 448
pCDMARSSIThreshList, 448
nas_CDMASysInfo, 448
baseId, 451
baseLat, 451
baseLong, 451
bsInfoValid, 451
bsPRev, 451
bsPRevValid, 451
ccsSupported, 451
ccsSupportedValid, 451
cdmaSysIdValid, 451
isSysPrlMatch, 451
isSysPrlMatchValid, 451
MCC, 451
MNC, 451
networkID, 451
networkIdValid, 451
pRevInUse, 451
pRevInUseValid, 452
packetZone, 451
packetZoneValid, 451
sysInfoCDMA, 452
systemID, 452
nas_CDMASysInfoExt, 452
imsi_11_12, 452
MCC, 452
nas_CSGID, 456
id, 456
mcc, 457
mnc, 457
mncPcsDigits, 457
rat, 457
nas_CallBarringSysInfo, 444
csBarStatus, 444
psBarStatus, 444
nas_CommInfo, 454
imsRegState, 456
modemMode, 456
psState, 456
systemMode, 456
temperature, 456
nas_DRCParams, 463
DRCCover, 464
DRCValue, 464
nas_DataStatusDetail, 458
IPAddress, 460
LastErrCode, 460
nas_DeviceConfigDetail, 462
Chipset, 463
HWVersion, 463

QLIC, 463
Technology, 463
nas_ECIOThresh, 465
ECIOThresListLen, 465
pECIOThresList, 465
nas_GERANInfo, 466
arfcn, 468
bsic, 468
cellID, 468
insNmrCellInfo, 468
lac, 468
nmrInst, 468
plmn, 468
rxLev, 468
timingAdvance, 468
nas_GSMRSSIThresh, 470
GSMRSSIThreshListLen, 470
pGSMRSSIThreshList, 470
nas_GSMSrvStatusInfo, 470
isPrefDataPath, 471
srvStatus, 471
trueSrvStatus, 471
nas_GSMSysInfo, 471
cellId, 473
cellIdValid, 473
dtmSupp, 473
dtmSuppValid, 473
egprsSupp, 474
egprsSuppValid, 474
lac, 474
lacValid, 474
MCC, 474
MNC, 474
networkIdValid, 474
regRejectInfoValid, 474
rejCause, 474
rejectSrvDomain, 474
sysInfoGSM, 474
nas_HDRECIOThresh, 474
HDRECIOThreshListLen, 474
pHDRECIOThreshList, 474
nas_HDRIOThresh, 474
HDRIOThreshListLen, 475
pHDRIOThreshList, 475
nas_HDRPersonality_Ind_Data, 475
pCurrentPersonality, 475
pPersonalityListLength, 476
pProtocolSubtypeElement, 476
nas_HDRRSSIThresh, 476
HDRRSSIThreshListLen, 476
pHDRRSSIThreshList, 476
nas_HDRSINRThresh, 476
HDRSINRThresListLen, 477
pHDRSINRThresList, 477
nas_HDRSINRThreshold, 477
HDRSINRThreshListLen, 478
pHDRSINRThreshList, 478
nas_HDRSysInfo, 478
hdrActiveProt, 479
hdrActiveProtValid, 479
hdrPersonality, 479
hdrPersonalityValid, 479
is856SysId, 479
is856SysIdValid, 479
isSysPrlMatch, 479
isSysPrlMatchValid, 479
sysInfoHDR, 479
nas_IOThresh, 481
IOThresListLen, 482
pIOThresList, 482
nas_LTEInfo, 484
band, 485
bandwidth, 485
emmConnState, 486
emmState, 486
emmSubState, 486
RXChan, 486
TXChan, 486
nas_LTEInfoInterfreq, 486
freqsLen, 486
InfoInterfreq, 486
ueInIdle, 486
nas_LTEInfoIntrafreq, 486
CellParams, 488
cellReselPriority, 488
cellsLen, 488
earfcn, 488
globalCellId, 488
plmn, 488
sIntraSearch, 488
sNonIntraSearch, 488
servingCellId, 488
tac, 488
threshServingLow, 488
ueInIdle, 488
nas_LTEInfoNeighboringGSM, 488
freqsLen, 489
LteGsmCellInfo, 489
ueInIdle, 489
nas_LTEInfoNeighboringWCDMA, 489
freqsLen, 490
ueInIdle, 490
nas_LTEOperationMode, 490
pLTEOperationMode, 490
TlvPresent, 490
nas_LTERSRPThresh, 491
LTERSRPThreshListLen, 492
pLTERSRPThreshList, 492
nas_LTERSRQThresh, 492
LTERSRQThreshListLen, 492
pLTERSRQThreshList, 492
nas_LTERSSIThresh, 492
LTERSSIThreshListLen, 493
pLTERSSIThreshList, 493
nas_LTESNRThresh, 495
LTESNRThresListLen, 496

pLTESNRThresList, 496
nas_LTESNRThreshold, 496
LTESNRThreshListLen, 496
pLTESNRThreshList, 496
nas_LTESigRptCfg, 493
avgPeriod, 494
rptRate, 494
nas_LTESigRptConfig, 494
avgPeriod, 495
rptRate, 495
nas_LTESysInfo, 497
cellId, 498
cellIdValid, 499
lac, 499
lacValid, 499
MCC, 499
MNC, 499
networkIdValid, 499
regRejectInfoValid, 499
rejCause, 499
rejectSrvDomain, 499
sysInfoLTE, 499
tac, 499
tacValid, 499
nas_MNRInfo, 501
mcc, 501
mnc, 501
rat, 501
nas_NetworkStat1x, 503
ActSetCnt, 505
NeighborSetCnt, 505
pActPilotPNElements, 505
pNeighborSetPilotPN, 505
RX_EC_IO, 505
RX_PWR, 505
SO, 505
State, 505
TX_PWR, 505
nas_NetworkStatEVDO, 505
MACIndex, 507
PER, 507
pSectorID, 507
PilotEnergy, 507
RX_PWR, 507
SNR, 507
SectorIDLen, 507
State, 507
nas_OperatorPLMNData, 509
lac1, 510
lac2, 510
mcc, 510
mnc, 510
PLMNRecID, 510
nas_PLMNNetworkName, 517
numInstance, 518
PLMNNetName, 518
nas_PLMNNetworkNameData, 518
codingScheme, 519
countryInitials, 519
longName, 519
longNameLen, 519
longNameSpareBits, 519
shortName, 519
shortNameLen, 519
shortNameSpareBits, 519
nas_PhyCaAggPcellInfo, 511
dl_bw_value, 511
freq, 511
iLTEbandValue, 511
pci, 511
TlvPresent, 511
nas_PhyCaAggScellDlBw, 512
dl_bw_value, 512
TlvPresent, 512
nas_PhyCaAggScellIndType, 513
freq, 513
pci, 513
scell_state, 513
TlvPresent, 513
nas_PhyCaAggScellIndex, 512
scell_idx, 512
TlvPresent, 512
nas_PhyCaAggScellInfo, 513
dl_bw_value, 516
freq, 516
iLTEbandValue, 516
pci, 516
scell_state, 516
TlvPresent, 516
nas_PilotSetData, 516
NumPilots, 516
pPilotSetInfo, 516
nas_PilotSetParams, 517
PilotPN, 517
PilotStrength, 517
PilotType, 517
nas_QmiNas3GppNetworkInfo, 522
Desription, 523
Forbidden, 523
InUse, 523
MCC, 523
MNC, 523
Preferred, 523
Roaming, 523
nas_QmiNas3GppNetworkRAT, 523
MCC, 523
MNC, 524
RAT, 524
nas_QmisNasPcsDigit, 524
includes_pcs_digit, 524
MCC, 524
MNC, 524
nas_RFBandInfoElements, 525
activeBandClass, 526
activeChannel, 526
radioInterface, 526

nas_RFInfoTlv, 530
activeBandClass, 531
activeChannel, 531
radioInterface, 531
radioInterfaceSize, 531
TlvPresent, 531
nas_RSRPThresh, 532
pRSRPThresList, 533
RSRPThresListLen, 533
nas_RSRQThresh, 534
pRSRQThresList, 534
RSRQThresListLen, 534
nas_RSSIThresh, 534
pRSSIThresList, 535
RSSIThresListLen, 535
nas_RankIndicatorTlv, 524
count0, 525
count1, 525
TlvPresent, 525
nas_RejectReasonTlv, 525
rejectCause, 525
serviceDomain, 525
TlvPresent, 525
nas_RfBandInfoExtFormat, 526
pInstancesSize, 527
pRfBandInfoExtFormatParam, 527
TlvPresent, 527
nas_RfBandInfoExtFormatElements, 527
activeBand, 527
activeChannel, 527
radioInterface, 527
nas_RfBandInfoList, 527
pInstanceSize, 528
pRFBandInfoParam, 528
nas_RfBandwidthInfo, 528
pInstancesSize, 528
pRfBandwidthInfoParam, 528
TlvPresent, 528
nas_RfBandwidthInfoElements, 529
bandwidth, 529
radioInterface, 529
nas_RfDedicatedBandInfo, 529
pInstancesSize, 530
pRfDedicatedBandInfoParam, 530
TlvPresent, 530
nas_RfDedicatedBandInfoElements, 530
dedicatedBand, 530
radioInterface, 530
nas_RxSigInfo, 536
isRadioTuned, 537
rsrp, 537
rxChainIndex, 537
rxPower, 537
nas_SLQSSignalStrengthsIndReq, 542
ecioDelta, 543
ecioThresholdList, 543
ecioThresholdListLen, 543
ioDelta, 544
lteRsrpDelta, 544
lteSnrDelta, 544
rsrqDelta, 544
rxSignalStrengthDelta, 544
sinrDelta, 544
sinrThresholdList, 544
sinrThresholdListLen, 544
nas_SLQSSignalStrengthsInformation, 544
ecioInfo, 544
errorRateInfo, 544
io, 544
lteRsrpinfo, 544
lteSnrinfo, 545
rsrqInfo, 545
rxSignalStrengthInfo, 545
sinr, 545
nas_SLQSSignalStrengthsTlv, 545
sSLQSSignalStrengthsInfo, 545
TlvPresent, 545
nas_SccRxInfo, 538
numInstances, 539
rsrq, 539
sigInfo, 539
snr, 539
TlvPresent, 539
nas_SignalStrengthTlv, 542
radioInterface, 542
signalStrength, 542
TlvPresent, 542
nas_SrvStatusInfo, 545
isPrefDataPath, 546
srvStatus, 546
nas_TDSCDMAECIOThresh, 549
nas_TDSCDMARSCPThresh, 549
nas_TDSCDMARSSIThresh, 550
nas_TDSCDMASINRCONFThresh, 550
nas_TDSCDMASINRThresh, 551
nas_UMTSInfo, 554
cellID, 556
ecio, 556
geranInst, 556
GeranInstInfo, 556
lac, 556
plmn, 556
psc, 556
rscp, 556
UMTSInstInfo, 556
uarfcn, 556
umtsInst, 556
nas_UMTSinstInfo, 556
umtsEcio, 557
umtsPsc, 557
umtsRscp, 557
umtsUarfcn, 557
nas_UniversalTime, 558
day, 559
dayOfWeek, 559
hour, 559

minute, 559
month, 559
second, 559
year, 559
nas_WCDMAECIOThresh, 560
nas_WCDMAInfoLTENeighborCell, 560
umtsLTENbrCellLen, 561
wcdmaRRCState, 561
nas_WCDMARSSIThresh, 561
nas_WCDMASysInfo, 562
cellId, 564
cellIdValid, 565
hsCallStatus, 565
hsCallStatusValid, 565
hsInd, 565
hsIndValid, 565
lac, 565
lacValid, 565
MCC, 565
MNC, 565
networkIdValid, 565
psc, 565
pscValid, 565
regRejectInfoValid, 565
rejCause, 565
rejectSrvDomain, 565
sysInfoWCDMA, 565
nas_acqOrderPref, 441
acqOrdeLen, 442
pAcqOrder, 442
nas_callBarStatus, 444
csBarStatus, 445
psBarStatus, 445
nas_cellParams, 452
pci, 453
rsrp, 453
rsrq, 453
rssi, 453
srxlev, 453
nas_ciotAcqOrderPref, 453
ciotAcqOrderLen, 454
pCiotAcqOrder, 454
nas_currentPLMN, 457
MCC, 457
MNC, 457
netDescr, 457
netDescrLength, 458
nas_dataSrvCapabilities, 458
dataCapabilities, 458
dataCapabilitiesLen, 458
nas_detailSvcInfo, 461
hdrHybrid, 462
hdrSrvStatus, 462
isSysForbidden, 462
srvCapability, 462
srvStatus, 462
nas_dirNum, 463
dirNum, 463
dirNumLen, 463
nas_ecioListElement, 464
ecio, 464
radioIf, 465
nas_errorRateListElement, 465
errorRate, 466
radioIf, 466
nas_geranInstInfo, 468
geranArfcn, 468
geranBsicBcc, 468
geranBsicNcc, 469
geranRssi, 469
nas_gsmCellInfo, 469
arfcn, 469
band1900, 470
bsicId, 470
cellIdValid, 470
rssi, 470
srxlev, 470
nas_homeSIDNID, 480
numInstances, 480
SidNid, 480
nas_infoInterFreq, 480
cell_resel_priority, 481
cellInterFreqParams, 481
cells_len, 481
earfcn, 481
threshXHigh, 481
threshXLow, 481
nas_lteEARFCN, 482
earfcn0, 482
earfcn1, 482
status, 482
nas_lteGsmCellInfo, 483
cellReselPriority, 483
cells_len, 483
GsmCellInfo, 484
nccPermitted, 484
threshGsmHigh, 484
threshGsmLow, 484
nas_ltePCI, 490
earfcn, 491
pci, 491
status, 491
nas_lteRsrpinformation, 491
rsrplevel, 491
nas_lteSnrinformation, 495
snrlevel, 495
nas_lteWcdmaCellInfo, 499
cellReselPriority, 500
cellsLen, 500
threshXhigh, 500
threshXlow, 500
uarfcn, 500
WCDMACellInfo, 500
nas_major
LteNasReleaseInfo_s, 422
NASLteNasReleaseInfoTlv, 583

nas_minBasedIMSI, 500
imsiM1112, 501
imsiMS1, 501
imsiMS2, 501
mccM, 501
nas_minor
LteNasReleaseInfo_s, 422
NASLteNasReleaseInfoTlv, 583
nas_namName, 502
namName, 502
namNameLen, 502
nas_netSelectionPref, 502
mcc, 503
mnc, 503
netReg, 503
nas_nmrCellInfo, 507
nmrArfcn, 508
nmrBsic, 508
nmrCellID, 508
nmrLac, 508
nmrPlmn, 508
nmrRxLev, 508
nas_nr5gBandPref, 508
bits_129_192, 509
bits_193_256, 509
bits_1_64, 509
bits_65_128, 509
nas_operatorNameString, 509
PLMNName, 509
nas_operatorPLMNList, 510
numInstance, 511
PLMNData, 511
nas_protocolSubtypeElement, 519
AccessMac, 520
AuthProt, 521
ControlMac, 521
EncryptProt, 521
ForwardMac, 521
IdleState, 521
KeyExchange, 521
MultDisc, 521
PhysicalLayer, 521
ReverseMac, 521
SecProt, 521
VirtStream, 521
nas_qaQmi3Gpp2TimeZone, 521
daylightSavings, 521
leapSeconds, 522
localTimeOffset, 522
nas_release
LteNasReleaseInfo_s, 422
NASLteNasReleaseInfoTlv, 583
nas_roamIndList, 532
numInstances, 532
radioInterface, 532
roamIndicator, 532
nas_rsrqInformation, 533
radioIf, 533
rsrq, 533
nas_rxInfo, 535
ecio, 536
isRadioTuned, 536
phase, 536
rscp, 536
rsrp, 536
rxPower, 536
nas_rxSignalStrengthListElement, 537
radioIf, 538
rxSignalStrength, 538
nas_servSystem, 540
csAttachState, 541
numRadioInterfaces, 541
psAttachState, 541
radioInterface, 541
regState, 541
selNetwork, 541
nas_serviceProviderName, 539
displayCondition, 539
spn, 539
spnLength, 539
nas_sidNid, 541
nid, 541
sid, 541
nas_sysInfoCommon, 546
isSysForbidden, 548
isSysForbiddenValid, 548
roamStatus, 548
roamStatusValid, 548
srvCapability, 548
srvCapabilityValid, 548
srvDomain, 548
srvDomainValid, 548
nas_timeInfo, 551
day, 553
dayLtSavingAdj, 553
dayOfWeek, 553
hour, 553
minute, 553
month, 553
radioInterface, 553
second, 553
timeZone, 553
TlvPresent, 553
year, 553
nas_trueIMSI, 553
imsiT1112, 554
imsiTS1, 554
imsiTS2, 554
imsiTaddrNum, 554
mccT, 554
nas_txInfo, 554
isInTraffic, 554
txPower, 554
nas_umtsLTENbrCell, 557
cellIsTDD, 558
earfcn, 558

pci, 558
rsrp, 558
rsrq, 558
srxlev, 558
nas_wcdmaCellInfo, 559
cpich_ecno, 560
cpich_rscp, 560
psc, 560
srxlev, 560
nas_wcdmaUARFCN, 565
status, 566
uarfcn, 566
nasCellLocationInfoResp, 566
pCDMAInfo, 567
pGERANInfo, 567
pLTEInfoInterfreq, 567
pLTEInfoIntrafreq, 567
pLTEInfoNeighboringGSM, 567
pLTEInfoNeighboringWCDMA, 567
pUMTSCellID, 567
pUMTSInfo, 567
pWCDMAInfoLTENeighborCell, 567
nasGet3GPP2SubscriptionInfoReq, 568
namID, 568
nasGet3GPP2SubscriptionInfoResp, 568
pCDMAChannel, 569
pDirNum, 569
pHomeSIDNID, 569
pMinBasedIMSI, 569
pNAMNameInfo, 569
pTrueIMSI, 569
nasGetHDRColorCodeResp, 570
pColorCode, 570
nasGetLTECphyCa, 570
sPhyCaAggPcellInfo, 571
sPhyCaAggScellDlBw, 571
sPhyCaAggScellIndType, 571
sPhyCaAggScellIndex, 571
sPhyCaAggScellInfo, 571
NasGetLTECphyCaInfo, 571
PhyCaAggPcellInfo, 571
PhyCaAggScellDlBw, 571
PhyCaAggScellIndType, 572
PhyCaAggScellIndex, 571
PhyCaAggScellInfo, 572
nasGetLTECphyCaResp, 572
pPhyCaAggPcellInfo, 572
pPhyCaAggScellDlBw, 572
pPhyCaAggScellIndType, 572
pPhyCaAggScellIndex, 572
pPhyCaAggScellInfo, 572
NasGetRFInfoResp, 572
pLTEOperationMode, 573
pRfBandInfoExtFormat, 573
pRfBandwidthInfo, 573
pRfDedicatedBandInfo, 573
rfbandInfoList, 573
nasGetSigInfoResp, 573
pCDMASSInfo, 574
pGSMSSInfo, 574
pHDRSSInfo, 574
pLTESSInfo, 574
pTDSCDMASigInfoExt, 574
pTDSCDMASigInfoRscp, 574
pWCDMASSInfo, 574
nasGetSysInfoResp, 574
pAddCDMASysInfo, 576
pAddGSMSysInfo, 576
pAddHDRSysInfo, 576
pAddLTESysInfo, 576
pAddWCDMASysInfo, 576
pCDMASrvStatusInfo, 576
pCDMASysInfo, 576
pGSMCallBarringSysInfo, 576
pGSMCipherDomainSysInfo, 576
pGSMSrvStatusInfo, 576
pGSMSysInfo, 576
pHDRSrvStatusInfo, 576
pHDRSysInfo, 576
pLTESrvStatusInfo, 577
pLTESysInfo, 577
pLTEVoiceSupportSysInfo, 577
pWCDMACallBarringSysInfo, 577
pWCDMACipherDomainSysInfo, 577
pWCDMASrvStatusInfo, 577
pWCDMASysInfo, 577
nasGetTxRxInfoReq, 577
radio_if, 577
nasGetTxRxInfoResp, 577
pRXChain0Info, 578
pRXChain1Info, 578
pTXInfo, 578
nasGeteDRXParamsResp, 569
pCycleLen, 570
pEdrxEnable, 570
pPagingTimeWindow, 570
nasIndicationRegisterReq, 578
pDDTMInd, 581
pDualStandByPrefInd, 581
pErrorRateInd, 581
pHDRNewUATIAssInd, 581
pHDRSessionCloseInd, 581
pLTECphyCa, 581
pManagedRoamingInd, 581
pNetworkRejectInd, 581
pNetworkTimeInd, 581
pServingSystemInd, 581
pSignalStrengthInd, 581
pSubscriptionInfoInd, 581
pSuppressSysInfoInd, 581
pSysInfoInd, 581
pSystemSelectionInd, 581
nasInitNetworkReg, 582
pChangeDuration, 582
pMNRInfo, 582
pMncPcsDigitStatus, 582

regAction, 582
nasNetworkReject, 584
pCsgId, 587
pPlmnId, 587
radioIf, 587
rejectCause, 587
serviceDomain, 587
nasNetworkTime, 587
pDayltSavAdj, 588
pRadioInterface, 588
pTimeZone, 588
universalTime, 588
nasOperatorNameResp, 588
pNITZInformation, 589
pOperatorNameString, 589
pOperatorPLMNList, 589
pPLMNNetworkName, 589
pSrvcProviderName, 589
nasPLMNNameReq, 593
mcc, 594
mnc, 594
pMncPcsStatus, 594
nasPLMNNameResp, 594
longName, 596
longNameCI, 596
longNameEn, 596
longNameLen, 596
longNameSB, 596
shortName, 596
shortNameCI, 596
shortNameEn, 596
shortNameLen, 596
shortNameSB, 596
spn, 596
spnEncoding, 596
spnLength, 597
nasRelInfoTlv
NASQmiCbkNasSwiOTAMessageInd, 597
nasSeteDRXParamsReq, 601
pCycleLen, 602
pEdrxEnable, 602
pPagingTimeWindow, 602
nasSigInfo, 602
pCDMASigInfo, 603
pGSMSigInfo, 603
pHDRSigInfo, 603
pLTESigInfo, 603
pRscp, 603
pTDSCDMASigInfoExt, 603
pWCDMASigInfo, 603
nasSwiGetChannelLockResp, 603
pLteEARFCN, 604
pLtePCI, 604
pWcdmaUARFCN, 604
NasSwiIndReg, 604
gsmUmtsDl, 605
gsmUmtsUl, 605
lteEmmDl, 605
lteEmmUl, 605
lteEsmDl, 605
lteEsmUl, 605
pRankIndicatorInd, 605
pTimer, 605
nasSwiSetChannelLockReq, 605
pLteEARFCN, 605
pLtePCI, 605
pWcdmaUARFCN, 606
nasSysInfo, 606
pAddCDMASysInfo, 608
pAddGSMSysInfo, 608
pAddHDRSysInfo, 608
pAddLTESysInfo, 608
pAddWCDMASysInfo, 608
pCDMASrvStatusInfo, 608
pCDMASysInfo, 608
pGSMCallBarringSysInfo, 608
pGSMCipherDomainSysInfo, 608
pGSMSrvStatusInfo, 608
pGSMSysInfo, 608
pHDRSrvStatusInfo, 608
pHDRSysInfo, 608
pLTESrvStatusInfo, 608
pLTESysInfo, 608
pLTEVoiceSupportSysInfo, 608
pSysInfoNoChange, 608
pWCDMACallBarringSysInfo, 608
pWCDMACipherDomainSysInfo, 608
pWCDMASrvStatusInfo, 608
pWCDMASysInfo, 609
nasTimers, 609
t3396_apn, 610
t3396_plmn_id, 610
t3396_val, 610
nccPermitted
lteGsmCellInfo, 415
nas_lteGsmCellInfo, 484
NeighborSetCnt
nas_NetworkStat1x, 505
NetworkStat1x, 614
netDescr
currentPLMN, 205
nas_currentPLMN, 457
netDescrLength
currentPLMN, 205
nas_currentPLMN, 458
netInfoLen
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
netReg
nas_netSelectionPref, 503
netSelectionPref, 610
NetSelPref
NASNetSelPreferenceTlv, 584
netSelectionPref, 610
mcc, 610
mnc, 610

netReg, 610
NetStats, 610
rx_bytes, 611
rx_errors, 611
rx_overflows, 611
rx_packets, 611
tx_bytes, 611
tx_errors, 611
tx_overflows, 611
tx_packets, 611
Network Access Service (NAS), 42
NetworkDebugResp, 612
pDataStatusDetail, 612
pDeviceConfigDetail, 612
pNetworkStat1x, 612
pNetworkStatEVDO, 612
pObjectVer, 612
NetworkID
qaQmiServingSystemParam, 833
unpack_nas_SLQSGetServingSystem_t, 1208
networkID
CDMASysInfo, 179
nas_CDMASysInfo, 451
networkIdValid
CDMASysInfo, 179
GSMSysInfo, 312
LTESysInfo, 433
nas_CDMASysInfo, 451
nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
networkInfo
voice_USSDNotificationNetworkInfo, 1444
networkInfoLen
unpack_wds_SLQSGetCurrDataSystemStat_t, 1370
NetworkStat1x, 612
ActSetCnt, 614
NeighborSetCnt, 614
pActPilotPNElements, 614
pNeighborSetPilotPN, 614
RX_EC_IO, 614
RX_PWR, 614
SO, 614
State, 614
TX_PWR, 614
NetworkStatEVDO, 615
MACIndex, 616
PER, 616
pSectorID, 616
PilotEnergy, 616
RX_PWR, 616
SNR, 616
SectorIDLen, 616
State, 616
NetworkType
CurrNetworkInfo, 207
currNetworkInfo, 209
wds_currNetworkInfo, 1510
NewMMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1279
newMTMessageTlv, 616
MTMessageInfo, 617
TlvPresent, 617
newPINLen
uim_unblockUIMPIN, 1067
unblockUIMPIN, 1100
newPINVal
uim_unblockUIMPIN, 1067
unblockUIMPIN, 1100
newPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 755
voiceSetCallBarringPwdInfo, 1483
newPasswdAgain
pack_voice_SLQSVoiceSetCallBarringPassword_t, 755
voiceSetCallBarringPwdInfo, 1483
newPin
pack_dms_UIMUnblockPIN_t, 646
newPwd
newPwdData, 617
voice_newPwdData, 1433
newPwdAgain
newPwdData, 617
voice_newPwdData, 1433
newPwdData, 617
newPwd, 617
newPwdAgain, 617
newValue
pack_dms_UIMChangePIN_t, 643
nextHeader
LibPackTFTIDParams, 377
TFTIDParams, 1031
nid
CDMAInfo, 166
nas_CDMAInfo, 447
nas_sidNid, 541
sidNid, 935
unpack_nas_GetHomeNetwork_t, 1189
NmeaPort
DcsUsbPortNames, 223
nmrArfcn
nas_nmrCellInfo, 508
nmrCellInfo, 618
nmrBsic
nas_nmrCellInfo, 508
nmrCellInfo, 618
nmrCellID
nas_nmrCellInfo, 508
nmrCellInfo, 618
nmrCellInfo, 617
nmrArfcn, 618
nmrBsic, 618
nmrCellID, 618

nmrLac, 618
nmrPlmn, 618
nmrRxLev, 618
nmrInst
GERANInfo, 268
nas_GERANInfo, 468
nmrLac
nas_nmrCellInfo, 508
nmrCellInfo, 618
nmrPlmn
nas_nmrCellInfo, 508
nmrCellInfo, 618
nmrRxLev
nas_nmrCellInfo, 508
nmrCellInfo, 618
noReplyTimer
callFWExtInfo, 152
callFWInfo, 153
voice_callFWExtInfo, 1414
voice_callFWInfo, 1415
Non-service specific APIs (SWI), 53
notifType
unpack_voice_SUPSNotificationCallback_ind_t, 1348
voiceSUPSNotification, 1496
Notification
_SwiAvmsEventInd, 102
notification
omaDmNotificationsTlv, 628
unpack_omaDmNotificationsTlv_t, 1242
notification_Type
unpack_voice_USSDNotificationCallback_ind_t, 1348
notificationType
SMSEtwsMessage, 974
sMSEtwsMessageInfo, 975
notused
unpack_dms_SetCrashAction_t, 1128
nr5gBandPref, 619
bits_129_192, 619
bits_193_256, 619
bits_1_64, 619
bits_65_128, 619
num_instances
_qaQmi3GPP2BroadcastCfgInfo, 73
_qaQmi3GPPBroadcastCfgInfo, 73
custSettingList, 215
DMScustSettingList, 240
sms_qaQmi3GPP2BroadcastCfgInfo, 965
sms_qaQmi3GPPBroadcastCfgInfo, 965
numApp
slotInf, 939
slotInfo, 941
uim_slotInfo, 1066
numCrashes
CrashInfo, 195
crashInformation, 197
numEntries
CurrentImgList, 204
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1144
numFeatures
personalizationStatus, 794
uim_personalizationStatus, 1058
numFiles
registerRefresh, 875
uim_registerRefresh, 1062
NumFlows
unpack_qos_SLQSSetQosEventCallback_ind_t, 1263
numInstance
nas_operatorPLMNList, 511
nas_PLMNNetworkName, 518
operatorPLMNList, 630
PLMNNetworkName, 803
numInstances
arrAlertingPattern, 130
arrAlertingType, 131
arrAlphaID, 131
arrCalledPartyNum, 132
arrCallEndReason, 133
arrCallInfo, 133
arrConnectPartyNum, 134
arrDiagInfo, 134
arrRedirPartyNum, 135
arrRemotePartyName, 135
arrRemotePartyNum, 136
arrSvcOption, 136
arrUUSInfo, 137
DomainNameList, 245
getCallFWExtInfo, 276
getCallFWInfo, 277
getMsgWaitingInfo, 295
homeSIDNID, 323
msgWaitingInfo, 440
nas_homeSIDNID, 480
nas_roamIndList, 532
nas_SccRxInfo, 539
PCSCFFQDNAddressList, 788
PCSCFIPv4ServerAddressList, 789
roamIndList, 885
SccRxInfo, 899
sms_getMsgWaitingInfo, 960
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t, 1289
voice_arrAlertingPattern, 1400
voice_arrAlertingType, 1401
voice_arrAlphaID, 1402
voice_arrCalledPartyNum, 1402
voice_arrCallEndReason, 1403
voice_arrCallInfo, 1403
voice_arrConnectPartyNum, 1404
voice_arrDiagInfo, 1404
voice_arrRedirPartyNum, 1405
voice_arrRemotePartyName, 1405
voice_arrRemotePartyNum, 1406
voice_arrSvcOption, 1407

voice_arrUUSInfo, 1407
voice_getCallFWExtInfo, 1432
voice_getCallFWInfo, 1432
wds_DomainNameList, 1518
wds_PCSCFFQDNAddressList, 1522
wds_PCSCFIPv4ServerAddressList, 1523
numLen
calledPartyInfo, 147
callFWExtInfo, 152
callFWInfo, 153
callingPartyInfo, 156
peerNumberInfo, 793
redirNumInfo, 874
remotePartyNum, 877
voice_calledPartyInfo, 1409
voice_callFWExtInfo, 1414
voice_callFWInfo, 1415
voice_callingPartyInfo, 1418
voice_peerNumberInfo, 1436
voice_redirNumInfo, 1439
voice_remotePartyNum, 1441
numOfFiles
uim_refreshevent, 1061
UIMRefreshEvent, 1081
numOfRoutes
sms_setRoutesReq, 970
smsSetRoutesReq, 985
numOpt
DHCPOptionList, 229
wds_DHCPLeaseOptTlv, 1512
wds_DHCPv4OptionList, 1516
WdsDHCPv4OptionList, 1536
wdsDhcpv4OptionList, 1536
numPI
peerNumberInfo, 793
voice_peerNumberInfo, 1436
NumPilots
nas_PilotSetData, 516
PilotSetData, 800
numPlan
calledPartyInfo, 147
callFWExtInfo, 152
callingPartyInfo, 156
connectNumInfo, 194
peerNumberInfo, 793
redirNumInfo, 874
voice_calledPartyInfo, 1409
voice_callFWExtInfo, 1414
voice_callingPartyInfo, 1418
voice_connectNumInfo, 1424
voice_peerNumberInfo, 1436
voice_redirNumInfo, 1439
numPresInd
connectNumInfo, 194
voice_connectNumInfo, 1424
numQosFlow
sQosStat, 988
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
numRadioInterfaces
nas_servSystem, 541
servSystem, 908
NumRxFilters
unpack_qos_QosFlowInfo_t, 1258
numSI
peerNumberInfo, 793
voice_peerNumberInfo, 1436
numSlot
cardStatus, 158
uim_cardStatus, 1050
NumSupUSBComps
unpack_dms_GetUSBComp_t, 1125
NumTxFilters
unpack_qos_QosFlowInfo_t, 1258
numType
calledPartyInfo, 147
callFWExtInfo, 152
callingPartyInfo, 156
connectNumInfo, 194
peerNumberInfo, 794
redirNumInfo, 874
voice_calledPartyInfo, 1409
voice_callFWExtInfo, 1414
voice_callingPartyInfo, 1418
voice_connectNumInfo, 1424
voice_peerNumberInfo, 1436
voice_redirNumInfo, 1439
number
calledPartyInfo, 147
callFWExtInfo, 152
callFWInfo, 153
callingPartyInfo, 156
ECTNum, 251
peerNumberInfo, 793
redirNumInfo, 874
voice_calledPartyInfo, 1409
voice_callFWExtInfo, 1414
voice_callFWInfo, 1415
voice_callingPartyInfo, 1418
voice_ECTNum, 1429
voice_peerNumberInfo, 1436
voice_redirNumInfo, 1439
numberPlan
callFwdTypeAndPlan, 150
voice_callFwdTypeAndPlan, 1412
numberType
callFwdTypeAndPlan, 150
voice_callFwdTypeAndPlan, 1412
NxtHdrProto
unpack_qos_swiQosFilter_t, 1268

OKtoRefresh
pack_uim_SLQSUIMRefreshOK_t, 738
UIMRefreshOKReq, 1083
OMADMCancelSession
qaGobiApiOmadm.h, 1941
OMADMEnabled

unpack_swioma_SLQSOMADMGetSettings_t, 1297
OMADMGetPendingNIA
qaGobiApiOmadm.h, 1941
OMADMGetSessionInfo
qaGobiApiOmadm.h, 1942
OMADMStartSession
qaGobiApiOmadm.h, 1943
OTASPStatus
unpack_voice_OTASPStatusCallback_ind_t, 1313
voiceOTASPStatusInfo, 1479
oddEvenInd
calledPartySubAdd, 148
voice_calledPartySubAdd, 1410
OfflineReason
unpack_dms_GetPower_t, 1121
offset
readTransparentInfo, 872
uim_readTransparentInfo, 1059
oldPINLen
changeUIMPIN, 182
uim_changeUIMPIN, 1051
oldPINVal
changeUIMPIN, 182
uim_changeUIMPIN, 1051
oldPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 755
voiceSetCallBarringPwdInfo, 1483
oldValue
pack_dms_UIMChangePIN_t, 643
omaDmConfig
sessionInfo, 909
sessionInfoExt, 909
omaDmConfigTlv, 620
alertmsg, 621
alertmsglength, 621
state, 621
userInputReq, 621
userInputTimeout, 621
omaDmConfigTlvExt, 621
alertmsg, 623
alertmsglength, 623
state, 623
userInputReq, 623
userInputTimeout, 623
omaDmFota
sessionInfo, 909
sessionInfoExt, 909
omaDmFotaTlv, 623
description, 624
descriptionlength, 624
fwdloadsize, 624
fwloadComplete, 625
namelength, 625
package_name, 625
sessionType, 625
severity, 625
state, 625
updateCompleteStatus, 625
userInputReq, 625
userInputTimeout, 625
version, 625
versionlength, 625
omaDmFotaTlvExt, 625
description, 627
descriptionlength, 627
fumoResultCode, 627
namelength, 627
package_name, 627
packageSize, 627
receivedBytes, 627
reserved, 627
state, 627
userInputTimeout, 627
version, 627
versionlength, 627
omaDmNotifications
sessionInfo, 909
omaDmNotificationsTlv, 627
notification, 628
sessionStatus, 628
opMode
dmsSwiGetPCInfo, 243
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 661
Open Mobile Alliance Service (OMA), 49
operatingMode
dms_OperatingModeTlv, 231
OperatingModeTlv
unpack_dms_SetEventReport_ind_t, 1129
operation
depersonalizationInformation, 225
pack_pds_SetPDSDefaults_t, 703
uim_depersonalizationInformation, 1052
OperationMode
unpack_dms_GetPower_t, 1121
operatorNameString, 628
PLMNName, 628
OperatorPLMNData, 629
lac1, 629
lac2, 629
mcc, 629
mnc, 629
PLMNRecID, 629
operatorPLMNList, 629
numInstance, 630
PLMNData, 630
optCode
DHCPOption, 228
wds_DHCPOpt, 1514
wds_DHCPv4Option, 1515
WdsDHCPv4Option, 1535
wdsDhcpv4Option, 1535
optList

wds_DHCPLeaseOptTlv, 1512
optListData
wds_DHCPLeaseOptTlv, 1512
optVal
wds_DHCPv4Option, 1515
WdsDHCPv4Option, 1535
wdsDhcpv4Option, 1535
optValLen
DHCPOption, 228
wds_DHCPOpt, 1514
wds_DHCPv4Option, 1516
WdsDHCPv4Option, 1535
wdsDhcpv4Option, 1535
OriginateUSSD
qaGobiApiVoice.h, 2082
otaMsgTlv
NASQmiCbkNasSwiOTAMessageInd, 598

p1Status
unpack_dms_UIMGetPINStatus_t, 1153
p1UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1153
p1VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1153
p2Status
unpack_dms_UIMGetPINStatus_t, 1153
p2UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1153
p2VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1153
p3GPP2Pri
swiQosFlow, 1012
p3GPP2TimeInfo
GetNetworkTimeResp, 295
unpack_nas_SLQSGetNetworkTime_t, 1201
p3GPPImCn
swiQosFlow, 1012
p3GPPResResidualBER
swiQosFlow, 1013
p3GPPSigInd
swiQosFlow, 1013
p3GPPTimeInfo
GetNetworkTimeResp, 295
unpack_nas_SLQSGetNetworkTime_t, 1201
p3GPPTraHdlPri
swiQosFlow, 1013
p3GppNetworkInfoInstances
unpack_nas_PerformNetworkScan_t, 1196
p3GppNetworkInstanceSize
unpack_nas_PerformNetworkScan_t, 1196
p3gppRelease
_slqs3GPPConfigItem, 85
pack_wds_SLQSSet3GPPConfigItem_t, 778
pAAASPI
pack_wds_SetMobileIPProfile_t, 771
PACK_WDS_IPV4
wds.h, 2256
PACK_WDS_IPV6
wds.h, 2257

pAMRStatus
pack_voice_SLQSVoiceGetConfig_t, 751
voiceGetConfigReq, 1471
pAPN
_SLQSAVMSSettingsAPNInfo, 88
pAPNClass
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 368
Profile3GPP, 815
pAPNClass3GPP2
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pAPNDisabledFlag
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 368
Profile3GPP, 815
pAPNEnabled3GPP2
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pAPNInfo
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 87
pAPNName
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 368
Profile3GPP, 815
qmiWdsRunTimeSettings, 864
swiPDPRuntimeSettingsResp, 1007
pAPNnameSize
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 368
Profile3GPP, 815
PASSWORD_LENGTH
qaGobiApiVoice.h, 2081
pAVRXAVCHeadroom
RXAVCList, 890
pAVRXAVCSens
RXAVCList, 890
pAccelAcceptReady
QmiCbkLocSensorStreamingInd, 851
unpack_loc_SensorStreamingCallback_Ind_t, 1179
pAccelSamplesAccepted
QmiCbkLocInjectSensorDataInd, 843
unpack_loc_InjectSensorDataCallback_Ind_t, 1170
pAccelTempAcceptReady
QmiCbkLocSensorStreamingInd, 851
unpack_loc_SensorStreamingCallback_Ind_t, 1179
pAccelTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 843
unpack_loc_InjectSensorDataCallback_Ind_t, 1170
pAcceleroData
LocInjectSensorDataReq, 409

pAcceleroTempData
LocInjectSensorDataReq, 409
pAcceleroTimeSrc
LocInjectSensorDataReq, 409
pAccuracy
unpack_pds_GetPDSDefaults_t, 1243
pAcqOrder
acqOrderPref, 117
nas_acqOrderPref, 442
pAcqOrderPref
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
pActPilotPNElements
nas_NetworkStat1x, 505
NetworkStat1x, 614
pActiveTimer
dmsPsmCfgParamChange, 242
pack_dms_SLQSSetPowerSaveModeConfig_t, 639
PowerSaveModeConfigReq, 806
PowerSaveModeConfigResp, 807
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1140
pAddCDMASysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pAddGSMSysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pAddHDRSysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pAddLTESysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pAddWCDMASysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pAddrAllocPref
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 367
Profile3GPP, 815
pAddress
pack_wds_SetMobileIPProfile_t, 771
pAirTimer
pack_voice_SLQSVoiceGetConfig_t, 751
voiceGetConfigReq, 1471
pAirTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pAirTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 757
voiceSetConfigReq, 1485
pAirTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1341
voiceSetConfigResp, 1487
pAlertPriority
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 172
pAlertType
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1450
pAlertingPattern
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1450
pAllowLinger
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pAlphaID
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
pAlphaIDInfo
unpack_voice_SLQSOriginateUSSD_t, 1316
unpack_voice_SLQSVoiceDialCall_t, 1318
unpack_voice_SLQSVoiceGetCallBarring_t, 1323
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1324
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
unpack_voice_SLQSVoiceGetCallWaiting_t, 1329
unpack_voice_SLQSVoiceGetCLIP_t, 1330
unpack_voice_SLQSVoiceGetCLIR_t, 1331
unpack_voice_SLQSVoiceGetCNAP_t, 1333
unpack_voice_SLQSVoiceGetCOLP_t, 1334
unpack_voice_SLQSVoiceGetCOLR_t, 1336
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1340
unpack_voice_SLQSVoiceSetSUPSService_t, 1343
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
USSResp, 1395
voiceCallInfoResp, 1450
voiceCallResponseParams, 1453
voiceGetCallBarringResp, 1460
voiceGetCallFWResp, 1462
voiceGetCallWaitInfo, 1463
voiceGetCLIPResp, 1465
voiceGetCLIRResp, 1466
voiceGetCNAPResp, 1467
voiceGetCOLPResp, 1468
voiceGetCOLRResp, 1470
voiceSetCallBarringPwdResp, 1484
voiceSetSUPSServiceResp, 1491

voiceSUPSInfo, 1494
pAlphaIdentifier
USSDNoWaitIndicationInfo, 1393
pAltitudeAssumed
QmiCbkLocPositionReportInd, 849
unpack_loc_PositionRpt_Ind_t, 1177
pAltitudeSrcInfo
LocInjectPositionReq, 407
pAltitudeWrtEllipsoid
LocInjectPositionReq, 407
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 849
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1177
pAltitudeWrtMeanSeaLevel
LocInjectPositionReq, 407
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 849
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1177
pAltitudeWrtSealevel
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
pAmrMode
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pAmrOctetAligned
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pAmrWBMode
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pAmrWBOctetAligned
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pAmrWbEnable
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pApnString
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pApnStringSize
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pApnname
pack_wds_SetDefaultProfile_t, 767
pAppName
loc_LocApplicationInfo, 392
LocApplicationInfo, 400
pAppPriority
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pAppProvider
loc_LocApplicationInfo, 392
LocApplicationInfo, 400
pAppSubType
HDRProtSubtypResp, 316
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1232
pAppType
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pAppVersion
loc_LocApplicationInfo, 392
LocApplicationInfo, 400
pApplication
pack_nas_SetCDMANetworkParameters_t, 667
pApplicationInfo
LOCStartReq, 412
pack_loc_Start_t, 664
pArrAlertingPattern
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrAlertingType
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrAlphaID
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrCallEndReason
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrCallInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
pArrCalledPartyNum
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrConnectPartyNum
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrDiagInfo
unpack_voice_allCallStatusCallback_ind_t, 1311

unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrRedirPartyNum
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrRemotePartyName
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrRemotePartyNum
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrSvcOption
unpack_voice_allCallStatusCallback_ind_t, 1311
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
voiceSetAllCallStatusCbkInfo, 1482
pArrUUSInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
voiceGetAllCallInfo, 1458
pAuth
pack_wds_SLQSStartDataSession_t, 782
pAuthPassword
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 820
pAuthPassword_tSize
LibPackprofile_3GPP2, 373
pAuthPasswordSize
LibpackProfile3GPP2, 362
Profile3GPP2, 821
pAuthProtocol
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 821
pAuthRetryCount
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 821
pAuthTimeout
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 821
pAuthenticateResult
UIMAuthenticateResp, 1070
unpack_uim_SLQSUIMAuthenticate_t, 1302
pAuthentication
qmiWdsRunTimeSettings, 864
ssdatasession_params, 992
pAuthenticationPref
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 368
Profile3GPP, 815
pAutoAnsStatus
unpack_voice_SLQSVoiceSetConfig_t, 1341
voiceSetConfigResp, 1487
pAutoAnswer
pack_voice_SLQSVoiceGetConfig_t, 751
pack_voice_SLQSVoiceSetConfig_t, 757
voiceGetConfigReq, 1471
voiceSetConfigReq, 1486
pAutoAnswerStat
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pAutoConnect
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 87
pAutoReboot
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 87
pAutoSelection
UIMGetConfigurationResp, 1074
unpack_uim_SLQSUIMGetConfiguration_t, 1304
pAutosdm
_SLQSOMADMSettings, 95
_SLQSOMADMSettingsReqParams, 96
_SLQSOMADMSettingsReqParams3, 97
pack_swioma_SLQSOMADMSetSettings_t, 729
pBandPref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pBdsSVInfo
LocDelAssDataReq, 401
pack_loc_Delete_Assist_Data_t, 649
pBearerID
QosFlowInfo, 869
pBearerId
swiPDPRuntimeSettingsResp, 1007
pBearerTech
DataBearerTechExt, 218
pBroadcast
pack_nas_SetCDMANetworkParameters_t, 667
pBroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 723
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1282
pBurstDTMFLengths
pack_voice_SLQSVoiceBurstDTMF_t, 745
voiceBurstDTMFInfo, 1448
pCCResType
unpack_voice_SLQSVoiceGetCallBarring_t, 1323
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1324
unpack_voice_SLQSVoiceGetCallWaiting_t, 1329
unpack_voice_SLQSVoiceGetCLIP_t, 1330
unpack_voice_SLQSVoiceGetCLIR_t, 1331
unpack_voice_SLQSVoiceGetCNAP_t, 1333

unpack_voice_SLQSVoiceGetCOLP_t, 1334
unpack_voice_SLQSVoiceGetCOLR_t, 1336
unpack_voice_SLQSVoiceSetCallBarringPassword_t, 1340
voiceGetCallBarringResp, 1460
voiceGetCallFWResp, 1462
voiceGetCallWaitInfo, 1464
voiceGetCLIPResp, 1465
voiceGetCLIRResp, 1466
voiceGetCNAPResp, 1467
voiceGetCOLPResp, 1469
voiceGetCOLRResp, 1470
voiceSetCallBarringPwdResp, 1484
pCCResultType
unpack_voice_SLQSVoiceDialCall_t, 1318
unpack_voice_SLQSVoiceSetSUPSService_t, 1343
voiceCallResponseParams, 1453
voiceSetSUPSServiceResp, 1491
pCCSUPSType
unpack_voice_SLQSVoiceDialCall_t, 1318
unpack_voice_SLQSVoiceGetCallBarring_t, 1323
unpack_voice_SLQSVoiceGetCallForwardingStatus_t, 1324
unpack_voice_SLQSVoiceGetCallWaiting_t, 1329
unpack_voice_SLQSVoiceGetCLIP_t, 1330
unpack_voice_SLQSVoiceGetCLIR_t, 1332
unpack_voice_SLQSVoiceGetCNAP_t, 1333
unpack_voice_SLQSVoiceGetCOLP_t, 1334
unpack_voice_SLQSVoiceGetCOLR_t, 1336
unpack_voice_SLQSVoiceSetCallBarringPassword_t, 1340
unpack_voice_SLQSVoiceSetSUPSService_t, 1343
voiceCallResponseParams, 1453
voiceGetCallBarringResp, 1460
voiceGetCallFWResp, 1462
voiceGetCallWaitInfo, 1464
voiceGetCLIPResp, 1465
voiceGetCLIRResp, 1466
voiceGetCNAPResp, 1467
voiceGetCOLPResp, 1469
voiceGetCOLRResp, 1470
voiceSetCallBarringPwdResp, 1484
voiceSetSUPSServiceResp, 1491
pCCSuppsType
unpack_voice_SLQSOriginateUSSD_t, 1316
USSResp, 1395
pCDMABroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 723
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1282
pCDMAChannel
nasGet3GPP2SubscriptionInfoResp, 569
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1220
pCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pCDMAECIOThreshList
CDMAECIOThresh, 165
nas_CDMAECIOThresh, 446
pCDMAFrameErrRate
GetErrRateResp, 284
unpack_nas_SLQSGetErrorRate_t, 1200
pCDMAInfo
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1221
pCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pCDMARSSIThreshList
CDMARSSIThresh, 174
nas_CDMARSSIThresh, 448
pCDMASSInfo
nasGetSigInfoResp, 574
pCDMASigInfo
nasSigInfo, 603
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1227
pCDMASrvStatusInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pCDMASysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pCLIPResp
unpack_voice_SLQSVoiceGetCLIP_t, 1330
voiceGetCLIPResp, 1465
pCLIPstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pCLIRCause
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pCLIRResp
unpack_voice_SLQSVoiceGetCLIR_t, 1332
voiceGetCLIRResp, 1466
pCLIRType
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
pCLIRstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346

voiceSUPSInfo, 1494
PCMparams, 787
iFaceTab, 787
iFaceTabLen, 787
pCNAPResp
unpack_voice_SLQSVoiceGetCNAP_t, 1333
voiceGetCNAPResp, 1467
pCNAPstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pCOLPResp
unpack_voice_SLQSVoiceGetCOLP_t, 1334
voiceGetCOLPResp, 1469
pCOLPstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pCOLRResp
unpack_voice_SLQSVoiceGetCOLR_t, 1336
voiceGetCOLRResp, 1470
pCOLRstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
PCSCFAddrPCO
unpack_wds_SLQSGetRuntimeSettings_t, 1378
PCSCFFQDNAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1378
PCSCFFQDNAddress, 788
fqdnAddr, 788
fqdnLen, 788
PCSCFFQDNAddressList, 788
numInstances, 788
pcsfFQDNAddress, 788
PCSCFIPv4ServerAddressList, 789
numInstances, 789
pscsfIPv4Addr, 789
pCSCFPortName
imsRegMgrConfigInfo, 338
SetRegMgrConfigReq, 927
pCSCFPortNameLen
SetRegMgrConfigReq, 927
pCSGID
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
pCUGIndex
unpack_voice_SUPSNotificationCallback_ind_t, 1348
voiceSUPSNotification, 1496
pCUGInfo
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
pCallBarPasswd
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pCallBarringPasswd
pack_voice_SLQSVoiceSetSUPSService_t, 759
voiceSetSUPSServiceReq, 1490
pCallEndReason
getDUNCallInfoResp, 282
pCallFWNum
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pCallFWTimerVal
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pCallForwardingNumber
pack_voice_SLQSVoiceSetSUPSService_t, 759
voiceSetSUPSServiceReq, 1490
pCallFwdInfo
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pCallFwdTypeAndPlan
pack_voice_SLQSVoiceSetSUPSService_t, 759
voiceSetSUPSServiceReq, 1490
pCallID
burstDTMFInfo, 144
pack_voice_SLQSVoiceManageCalls_t, 753
pack_voice_SLQSVoiceSendFlash_t, 754
pack_voice_SLQSVoiceStartContDTMF_t, 760
unpack_voice_SLQSVoiceBurstDTMF_t, 1317
unpack_voice_SLQSVoiceDialCall_t, 1318
unpack_voice_SLQSVoiceGetCallBarring_t, 1323
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1324
unpack_voice_SLQSVoiceGetCallWaiting_t, 1329
unpack_voice_SLQSVoiceGetCLIP_t, 1330
unpack_voice_SLQSVoiceGetCLIR_t, 1331
unpack_voice_SLQSVoiceGetCNAP_t, 1333
unpack_voice_SLQSVoiceGetCOLP_t, 1334
unpack_voice_SLQSVoiceGetCOLR_t, 1336
unpack_voice_SLQSVoiceSendFlash_t, 1339
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1340
unpack_voice_SLQSVoiceSetSUPSService_t, 1343
unpack_voice_SLQSVoiceStartContDTMF_t, 1343
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voice_burstDTMFInfo, 1408
voiceCallResponseParams, 1453
voiceContDTMFinfo, 1454
voiceFlashInfo, 1455
voiceGetCallBarringResp, 1460
voiceGetCallFWResp, 1462
voiceGetCallWaitInfo, 1464
voiceGetCLIPResp, 1465
voiceGetCLIRResp, 1466
voiceGetCNAPResp, 1467
voiceGetCOLPResp, 1468

voiceGetCOLRResp, 1470
voiceManageCallsReq, 1477
voiceSetCallBarringPwdResp, 1484
voiceSetSUPSServiceResp, 1491
voiceSUPSInfo, 1494
pCallId
pack_voice_SLQSVoiceAnswerCall_t, 744
pack_voice_SLQSVoiceEndCall_t, 747
unpack_voice_SLQSOriginateUSSD_t, 1316
unpack_voice_SLQSVoiceAnswerCall_t, 1317
unpack_voice_SLQSVoiceEndCall_t, 1319
USSResp, 1395
voiceAnswerCall, 1447
pCallInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1450
pCallPartySubAdd
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
pCallType
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
pCallWaitInd
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pCallbackAddr
cdmaMsgEncodingParams, 174
pCallbkAddr
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 172
pCallbkAddrLength
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 172
pCalledPartyInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pCallerIDInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pCallerNameInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pCallingPartyInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pCardResult
UIMAuthenticateResp, 1070
UIMGetFileAttributesResp, 1075
UIMReadTransparentResp, 1079
unpack_uim_ReadTransparent_t, 1300
unpack_uim_SLQSUIMAuthenticate_t, 1302
unpack_uim_SLQSUIMGetFileAttributes_t, 1305
pCardStatus
UIMGetCardStatusResp, 1072
unpack_uim_GetCardStatus_t, 1299
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t, 1307
pCcResultType
unpack_voice_SLQSOriginateUSSD_t, 1316
USSResp, 1395
pCellDataMask
pack_pds_ResetPDSData_t, 701
pCellDb
LocDelAssDataReq, 401
pack_loc_Delete_Assist_Data_t, 649
pChangeDuration
nasInitNetworkReg, 582
pack_nas_SLQSInitiateNetworkRegistration_t, 671
pChannelRate
getDUNCallInfoResp, 282
pChgDuration
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
pCiotAcqOrder
ciotAcqOrderPref, 184
nas_ciotAcqOrderPref, 454
pCiotAcqOrderPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
sysSelectPrefParamsExt, 1022
pCiotLteOpMode
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
sysSelectPrefParamsExt, 1022
pCiotLteOpModePref
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
sysSelectPrefParamsExt, 1022
pClkInfo
LocDelAssDataReq, 401
pack_loc_Delete_Assist_Data_t, 649
pCodecSTGain
GetAudioPathConfigResp, 272
SetAudioPathConfigReq, 911
pColorCode
nasGetHDRColorCodeResp, 570
unpack_nas_SLQSNasGetHDRColorCode_t, 1223
pConfidence
LocSetCradleMountReq, 410
pConfigAltitudeAssumed
LOCStartReq, 412
pack_loc_Start_t, 664
pConfigurationMask
pack_uim_SLQSUIMGetConfiguration_t, 735
UIMGetConfigurationReq, 1073
pConnectNumInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceCallInfoResp, 1450
voiceInfoRec, 1476
pConnectionRetryTimers
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 87
pConnectionStatus
getDUNCallInfoResp, 282
pContextId
swiPDPRuntimeSettingsResp, 1007
pCrashInfo

CrashInfoParams, 196
pCrashString
CrashInfo, 195
pCreateProfileOut
unpack_wds_SLQSCreateProfile_t, 1366
pCsgId
nasNetworkReject, 587
pCurAMRConfig
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pCurDataBearerTechnology
dataBearers, 215
pCurPrefVoiceSO
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pCurProfile
pack_wds_SLQSCreateProfile_t, 772
pCurVoiceDomainPref
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pCurVoicePrivacyPref
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pCurrChannelRateInd
pack_wds_SLQSWdsSetEventReport_t, 784
wdsSetEventReportReq, 1545
pCurrDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 784
wdsSetEventReportReq, 1545
pCurrImgInfo
CurrentImgList, 204
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1144
pCurrNetworkInfo
CurrDataSysStat, 203
pCurrPrefDataSysInd
pack_wds_SLQSWdsSetEventReport_t, 784
wdsSetEventReportReq, 1545
pCurrTTYMode
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pCurrentChannelRXRate
WdsConnectionRateElmnts, 1532
pCurrentChannelTXRate
WdsConnectionRateElmnts, 1532
pCurrentPersonality
HDRPersonalityInd, 315
HDRPersonalityResp, 315
nas_HDRPersonality_Ind_Data, 475
unpack_nas_SLQSSwiGetHDRPersonality_t, 1231
pCurrentPrsnlty
HDRProtSubtypResp, 316
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1232
pCurrentmitigationLvl
TmdGetMitigationLvlResp, 1035
pCustSettingInfo
DMSgetCustomFeatureV2, 240
getCustomFeatureV2, 277
pCustSettingList
DMSgetCustomFeatureV2, 240
getCustomFeatureV2, 277
pCustomSCP
pack_nas_SetCDMANetworkParameters_t, 667
pCwtMute
SetM2MAudioProfileReq, 923
SetM2MAVMuteReq, 924
pCycleLen
nasGeteDRXParamsResp, 570
nasSeteDRXParamsReq, 602
pack_nas_SLQSNASSeteDRXParams_t, 681
unpack_nas_SLQSNASGeteDRXParams_t, 1222
pDDTMInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 679
pDHCPRelayEnabled
custFeaturesInfo, 211
custFeaturesSetting, 213
PDOP
loc_precisionDilution, 393
precisionDilution_s, 807
PDPType
unpack_wds_SLQSGetRuntimeSettings_t, 1378
pDRCParams
GetHRPDStatsResp, 285
unpack_nas_SLQSSwiGetHRPDStats_t, 1233
PDS_SRV
qaGobiApiCbk.h, 1726
PDSInjectTimeReference
qaGobiApiPds.h, 1949
PDSPosMethodStateReq, 791
pWifiState, 792
pXtraDataState, 792
pXtraTimeState, 792
PDSPositionData, 789
pAltitudeWrtEllipsoid, 791
pAltitudeWrtSealevel, 791
pHorizontalConfidence, 791
pHorizontalUncCircular, 791
pLatitude, 791
pLongitude, 791
pPositionSource, 791
pTimeStamp, 791
pTimeType, 791
pVerticalConfidence, 791
pVerticalUnc, 791
pDTMFTXGain
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 911
pDataBearer
QosEventInfo, 868
slqsWdsEventInfo, 957
unpack_wds_GetDataBearerTechnology_t, 1354
pDataBearerTech
getDUNCallInfoResp, 282
pDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 784

wdsSetEventReportReq, 1545
pDataCallStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 784
wdsSetEventReportReq, 1545
pDataMode
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 821
pDataRate
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 821
swiQosFlow, 1013
pDataSrc
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pDataStatusDetail
NetworkDebugResp, 612
unpack_nas_SLQSSwiNetworkDebug_t, 1235
pDataSystemStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 784
wdsSetEventReportReq, 1545
pDate
_SLQSOMADMSessionInfo, 93
pDateLength
_SLQSOMADMSessionInfo, 93
pDayltSavAdj
nasNetworkTime, 588
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1226
pDefaultPDNEnabled
_slqs3GPPConfigItem, 85
pack_wds_SLQSSet3GPPConfigItem_t, 778
pDeferTime
pack_swioma_SLQSOMADMSendSelection_t, 728
pDesription
QmiNas3GppNetworkInfo, 857
pDestAddr
cdmaMsgEncodingParams, 174
wcdmaMsgEncodingParams, 1502
pDestSMSContent
getDyingGaspCfg, 283
pack_dms_SLQSSwiSetDyingGaspCfg_t, 640
packgetDyingGaspCfg, 786
setDyingGaspCfg, 916
pDestSMSNum
getDyingGaspCfg, 283
pack_dms_SLQSSwiSetDyingGaspCfg_t, 640
packgetDyingGaspCfg, 786
setDyingGaspCfg, 916
pDetachAction
pack_nas_SLQSSwiPSDetach_t, 700
PSDetachReq, 824
pDevCrashStatus
CrashInfoParams, 196
pDeviceConfigDetail
NetworkDebugResp, 612
unpack_nas_SLQSSwiNetworkDebug_t, 1235
pDiagInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1450
pDigitBuff
burstDTMFInfo, 144
voice_burstDTMFInfo, 1408
pDirNum
nasGet3GPP2SubscriptionInfoResp, 569
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1220
pDisableIMSI
custFeaturesInfo, 211
pDispInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pDisplayMode
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 172
pDomainList
qmiWdsRunTimeSettings, 864
pDormancyStatus
getDUNCallInfoResp, 282
slqsWdsEventInfo, 957
pDormancyStatusInd
pack_wds_SLQSWdsSetEventReport_t, 785
wdsSetEventReportReq, 1545
pDualStandByPrefInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 679
pDuration
unpack_pds_GetXTRAValidity_t, 1247
pDurationDueToOOS
PowerSaveModeConfigResp, 807
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1140
pDurationThreshold
PowerSaveModeConfigResp, 807
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1140
pECIOThresList
ECIOThresh, 250
nas_ECIOThresh, 465
pECIOThresh
pack_nas_SLQSConfigSigInfo_t, 670
sigInfo, 936
pECMode
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pECTNum
unpack_voice_SUPSNotificationCallback_ind_t, 1348
voiceSUPSNotification, 1496
PER
nas_NetworkStatEVDO, 507
NetworkStatEVDO, 616
pESNString

serialNumbersInfo, 904
pEVDOPageMonPerChangeInd
pack_wds_SLQSWdsSetEventReport_t, 785
wdsSetEventReportReq, 1545
pEarMute
SetM2MAudioProfileReq, 923
pEarlyWakeupTime
PowerSaveModeConfigResp, 807
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1140
pEdrxEnable
nasGeteDRXParamsResp, 570
nasSeteDRXParamsReq, 602
pack_nas_SLQSNASSeteDRXParams_t, 681
unpack_nas_SLQSNASGeteDRXParams_t, 1222
pEhrpdMTUSize
SwiDmsGetMTUResp, 999
unpack_swidms_SLQSSwiDmsGetMTU_t, 1290
pEmerMode
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pEmergencyCategory
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
pEnableNotification
pack_wds_DHCPv4ClientLeaseChange_t, 761
WdsClientLeaseChange, 1530
pEnableState
dmsPsmCfgParamChange, 242
pEnabled
pack_wds_SetMobileIPProfile_t, 771
pEnabledStatus
unpack_pds_GetPDSState_t, 1244
pEncodingAlphabet
cdmaMsgEncodingParams, 174
pEncryptData
pack_uim_ReadTransparent_t, 732
UIMReadTransparentReq, 1078
pEncryptedData
UIMReadTransparentResp, 1079
unpack_uim_ReadTransparent_t, 1300
pEncryptedPIN1
pack_uim_VerifyPin_t, 741
UIMPinResp, 1076
UIMVerifyPinReq, 1090
unpack_uim_ChangePin_t, 1298
unpack_uim_SetPinProtection_t, 1301
unpack_uim_UnblockPin_t, 1307
unpack_uim_VerifyPin_t, 1308
pError
USSDNoWaitIndicationInfo, 1393
pErrorCodeStr
imsaRatStatusInfo, 330
pErrorMask
CATEventDataType, 161
pErrorRateInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pEspSpi
swiQosFilter, 1009
pEtwsMessageInfo
SMSEventInfo_s, 978
pEtwsPlmnInfo
SMSEventInfo_s, 978
pExtDispInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pExtDispRecInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pExtErrCode
_GetProfileSettingOut, 67
UnPackGetProfileSettingOut, 1388
pExtErrorCode
CreateProfileOut, 199
ModifyProfileOut, 439
unpack_wds_SLQSModifyProfile_t, 1379
pFOTAUpdate
_SLQSOMADMSettings, 95
pFOTAdownload
_SLQSOMADMSettings, 95
pFailCause
unpack_voice_SLQSVoiceGetCallBarring_t, 1323
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1324
unpack_voice_SLQSVoiceGetCallWaiting_t, 1329
unpack_voice_SLQSVoiceGetCLIP_t, 1330
unpack_voice_SLQSVoiceGetCLIR_t, 1332
unpack_voice_SLQSVoiceGetCNAP_t, 1333
unpack_voice_SLQSVoiceGetCOLP_t, 1334
unpack_voice_SLQSVoiceGetCOLR_t, 1336
unpack_voice_SLQSVoiceManageCalls_t, 1338
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1340
unpack_voice_SLQSVoiceSetSUPSService_t, 1343
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceGetCallBarringResp, 1460
voiceGetCallFWResp, 1462
voiceGetCallWaitInfo, 1464
voiceGetCLIPResp, 1465
voiceGetCLIRResp, 1466
voiceGetCNAPResp, 1467
voiceGetCOLPResp, 1469
voiceGetCOLRResp, 1470
voiceManageCallsResp, 1478
voiceSetCallBarringPwdResp, 1484
voiceSetSUPSServiceResp, 1491
voiceSUPSInfo, 1494
pFailErrorCode
imsaPdpStatusInfo, 329
pFailureCause

USSDNoWaitIndicationInfo, 1393
pFailureReason
unpack_wds_SLQSStartDataSession_t, 1385
pFile
ERIFileparams, 253
pFileAttributes
UIMGetFileAttributesResp, 1075
unpack_uim_SLQSUIMGetFileAttributes_t, 1305
pFileSize
ERIFileparams, 253
pFixId
QmiCbkLocPositionReportInd, 849
unpack_loc_PositionRpt_Ind_t, 1178
pFlag
RXPCMIIRFltr, 893
TXPCMIIRFltr, 1043
pFlashPayLd
pack_voice_SLQSVoiceSendFlash_t, 754
voiceFlashInfo, 1455
pFlashType
pack_voice_SLQSVoiceSendFlash_t, 754
voiceFlashInfo, 1455
pFollowOnDC
slqssendasyncsmsparams_s, 948
sms_sendAsyncsmsParams, 968
pForbidden
QmiNas3GppNetworkInfo, 857
pForceOnDC
slqssendasyncsmsparams_s, 948
sms_sendAsyncsmsParams, 968
pForceRev0
pack_nas_SetCDMANetworkParameters_t, 667
pFwAutoCheck
_SLQSOMADMSettings, 95
_SLQSOMADMSettingsReqParams3, 97
pack_swioma_SLQSOMADMSetSettings_t, 729
pFwAutoSDM
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 87
pFwPromptUpdate
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 87
pFwPromptdownload
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 87
pGCDumpString
CrashInfo, 195
pGERANInfo
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1221
pGPRSGrantedQoS
qmiWdsRunTimeSettings, 864
pGPRSMinimumQoS
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 368
Profile3GPP, 815
pGPRSRequestedQos
LibpackProfile3GPP, 356
LibPackprofile_3GPP, 368
Profile3GPP, 815
pGPSDataMask
pack_pds_ResetPDSData_t, 701
pGPSEnable
custFeaturesSetting, 213
pGPSLPM
custFeaturesInfo, 211
custFeaturesSetting, 213
pGPSSel
custFeaturesInfo, 211
custFeaturesSetting, 213
pGPSWeek
unpack_pds_GetXTRAValidity_t, 1247
pGPSWeekOffset
unpack_pds_GetXTRAValidity_t, 1247
pGSMBER
GetErrRateResp, 284
unpack_nas_SLQSGetErrorRate_t, 1200
pGSMCallBarringSysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pGSMCipherDomainSysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pGSMRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pGSMRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pGSMRSSIThreshList
GSMRSSIThresh, 308
nas_GSMRSSIThresh, 470
pGSMSSInfo
nasGetSigInfoResp, 574
pGSMSigInfo
nasSigInfo, 603
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1227
pGSMSrvStatusInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pGSMSysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pGWAcqOrderPref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 112

pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pGWAddressV4
qmiWdsRunTimeSettings, 864
pGenerator
GetM2MAudioProfileReq, 290
SetM2MAudioProfileReq, 923
pGetCallFWExtInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1324
voiceGetCallFWResp, 1462
pGetCallFWInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1324
voiceGetCallFWResp, 1462
pGetCustomInput
DMSgetCustomFeatureV2, 240
getCustomFeatureV2, 277
pGetDyingGaspCfg
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1142
pGetDyingGaspStatistics
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1143
pGetIndicationRegInfo
unpack_sms_SLQSGetIndicationRegister_t, 1280
pGetMsgWaitingInfoResp
unpack_sms_SLQSGetMessageWaiting_t, 1281
pGetTransLayerInfo
unpack_sms_SLQSGetTransLayerInfo_t, 1283
pGetTransNWRegInfo
unpack_sms_SLQSGetTransNWRegInfo_t, 1283
pGnssData
LocDelAssDataReq, 401
pack_loc_Delete_Assist_Data_t, 650
pGpsTime
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pGyroAcceptReady
QmiCbkLocSensorStreamingInd, 851
unpack_loc_SensorStreamingCallback_Ind_t, 1179
pGyroData
LocInjectSensorDataReq, 409
pGyroSamplesAccepted
QmiCbkLocInjectSensorDataInd, 843
unpack_loc_InjectSensorDataCallback_Ind_t, 1170
pGyroTempAcceptReady
QmiCbkLocSensorStreamingInd, 851
unpack_loc_SensorStreamingCallback_Ind_t, 1179
pGyroTempData
LocInjectSensorDataReq, 409
pGyroTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 843
unpack_loc_InjectSensorDataCallback_Ind_t, 1170
pGyroTimeSrc
LocInjectSensorDataReq, 409
pHA2002bis
pack_wds_SetMobileIPParameters_t, 769
pHAAuthenticator
pack_wds_SetMobileIPParameters_t, 769
pHASPI
pack_wds_SetMobileIPProfile_t, 771
pHDRECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pHDRECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pHDRECIOThreshList
HDRECIOThresh, 314
nas_HDRECIOThresh, 474
pHDRIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pHDRIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pHDRIOThreshList
HDRIOThresh, 314
nas_HDRIOThresh, 475
pHDRNewUATIAssInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pHDRPackErrRate
GetErrRateResp, 284
unpack_nas_SLQSGetErrorRate_t, 1200
pHDRPersInd
unpack_nas_SLQSSwiHDRPersonalityCallback_-Ind_t, 1234
pHDRRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pHDRRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pHDRRSSIThreshList
HDRRSSIThresh, 317
nas_HDRRSSIThresh, 476
pHDRSINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pHDRSINRThresList
HDRSINRThresh, 317
nas_HDRSINRThresh, 477
pHDRSINRThresh
pack_nas_SLQSConfigSigInfo_t, 670
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
sigInfo, 936
pHDRSINRThreshList

HDRSINRThreshold, 318
nas_HDRSINRThreshold, 478
pHDRSSInfo
nasGetSigInfoResp, 574
pHDRSessionCloseInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pHDRSigInfo
nasSigInfo, 603
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1227
pHDRSrvStatusInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pHDRSysInfo
nasGetSysInfoResp, 576
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pHaltSubscription
UIMGetConfigurationResp, 1074
unpack_uim_SLQSUIMGetConfiguration_t, 1304
pHeading
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pHeadingUnc
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pHomeSIDNID
nasGet3GPP2SubscriptionInfoResp, 569
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1220
pHorCirConf
QmiCbkLocBestAvailPosInd, 838
unpack_loc_BestAvailPos_Ind_t, 1162
pHorConfidence
LocInjectPositionReq, 407
QmiCbkLocPositionReportInd, 850
unpack_loc_PositionRpt_Ind_t, 1178
pHorEllpConf
QmiCbkLocBestAvailPosInd, 838
unpack_loc_BestAvailPos_Ind_t, 1162
pHorReliability
LocInjectPositionReq, 408
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pHorUncCircular
LocInjectPositionReq, 408
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pHorUncEllipseOrientAzimuth
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pHorUncEllipseSemiMajor
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pHorUncEllipseSemiMinor
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pHorizontalAccuracyLvl
LOCStartReq, 412
pack_loc_Start_t, 664
pHorizontalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
pHorizontalUncCircular
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
pHotSwapStatus
UIMGetCardStatusResp, 1072
unpack_uim_GetCardStatus_t, 1299
pHrpdMTUSize
SwiDmsGetMTUResp, 999
unpack_swidms_SLQSSwiDmsGetMTU_t, 1290
pHwConfig
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 781
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1383
WdsDHCPv4Config, 1533
PI
calledPartyInfo, 148
callerIDInfo, 149
callFWExtInfo, 152
callingPartyInfo, 156
redirNumInfo, 874
voice_calledPartyInfo, 1410
voice_callerIDInfo, 1411
voice_callFWExtInfo, 1414
voice_callingPartyInfo, 1418
voice_redirNumInfo, 1439
PIFACEId
SetM2MAudioAVCFGReq, 921
pIMCNflag
qmiWdsRunTimeSettings, 864
pIMEIString
serialNumbersInfo, 904
pIMSDomain
GetIMSUserConfigParams, 287
imsUserConfigInfo, 340
SetIMSUserConfigReq, 917

pIMSDomainLen
GetIMSUserConfigParams, 287
SetIMSUserConfigReq, 917
pIMSTestMode
GetRegMgrConfigParams, 296
imsRegMgrConfigInfo, 338
SetRegMgrConfigReq, 927
pIOThresList
IOThresh, 345
nas_IOThresh, 482
pIOThresh
pack_nas_SLQSConfigSigInfo_t, 670
sigInfo, 936
pIPAddressV4
QmiWdsIpAddressInfo, 861
qmiWdsRunTimeSettings, 864
pIPAddressV6
QmiWdsIpAddressInfo, 861
pIPFamSupport
custFeaturesInfo, 211
pIPFamily
GetInstIDResp, 289
pIPFamilyPreference
qmiWdsRunTimeSettings, 864
pIPV6AddrInfo
qmiWdsRunTimeSettings, 864
pIPV6GWAddrInfo
qmiWdsRunTimeSettings, 864
pIPv4Addr
WdsDHCPv4ClientLeaseInd, 1532
pIPv4AddrInfo
QmiCbkLocGetServerInd, 841
unpack_loc_GetServer_Ind_t, 1168
pIPv4AddrPref
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pIPv4Address
swiPDPRuntimeSettingsResp, 1007
pIPv4Config
LocSetServerReq, 411
pack_loc_SLQSLOCSetServer_t, 663
pIPv4DstAddr
swiQosFilter, 1009
pIPv4GWAddress
swiPDPRuntimeSettingsResp, 1007
pIPv4SrcAddr
swiQosFilter, 1009
pIPv6AddPref
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pIPv6AddrInfo
QmiCbkLocGetServerInd, 841
unpack_loc_GetServer_Ind_t, 1168
pIPv6Address
swiPDPRuntimeSettingsResp, 1007
pIPv6Config
LocSetServerReq, 411
pack_loc_SLQSLOCSetServer_t, 663
pIPv6DstAddr
swiQosFilter, 1009
pIPv6GWAddress
swiPDPRuntimeSettingsResp, 1007
pIPv6Label
swiQosFilter, 1009
pIPv6SrcAddr
swiQosFilter, 1009
pIPv6TrafCls
swiQosFilter, 1009
pIPv6prefixlen
QmiWdsIpAddressInfo, 861
pId
swiQosFilter, 1009
pIds
swiQosIds, 1014
pIgnoreHotSwapSwitch
pack_uim_SLQSUIMPowerUp_t, 737
UIMPowerUpReq, 1077
pImCnFlag
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pImageList
pack_fms_SetImagesPreference_t, 649
unpack_fms_GetImagesPreference_t, 1156
pImeiSvnString
serialNumbersInfo, 904
pImgType
FirmwareUpdatStat, 261
pImsRegErrCode
IMSARegistrationStatus, 331
pImsRegStatus
IMSARegistrationStatus, 331
imsaRegStatusInfo, 331
pInUse
QmiNas3GppNetworkInfo, 857
pIndFieldsList
IMSASupportedFieldsResp, 334
pIndicationToken
pack_uim_ChangePin_t, 731
pack_uim_ReadTransparent_t, 732
pack_uim_SetPinProtection_t, 733
pack_uim_SLQSUIMAuthenticate_t, 734
pack_uim_SLQSUIMGetFileAttributes_t, 736
pack_uim_UnblockPin_t, 740
pack_uim_VerifyPin_t, 741
UIMAuthenticateReq, 1069
UIMAuthenticateResp, 1070
UIMChangePinReq, 1070
UIMGetFileAttributesReq, 1074
UIMGetFileAttributesResp, 1075
UIMPinResp, 1076
UIMReadTransparentReq, 1078
UIMReadTransparentResp, 1079
UIMSetPinProtectionReq, 1085

UIMUnblockPinReq, 1089
UIMVerifyPinReq, 1090
unpack_uim_ChangePin_t, 1298
unpack_uim_ReadTransparent_t, 1300
unpack_uim_SetPinProtection_t, 1301
unpack_uim_SLQSUIMAuthenticate_t, 1302
unpack_uim_SLQSUIMGetFileAttributes_t, 1305
unpack_uim_UnblockPin_t, 1307
unpack_uim_VerifyPin_t, 1308
pInfo
pack_voice_AnswerUSSD_t, 742
pack_voice_OriginateUSSD_t, 742
pInstanceID
GetInstIDResp, 289
pInstanceSize
nas_RfBandInfoList, 528
QmiNasPerformNetworkScanResp, 858
RfBandInfoList, 881
pInstances
QmiNasPerformNetworkScanResp, 858
pInstancesSize
nas_RfBandInfoExtFormat, 527
nas_RfBandwidthInfo, 528
nas_RfDedicatedBandInfo, 530
QmiNasGetRFBandInfoResp, 858
RfBandInfoExtFormat, 880
RfBandwidthInfo, 882
RfDedicatedBandInfo, 883
pIntermediateReportState
LOCStartReq, 412
pack_loc_Start_t, 664
pInterval
unpack_pds_GetPDSDefaults_t, 1243
unpack_pds_GetXTRAAutomaticDownload_t, 1245
pIpcpAckTimeout
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 373
Profile3GPP2, 821
pIpcpCreqRetryCount
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pIsPcscfAddressNedded
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pIsUDHPresent
cdmaMsgDecodingParamsExt, 172
wcdmaLongMsgDecodingParams, 1499
pIsVoiceEnabled
custFeaturesInfo, 211
custFeaturesSetting, 213
pJitter
swiQosFlow, 1013
pKeyReferenceID
pack_uim_ChangePin_t, 731
pack_uim_SetPinProtection_t, 733
pack_uim_UnblockPin_t, 740
pack_uim_VerifyPin_t, 741
UIMChangePinReq, 1070
UIMSetPinProtectionReq, 1085
UIMUnblockPinReq, 1089
UIMVerifyPinReq, 1090
PLMN_LENGTH
qaGobiApiNas.h, 1896
PLMNData
nas_operatorPLMNList, 511
operatorPLMNList, 630
PLMNName
nas_operatorNameString, 509
operatorNameString, 628
PLMNNetName
nas_PLMNNetworkName, 518
PLMNNetworkName, 803
PLMNNetworkName, 802
numInstance, 803
PLMNNetName, 803
PLMNNetworkNameData, 803
codingScheme, 804
countryInitials, 804
longName, 804
longNameLen, 804
longNameSpareBits, 804
shortName, 804
shortNameLen, 804
shortNameSpareBits, 805
PLMNRecID
nas_OperatorPLMNData, 510
OperatorPLMNData, 629
pLTEAttachProfile
_slqs3GPPConfigItem, 85
pack_wds_SLQSSet3GPPConfigItem_t, 778
pLTEAttachProfileList
_slqs3GPPConfigItem, 85
pack_wds_SLQSSet3GPPConfigItem_t, 778
pLTEBandPref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pLTECphyCa
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pLTEInfo
swiModemStatusResp, 1004
unpack_nas_SLQSNasSwiModemStatus_t, 1229
pLTEInfoInterfreq
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1222
pLTEInfoIntrafreq
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1222

pLTEInfoNeighboringGSM
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1222
pLTEInfoNeighboringWCDMA
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1222
pLTEOperationMode
LTEOperationMode, 423
nas_LTEOperationMode, 490
NasGetRFInfoResp, 573
QmiNasGetRFBandInfoResp, 858
unpack_nas_SLQSNasGetRFInfo_t, 1224
pLTERSRPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 675
setSignalStrengthInfo, 931
pLTERSRPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 931
pLTERSRPThreshList
LTERSRPThresh, 424
nas_LTERSRPThresh, 492
pLTERSRQDelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 931
pLTERSRQThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 931
pLTERSRQThreshList
LTERSRQThresh, 425
nas_LTERSRQThresh, 492
pLTERSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pLTERSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pLTERSSIThreshList
LTERSSIThresh, 425
nas_LTERSSIThresh, 493
pLTESNRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pLTESNRThresList
LTESNRThresh, 429
nas_LTESNRThresh, 496
pLTESNRThresh
pack_nas_SLQSConfigSigInfo_t, 670
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
sigInfo, 936
pLTESNRThreshList
LTESNRThreshold, 429
nas_LTESNRThreshold, 496
pLTESSInfo
nasGetSigInfoResp, 574
pLTESigInfo
nasSigInfo, 603
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1227
pLTESigRptCfg
pack_nas_SLQSConfigSigInfo_t, 670
sigInfo, 936
pLTESigRptConfig
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pLTESrvStatusInfo
nasGetSysInfoResp, 577
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pLTESysInfo
nasGetSysInfoResp, 577
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pLTEVoiceSupportSysInfo
nasGetSysInfoResp, 577
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1238
pLanguage
cdmaMsgDecodingParams, 168
cdmaMsgDecodingParamsExt, 172
pLastBearerTech
DataBearerTechExt, 218
pLastCallDataBearerTech
getDUNCallInfoResp, 282
pLastCallDataBearerTechnology
dataBearers, 215
pLastCallRXOKBytesCnt
getDUNCallInfoResp, 282
pLastCallTXOKBytesCnt
getDUNCallInfoResp, 282
pLatency
swiQosFlow, 1013
pLatitude
LocInjectPositionReq, 408
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pLcpAckTimeout
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pLcpCreqRetryCount
LibpackProfile3GPP2, 362
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pLeapSeconds
QmiCbkLocPositionReportInd, 850
unpack_loc_PositionRpt_Ind_t, 1178

pLeaseState
WdsDHCPv4ClientLeaseInd, 1532
pLineCtrlInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pLinktimer
pack_sms_SendSMS_t, 716
slqssendasyncsmsparams_s, 948
slqssendsmsparams_s, 949
sms_sendAsyncsmsParams, 968
pLogString
FirmwareUpdatStat, 261
pLogStringLen
FirmwareUpdatStat, 262
pLongitude
LocInjectPositionReq, 408
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pLoopbackMode
WDSSetLoopbackData, 1546
pLoopbackMultiplier
WDSSetLoopbackData, 1546
pLpmFlag
dmsSwiGetPCInfo, 244
pLteBandCapability
BandCapabilityResp, 142
pLteEARFCN
nasSwiGetChannelLockResp, 604
nasSwiSetChannelLockReq, 605
pack_nas_SLQSNASSwiSetChannelLock_t, 683
unpack_nas_SLQSNASSwiGetChannelLock_t, 1228
pLteM1BandPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
sysSelectPrefParamsExt, 1022
pLteNasRelInfo
SwiOTAMsg_s, 1004
pLteNb1BandPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
sysSelectPrefParamsExt, 1022
pLtePCI
nasSwiGetChannelLockResp, 604
nasSwiSetChannelLockReq, 605
pack_nas_SLQSNASSwiSetChannelLock_t, 683
unpack_nas_SLQSNASSwiGetChannelLock_t, 1228
pLteQci
swiQosFlow, 1013
pMCC
QmiNas3GppNetworkInfo, 857
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pMEIDString
serialNumbersInfo, 904
pMICGainSelect
GetAudioPathConfigResp, 273
pMIPStatusInd
pack_wds_SLQSWdsSetEventReport_t, 785
wdsSetEventReportReq, 1545
pMNAAA
pack_wds_SetMobileIPProfile_t, 771
pMNC
QmiNas3GppNetworkInfo, 857
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pMNCIncPCSDigStat
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
pMNHA
pack_wds_SetMobileIPProfile_t, 771
pMNRInfo
nasInitNetworkReg, 582
pack_nas_SLQSInitiateNetworkRegistration_t, 671
pMTMessageInfo
SMSEventInfo_s, 978
pMTUSize3gpp
SwiDmsGetMTUResp, 999
unpack_swidms_SLQSSwiDmsGetMTU_t, 1290
pMagneticDeviation
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pManString
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
pManagedRoamingInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pMaxAllowedPktSz
swiQosFlow, 1013
pMaxChannelRXRate
WdsConnectionRateElmnts, 1532
pMaxChannelTXRate
WdsConnectionRateElmnts, 1532
pMaxStorageSizeReq
pack_sms_SLQSSmsGetMaxStorageSize_t, 724
pMaxStorageSizeResp
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1285
pMdmCallDurationActive
getDUNCallInfoResp, 282
pMeidString
_SLQSSwiGetSerialNoExtParams, 99
pMessage
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
cdmaMsgEncodingParams, 174
pack_sms_SaveSMS_t, 716
pack_sms_SendSMS_t, 717
slqssendasyncsmsparams_s, 948
slqssendsmsparams_s, 949
sms_sendAsyncsmsParams, 968

wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
pMessageID
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
pMessageIndex
pack_sms_SLQSDeleteSMS_t, 718
unpack_sms_SaveSMS_t, 1278
pMessageMode
pack_sms_SLQSDeleteSMS_t, 718
pack_sms_SLQSGetSMS_t, 719
pack_sms_SLQSGetSMSList_t, 720
pack_sms_SLQSModifySMSStatus_t, 721
sms_maxStorageSizeReq, 962
smsMaxStorageSizeReq, 978
pMessageModeInfo
SMSEventInfo_s, 978
pMessageProtocol
unpack_sms_SLQSSmsGetMessageProtocol_t, 1285
pMessageSize
cdmaMsgEncodingParams, 174
pMessageTag
pack_sms_SLQSDeleteSMS_t, 718
pMicMute
SetM2MAudioProfileReq, 923
pMinBasedIMSI
nasGet3GPP2SubscriptionInfoResp, 569
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1220
pMinIntervalTime
LOCStartReq, 412
pack_loc_Start_t, 664
pMinPolicedPktSz
swiQosFlow, 1013
pMinSessionExpiryTimer
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pMitigationDevList
TmdGetMitigationDevListResp, 1034
pMitigationDevListLen
TmdGetMitigationDevListResp, 1034
pMncPcsDigitStatus
nasInitNetworkReg, 582
pack_nas_SLQSInitiateNetworkRegistration_t, 671
pMncPcsStatus
nasPLMNNameReq, 594
pack_nas_SLQSGetPLMNName_t, 671
pMode
pack_wds_SetMobileIPParameters_t, 769
pModePref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pModelString
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
pMtu
qmiWdsRunTimeSettings, 864
pMuxID
pack_qos_BindDataPort_t, 713
pNAI
pack_wds_SetMobileIPProfile_t, 771
pNAMNameInfo
nasGet3GPP2SubscriptionInfoResp, 569
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1220
pNID
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNITZInformation
nasOperatorNameResp, 589
unpack_nas_SLQSGetOperatorNameData_t, 1202
pNSEnable
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pNSSAudioCtrl
unpack_voice_VoiceInfoRecCallback_ind_t, 1350
voiceInfoRec, 1476
pNSSRelease
unpack_voice_VoiceInfoRecCallback_ind_t, 1351
voiceInfoRec, 1476
pNamID
pack_voice_SLQSVoiceGetConfig_t, 751
voiceGetConfigReq, 1471
pName
pack_wds_SetDefaultProfile_t, 767
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNameString
_SLQSSwiGetOSInfoParams, 98
_SLQSSwiSetOSInfoParams, 101
pNegoDnsSrvrPref
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pNeighborSetPilotPN
nas_NetworkStat1x, 505
NetworkStat1x, 614
pNetSelPref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pNetworkInfo
_slqsNetworkScanInfo, 90
pNetworkInfoInstances
_slqsNetworkScanInfo, 90
pNetworkInfoLen
CurrDataSysStat, 203
pNetworkMode
pack_pds_SLQSGetAGPSConfig_t, 706
pack_pds_SLQSSetAGPSConfig_t, 709
pNetworkRejectInd

nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pNetworkStat1x
NetworkDebugResp, 612
unpack_nas_SLQSSwiNetworkDebug_t, 1235
pNetworkStatEVDO
NetworkDebugResp, 612
unpack_nas_SLQSSwiNetworkDebug_t, 1235
pNetworkTimeInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pNewImsRegStatus
IMSARegistrationStatus, 331
pNewPwdData
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pNotifStore
_SLQSAVMSSetSettingsReqPara, 87
pNotificationStore
_SLQSAVMSSettings, 87
pNr5gBandPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
sysSelectPrefParamsExt, 1022
pNumSupUSBComps
USBCompParams, 1392
pNumberOfPhySlot
UIMGetSlotsStatusResp, 1075
unpack_uim_SLQSUIMGetSlotsStatus_t, 1305
pNw2DescDisp
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNw2DescEnc
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNw2DescLen
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNw2MCC
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNw2MNC
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNw2Name
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pNxtHdrProto
swiQosFilter, 1009
pOMADMEnabled
_SLQSAVMSSettings, 88
_SLQSOMADMSettings, 95
PORTNAM_LEN
qaGobiApiDcs.h, 1805
pOTASPStatus
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1450
voiceGetAllCallInfo, 1458
pObjectVer
NetworkDebugResp, 612
unpack_nas_SLQSSwiNetworkDebug_t, 1235
pOffLength
unpack_voice_DTMFEventCallback_ind_t, 1312
voiceDTMFEventInfo, 1455
pOnLength
unpack_voice_DTMFEventCallback_ind_t, 1312
voiceDTMFEventInfo, 1455
pOpaqueIdentifier
LocInjectSensorDataReq, 409
QmiCbkLocInjectSensorDataInd, 843
unpack_loc_InjectSensorDataCallback_Ind_t, 1170
pOperation
unpack_pds_GetPDSDefaults_t, 1243
pOperatorNameString
nasOperatorNameResp, 589
unpack_nas_SLQSGetOperatorNameData_t, 1202
pOperatorPLMNList
nasOperatorNameResp, 589
unpack_nas_SLQSGetOperatorNameData_t, 1202
pOptList
wds_DHCPv4OptionList, 1516
WdsDHCPv4ClientLeaseInd, 1532
WdsDHCPv4OptionList, 1536
wdsDhcpv4OptionList, 1537
pOptVal
DHCPOption, 228
wds_DHCPOpt, 1514
pOptions
DHCPOptionList, 229
pPCMParams
SetM2MAudioAVCFGReq, 921
pPCSCFAddrPCO
qmiWdsRunTimeSettings, 864
pPCSCFFQDNAddrList
qmiWdsRunTimeSettings, 864
pPCSCFPort
GetRegMgrConfigParams, 296
pPCSDigitInfo
_slqsNetworkScanInfo, 90
pPCSDigitInstances
_slqsNetworkScanInfo, 90
pPCSInstance
unpack_nas_PerformNetworkScan_t, 1196
pPCSInstanceSize
unpack_nas_PerformNetworkScan_t, 1196
pPDNInactivTimeout
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPDNInactivTimeout3GPP2
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pPDPType
qmiWdsRunTimeSettings, 864
pPDPtype
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pPDUMessage

wcdmaMsgEncodingParams, 1502
pPLMNNetworkName
nasOperatorNameResp, 589
unpack_nas_SLQSGetOperatorNameData_t, 1202
pPRLPref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 112
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pPRLPreference
dmsCurrentPRLInfo, 238
pPRLVersion
dmsCurrentPRLInfo, 238
pPWD
_SLQSAVMSSettingsAPNInfo, 88
pPacketsCountRX
QosEventInfo, 868
slqsWdsEventInfo, 957
pPacketsCountTX
QosEventInfo, 868
slqsWdsEventInfo, 957
pPagingTimeWindow
nasGeteDRXParamsResp, 570
nasSeteDRXParamsReq, 602
pack_nas_SLQSNASSeteDRXParams_t, 681
unpack_nas_SLQSNASGeteDRXParams_t, 1222
pPartNum
cdmaMsgDecodingParamsExt, 172
wcdmaLongMsgDecodingParams, 1500
pPass
pack_wds_SLQSStartDataSession_t, 782
pPassword
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
pack_wds_SetDefaultProfile_t, 767
Profile3GPP, 815
ssdatasession_params, 992
pPasswordSize
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPcscfAddrUsingDhcp
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPcscfAddrUsingPCO
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPdnType
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pPdpAccessConFlag
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPdpContext
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPdpDataCompType
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPdpHdrCompType
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 815
pPdpStatusConfig
IMSAIndRegisterInfo, 328
pPeriodicUpdateTimer
dmsPsmCfgParamChange, 242
pack_dms_SLQSSetPowerSaveModeConfig_t, 639
PowerSaveModeConfigReq, 806
PowerSaveModeConfigResp, 807
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1140
pPeripheralEndPointID
pack_qos_BindDataPort_t, 713
pPeroidInfo
_SLQSAVMSSetSettingsReqPara, 87
pPeroidsInfo
_SLQSAVMSSettings, 88
pPersistentLpm
dmsSwiGetPCInfo, 244
pPersonalityListLength
HDRPersonalityInd, 315
HDRPersonalityResp, 315
HDRProtSubtypResp, 316
nas_HDRPersonality_Ind_Data, 476
unpack_nas_SLQSSwiGetHDRPersonality_t, 1231
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1232
pPersonalizationStatus
UIMGetConfigurationResp, 1074
unpack_uim_SLQSUIMGetConfiguration_t, 1304
pPhoneCtxtURI
GetIMSSMSConfigParams, 286
imsSMSConfigInfo, 339
SetIMSSMSConfigReq, 916
pPhoneCtxtURILen
GetIMSSMSConfigParams, 286
SetIMSSMSConfigReq, 916
pPhyCaAggPcellInfo
nasGetLTECphyCaResp, 572
pPhyCaAggScellDlBw
nasGetLTECphyCaResp, 572
pPhyCaAggScellIndType
nasGetLTECphyCaResp, 572
pPhyCaAggScellIndex
nasGetLTECphyCaResp, 572
pPhyCaAggScellInfo
nasGetLTECphyCaResp, 572

pPilotSetData
GetHRPDStatsResp, 285
unpack_nas_SLQSSwiGetHRPDStats_t, 1233
pPilotSetInfo
nas_PilotSetData, 516
PilotSetData, 800
pPkgDescLength
_SLQSOMADMSessionInfo, 93
pPkgDescription
_SLQSOMADMSessionInfo, 93
pPkgName
_SLQSOMADMSessionInfo, 93
pPkgNameLength
_SLQSOMADMSessionInfo, 93
pPktErrRate
swiQosFlow, 1013
pPlasmaIDString
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
pPlmnId
nasNetworkReject, 587
pPollingTimer
_SLQSAVMSSetSettingsReqPara, 87
_SLQSAVMSSettings, 88
pPositionSource
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
pPositionSrc
LocInjectPositionReq, 408
pPowerOffMode
dmsSwiGetPCInfo, 244
pPppSessCloseTimer1x
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pPppSessCloseTimerDO
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pPrDNSIPv4Address
swiPDPRuntimeSettingsResp, 1007
pPrDNSIPv6Address
swiPDPRuntimeSettingsResp, 1007
pPrPCSCFIPv4Address
swiPDPRuntimeSettingsResp, 1007
pPrPCSCFIPv6Address
swiPDPRuntimeSettingsResp, 1007
pPrecedence
swiQosFilter, 1010
pPrecisionDilution
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pPrefNetwork
CurrDataSysStat, 203
pPrefVoiceDomain
pack_voice_SLQSVoiceSetConfig_t, 757
voiceSetConfigReq, 1486
pPrefVoicePrivacy
pack_voice_SLQSVoiceGetConfig_t, 751
voiceGetConfigReq, 1471
pPrefVoiceSO
pack_voice_SLQSVoiceGetConfig_t, 751
pack_voice_SLQSVoiceSetConfig_t, 757
voiceGetConfigReq, 1471
voiceSetConfigReq, 1486
pPrefVoiceSOStatus
unpack_voice_SLQSVoiceSetConfig_t, 1341
voiceSetConfigResp, 1487
pPreference
unpack_pds_GetXTRANetwork_t, 1246
pPreferred
QmiNas3GppNetworkInfo, 857
pPriCSCFPort
imsRegMgrConfigInfo, 338
SetRegMgrConfigReq, 927
pPriCSCFPortName
GetRegMgrConfigParams, 296
pPriCSCFPortNameLen
GetRegMgrConfigParams, 296
pPriDNSIPv4AddPref
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pPriDNSIPv6addpref
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pPriV6DnsAddress
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pPrimaryDNSV4
qmiWdsRunTimeSettings, 864
pPrimaryDNSV6
qmiWdsRunTimeSettings, 864
pPrimaryHA
pack_wds_SetMobileIPProfile_t, 771
pPrimaryID
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pPrimaryV4DnsAddress
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pPriority
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
cdmaMsgEncodingParams, 174
pPrivacy
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
pProfSz
NWProfile, 620

pProfValues
NWProfile, 620
pProfileID
CreateProfileIn, 198
ModifyProfileIn, 439
qmiWdsRunTimeSettings, 864
unpack_wds_SLQSCreateProfile_t, 1366
pProfileId
pack_wds_SLQSCreateProfile_t, 772
pack_wds_SLQSModifyProfile_t, 777
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 780
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 781
WdsDHCPv4ClientLeaseInd, 1533
WdsDHCPv4Config, 1533
pProfileId3GPP
ssdatasession_params, 992
pProfileId3GPP2
ssdatasession_params, 992
swiQosFlow, 1013
pProfileIndex
CreateProfileOut, 199
pProfileList
_slqs3GPPConfigItem, 85
pack_wds_SLQSSet3GPPConfigItem_t, 778
pProfileName
qmiWdsRunTimeSettings, 864
pProfileSettings
unpack_wds_SLQSGetProfileSettings_t, 1375
pProfileType
CreateProfileIn, 198
CreateProfileOut, 199
ModifyProfileIn, 439
pack_wds_SLQSCreateProfile_t, 772
pack_wds_SLQSModifyProfile_t, 777
pProfilename
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pProfilenameSize
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pProtoSubTypElmnt
HDRProtSubtypResp, 316
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1232
pProtocol
pack_nas_SetCDMANetworkParameters_t, 667
pProtocolSubtypeElement
HDRPersonalityInd, 315
HDRPersonalityResp, 315
nas_HDRPersonality_Ind_Data, 476
unpack_nas_SLQSSwiGetHDRPersonality_t, 1231
pPsmEnableState
pack_dms_SLQSSetPowerSaveModeConfig_t, 639
PowerSaveModeConfigReq, 806
PowerSaveModeConfigResp, 807
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1140
pQFlowState
QosFlowInfo, 869
pQmiInterfaceInfo
_packetSrvStatus, 72
slqsSessionStateInfo, 950
slqsWdsEventInfo, 957
pQosClassID
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pRAT
_sysSelectPrefParams, 113
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 699
pRATINstance
unpack_nas_PerformNetworkScan_t, 1196
pRATInfo
_slqsNetworkScanInfo, 90
pRATInstanceSize
unpack_nas_PerformNetworkScan_t, 1196
pRATInstances
_slqsNetworkScanInfo, 90
pRATStatus
imsaRatStatusInfo, 330
pRATType
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
PREFERRED_INDEX
qaNasPerformNetworkScan.h, 2143
pRFBandInfoElements
QmiNasGetRFBandInfoResp, 858
pRFBandInfoParam
nas_RfBandInfoList, 528
RfBandInfoList, 881
PRI_UPDATE_FAIL
qaGobiApiFms.h, 1857
PRIString
unpack_dms_GetFirmwareRevision_t, 1115
unpack_dms_GetFirmwareRevisions_t, 1116
PRLInd
qaQmiServingSystemParam, 833
unpack_nas_SLQSGetServingSystem_t, 1209
PRLPTlv
NASQmiCbkNasSystemSelPrefInd, 599
PRLPref
NASPRLPreferenceTlv, 597
pRMAutoConnect
custFeaturesInfo, 211
pRSRPThresList
nas_RSRPThresh, 533
RSRPThresh, 887
pRSRPThresh
pack_nas_SLQSConfigSigInfo_t, 670
sigInfo, 936
pRSRQThresList

nas_RSRQThresh, 534
RSRQThresh, 888
pRSRQThresh
pack_nas_SLQSConfigSigInfo_t, 670
sigInfo, 936
pRSSIThresList
nas_RSSIThresh, 535
RSSIThresh, 889
pRSSIThresh
pack_nas_SLQSConfigSigInfo_t, 670
sigInfo, 936
pRTPRTCPInactTimer
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pRXAGCList
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pRXAIG
RXAGCList, 890
pRXAVCAGCSwitch
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pRXAVCList
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pRXChain0Info
nasGetTxRxInfoResp, 578
unpack_nas_SLQSNasGetTxRxInfo_t, 1225
pRXChain1Info
nasGetTxRxInfoResp, 578
unpack_nas_SLQSNasGetTxRxInfo_t, 1225
pRXComprSlope
RXAGCList, 890
pRXComprThres
RXAGCList, 890
pRXDroppedCount
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1540
pRXExpSlope
RXAGCList, 890
pRXExpThres
RXAGCList, 890
pRXOKBytesCount
getDUNCallInfoResp, 282
pRXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1540
pRXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1540
pRXPCMIIRFltr
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pRXPacketErrors
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1540
pRXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1540
pRXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1540
pRXStaticGain
RXAGCList, 890
pRXTotalBytes
unpack_wds_GetByteTotals_t, 1353
WdsByteTotalsElmnts, 1530
pRadioInterface
nasNetworkTime, 588
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1226
pRandomizationWindow
PowerSaveModeConfigResp, 807
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1140
pRankIndicatorInd
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
pRatHandoverStatusConfig
IMSAIndRegisterInfo, 328
pRawHorConfidence
LocInjectPositionReq, 408
pRawHorUncCircular
LocInjectPositionReq, 408
pReRegPeriod
pack_wds_SetMobileIPParameters_t, 769
pReRegTraffic
pack_wds_SetMobileIPParameters_t, 769
pReadAcknowledgementReq
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
pReadResult
UIMReadTransparentResp, 1079
unpack_uim_ReadTransparent_t, 1300
pReason
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pReasonMask
unpack_dms_GetOfflineReason_t, 1120
pRecurrenceType
LOCStartReq, 412
pack_loc_Start_t, 664
pRedirNumInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1351
voiceInfoRec, 1476
pRefData
FirmwareUpdatStat, 262
pRefString
FirmwareUpdatStat, 262
pRefStringLen
FirmwareUpdatStat, 262
pReferenceNum
cdmaMsgDecodingParamsExt, 172
wcdmaLongMsgDecodingParams, 1500

pRefreshEvent
UIMRefreshGetLastEventResp, 1082
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 1306
pRegCallStatInfoEvt
_getIndicationRegResp, 66
_setIndicationRegReq, 83
sms_getIndicationReg, 960
sms_setIndicationReg, 969
pRegDTMFEvents
pack_voice_SLQSVoiceIndicationRegister_t, 752
voiceIndicationRegisterInfo, 1474
pRegInd
_getTransLayerInfoResp, 68
sms_getTransLayerInfo, 961
pRegMgrConfigEvents
imsCfgIndRegisterInfo, 337
pRegStatus
_getTransNWRegInfoResp, 69
sms_getTransNWRegInfo, 962
pRegStatusConfig
IMSAIndRegisterInfo, 329
pRegStatusErrorCode
imsaRegStatusInfo, 331
pRegTransLayerInfoEvt
_getIndicationRegResp, 66
_setIndicationRegReq, 83
sms_getIndicationReg, 960
sms_setIndicationReg, 969
pRegTransNWRegInfoEvt
_getIndicationRegResp, 66
_setIndicationRegReq, 83
sms_getIndicationReg, 960
sms_setIndicationReg, 969
pRegVoicePrivacyEvents
pack_voice_SLQSVoiceIndicationRegister_t, 752
voiceIndicationRegisterInfo, 1474
pRejectReason
pack_swioma_SLQSOMADMSendSelection_t, 728
pRelValidity
cdmaMsgEncodingParams, 174
pRelativeValidity
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
pRemainingRetries
UIMDepersonalizationResp, 1071
UIMPinResp, 1076
unpack_uim_ChangePin_t, 1299
unpack_uim_SetPinProtection_t, 1301
unpack_uim_SLQSUIMDepersonalization_t, 1303
unpack_uim_UnblockPin_t, 1307
unpack_uim_VerifyPin_t, 1308
pRemotePartyName
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1450
pRemotePartyNum
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1450
pReportChannelRate
getDUNCallInfoReq, 279
pack_wds_SLQSGetDUNCallInfo_t, 774
pReportConnStatus
getDUNCallInfoReq, 279
pack_wds_SLQSGetDUNCallInfo_t, 774
pReportDataBearerTech
getDUNCallInfoReq, 279
pack_wds_SLQSGetDUNCallInfo_t, 774
pReportDormStatus
getDUNCallInfoReq, 279
pack_wds_SLQSGetDUNCallInfo_t, 774
pReqFieldsList
IMSASupportedFieldsResp, 334
pReqMitigationLvl
TmdGetMitigationLvlResp, 1035
pReqSettings
pack_wds_SLQSGetRuntimeSettings_t, 776
pRequestOptionList
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 781
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1383
WdsDHCPv4Config, 1533
pRequestedTag
pack_sms_SLQSGetSMSList_t, 720
pRespData
USSDRespFNetwork, 1394
pRespFieldsList
IMSASupportedFieldsResp, 334
pRetryCount
_SLQSOMADMSessionInfo, 93
pRetryInterval
pack_wds_SetMobileIPParameters_t, 769
pRetryLimit
pack_wds_SetMobileIPParameters_t, 769
pRetryMessage
slqssendasyncsmsparams_s, 948
sms_sendAsyncsmsParams, 968
pRetryMessageId
slqssendasyncsmsparams_s, 948
sms_sendAsyncsmsParams, 968
pRevInUse
CDMASysInfo, 179
nas_CDMASysInfo, 451
pRevInUseValid
CDMASysInfo, 179
nas_CDMASysInfo, 452
pRevTunneling
pack_wds_SetMobileIPProfile_t, 771
pRfBandInfoExtFormat
NasGetRFInfoResp, 573
QmiNasGetRFBandInfoResp, 858
unpack_nas_SLQSNasGetRFInfo_t, 1224
pRfBandInfoExtFormatParam
nas_RfBandInfoExtFormat, 527
RfBandInfoExtFormat, 880
pRfBandwidthInfo

NasGetRFInfoResp, 573
QmiNasGetRFBandInfoResp, 858
unpack_nas_SLQSNasGetRFInfo_t, 1224
pRfBandwidthInfoParam
nas_RfBandwidthInfo, 528
RfBandwidthInfo, 882
pRfDedicatedBandInfo
NasGetRFInfoResp, 573
QmiNasGetRFBandInfoResp, 858
unpack_nas_SLQSNasGetRFInfo_t, 1224
pRfDedicatedBandInfoParam
nas_RfDedicatedBandInfo, 530
RfDedicatedBandInfo, 883
pRingBackTimer
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pRingingTimer
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pRoamPref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 113
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 700
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pRoamTimer
pack_voice_SLQSVoiceGetConfig_t, 751
voiceGetConfigReq, 1471
pRoamTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1338
voiceGetConfigResp, 1473
pRoamTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 757
voiceSetConfigReq, 1486
pRoamTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1342
voiceSetConfigResp, 1487
pRoaming
pack_nas_SetCDMANetworkParameters_t, 667
QmiNas3GppNetworkInfo, 857
pRscp
nasSigInfo, 603
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1227
pRxFilter
swiQosModifyReq, 1014
swiQosReq, 1015
pRxFlow
swiQosGranted, 1013
swiQosModifyReq, 1014
swiQosReq, 1015
pRxQFilter
QosFlowInfo, 869
pRxQFlowGranted
QosFlowInfo, 869
PSDetachReq, 824
pDetachAction, 824
PSDomain
unpack_nas_GetServingNetwork_t, 1193
pSID
unpack_nas_GetHomeNetwork3GPP2_t, 1188
pSIODataPort
pack_qos_BindDataPort_t, 713
pSIPConfigEvents
imsCfgIndRegisterInfo, 337
pSIPLocalPort
GetSIPConfigResp, 297
imsSIPConfigInfo, 339
SetSIPConfigReq, 933
PSMActiveTimerTlv, 824
activeTimer, 824
TlvPresent, 824
PSMCfgChangeInfo
pack_dms_SetIndicationRegister_t, 636
PSMDurationDueToOOSTlv, 825
durationDueToOOS, 825
TlvPresent, 825
PSMDurationThresholdTlv, 825
durationThreshold, 825
TlvPresent, 825
PSMEarlyWakeupTimeTlv, 826
earlyWakeupTime, 826
TlvPresent, 826
PSMEnableStateTlv, 826
enableState, 826
TlvPresent, 826
PSMPeriodicUpdateTimerTlv, 827
periodicUpdateTimer, 827
TlvPresent, 827
PSMRandomizationWindowTlv, 827
randomizationWindow, 827
TlvPresent, 827
pSMSAttemptedFlag
getDyingGaspStatistics, 283
packgetDyingGaspStatistics, 787
pSMSCAddress
pack_sms_SetSMSCAddress_t, 717
unpack_sms_GetSMSCAddress_t, 1277
pSMSCAddressInfo
SMSEventInfo_s, 978
pSMSCType
pack_sms_SetSMSCAddress_t, 717
unpack_sms_GetSMSCAddress_t, 1277
pSMSConfigEvents
imsCfgIndRegisterInfo, 337
pSMSFormat
GetIMSSMSConfigParams, 286
imsSMSConfigInfo, 339
SetIMSSMSConfigReq, 916
pSMSOnIMSInfo
SMSEventInfo_s, 978
pSMSOverIPNwInd
GetIMSSMSConfigParams, 286
imsSMSConfigInfo, 339

SetIMSSMSConfigReq, 916
pSMSSupport
custFeaturesInfo, 211
pSMSSvcRAT
imsaSvcStatusInfo, 335
pSMSSvcStatus
imsaSvcStatusInfo, 335
PSMStatus
pack_dms_SetIndicationRegister_t, 636
pSPC
pack_nas_SetCDMANetworkParameters_t, 667
pack_wds_SetMobileIPParameters_t, 769
pSV
BdsSVInfo, 143
loc_BdsSVInfo, 383
loc_SVInfo, 397
SVInfo, 995
pSVInfo
LocDelAssDataReq, 401
pack_loc_Delete_Assist_Data_t, 650
pSWVerString
_SLQSSwiGetHostDevInfoParams, 98
_SLQSSwiSetHostDevInfoParams, 100
pSarRFState
unpack_sar_SLQSGetRfSarState_t, 1277
pSatelliteInfo
gnssSvInfoNotification, 300
unpack_loc_GnssSvInfo_Ind_t, 1168
pScAddr
wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
pScAddrLength
wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
pScanResult
_slqsNetworkScanInfo, 90
unpack_nas_PerformNetworkScan_t, 1196
pSccRxInfo
LteSccRxInfoResp, 426
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1234
pScrAmrEnable
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pScrAmrWbEnable
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pSeDNSIPv4Address
swiPDPRuntimeSettingsResp, 1007
pSeDNSIPv6Address
swiPDPRuntimeSettingsResp, 1007
pSePCSCFIPv4Address
swiPDPRuntimeSettingsResp, 1007
pSePCSCFIPv6Address
swiPDPRuntimeSettingsResp, 1007
pSecDNSIPv4AddPref
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pSecDNSIPv6addpref
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 368
Profile3GPP, 816
pSecV6DnsAddress
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pSecondaryDNSV4
qmiWdsRunTimeSettings, 865
pSecondaryDNSV6
qmiWdsRunTimeSettings, 865
pSecondaryFlag
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 369
Profile3GPP, 816
pSecondaryHA
pack_wds_SetMobileIPProfile_t, 771
pSecondaryV4DnsAddress
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pSectorID
nas_NetworkStatEVDO, 507
NetworkStatEVDO, 616
pSendSmsParams
pack_sms_SLQSSendAsyncSMS_t, 721
pSenderAddr
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
pSenderAddrLength
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
pSensorDataUsage
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pServerAddrList
qmiWdsRunTimeSettings, 865
pServerAddrTypeMask
LocGetServerReq, 404
pServerAddress
pack_pds_SLQSSetAGPSConfig_t, 709
unpack_pds_SLQSGetAGPSConfig_t, 1250
pServerPort
pack_pds_SLQSSetAGPSConfig_t, 709
unpack_pds_SLQSGetAGPSConfig_t, 1250
pServerURL
pack_pds_SLQSSetAGPSConfig_t, 709
unpack_pds_SLQSGetAGPSConfig_t, 1250
pServerURLLength

pack_pds_SLQSSetAGPSConfig_t, 709
unpack_pds_SLQSGetAGPSConfig_t, 1250
pServiceClass
pack_voice_SLQSVoiceSetSUPSService_t, 759
voiceSetSUPSServiceReq, 1490
pServiceOption
slqssendasyncsmsparams_s, 948
sms_sendAsyncsmsParams, 968
pServiceStatusConfig
IMSAIndRegisterInfo, 329
pServingSystemInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pSessionExpiryTimer
GetIMSVoIPConfigResp, 289
imsVoIPConfigInfo, 342
SetIMSVoIPConfigReq, 920
pSessionIDv4
GetSessionIDResp, 296
pSessionIDv6
GetSessionIDResp, 296
pSessionState
_SLQSOMADMSessionInfo, 93
pSessionType
_SLQSOMADMSessionInfo, 93
pSetIndicationRegReq
pack_sms_SLQSSetIndicationRegister_t, 722
pSetRoutesReq
pack_sms_SLQSSmsSetRoutes_t, 724
pSettingResp
GetIMSSMSConfigParams, 286
GetIMSUserConfigParams, 287
GetIMSVoIPConfigResp, 289
GetRegMgrConfigParams, 296
GetSIPConfigResp, 297
SetIMSSMSConfigResp, 917
SetIMSUserConfigResp, 918
SetIMSVoIPConfigResp, 921
SetRegMgrConfigResp, 927
SetSIPConfigResp, 934
pSeverity
_SLQSOMADMSessionInfo, 93
pSigCompEnabled
GetSIPConfigResp, 297
imsSIPConfigInfo, 339
SetSIPConfigReq, 933
pSigIndReq
pack_nas_SLQSSetSignalStrengthsCallback_t, 685
pSignalInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 1351
voiceInfoRec, 1477
pSignalStrengthInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pSmsOnIms
slqssendasyncsmsparams_s, 948
slqssendsmsparams_s, 949
sms_sendAsyncsmsParams, 968
pSmsServiceRat
IMSAServiceStatus, 333
pSmsServiceStatus
IMSAServiceStatus, 333
pSmsStorage
unpack_sms_SLQSSwiGetSMSStorage_t, 1286
pSource
_SLQSOMADMSessionInfo, 93
pSourceIP
LibPackTFTIDParams, 377
TFTIDParams, 1031
pSourceLength
_SLQSOMADMSessionInfo, 93
pSpeedHorizontal
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pSpeedUnc
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pSpeedVertical
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pSpeedVerticalUnc
QmiCbkLocBestAvailPosInd, 838
unpack_loc_BestAvailPos_Ind_t, 1162
pSrcRAT
imsaRatStatusInfo, 330
pSrvDomainPref
_sysSelectPrefInfo, 106
_sysSelectPrefParams, 113
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 700
unpack_nas_SLQSGetSysSelectionPref_t, 1219
pSrvOpt
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1451
pSrvRegRestriction
_sysSelectPrefParams, 113
pack_nas_SLQSSetSysSelectionPref_t, 690
pack_nas_SLQSSetSysSelectionPrefExt_t, 700
pSrvcProviderName
nasOperatorNameResp, 589
unpack_nas_SLQSGetOperatorNameData_t, 1202
pStage0Val
RXPCMIIRFltr, 893
TXPCMIIRFltr, 1043
pStage1Val
RXPCMIIRFltr, 893
TXPCMIIRFltr, 1043
pStage2Val
RXPCMIIRFltr, 893

TXPCMIIRFltr, 1043
pStage3Val
RXPCMIIRFltr, 893
TXPCMIIRFltr, 1043
pStage4Val
RXPCMIIRFltr, 893
TXPCMIIRFltr, 1043
pStageCnt
RXPCMIIRFltr, 893
TXPCMIIRFltr, 1043
pStatMask
pack_wds_GetPacketStatistics_t, 763
WdsPktStatisticsReq, 1541
pStatus
_SLQSOMADMSessionInfo, 93
pSubnetMaskV4
qmiWdsRunTimeSettings, 865
pSubscribeTimer
GetSIPConfigResp, 297
imsSIPConfigInfo, 339
SetSIPConfigReq, 933
pSubscriptionInfoInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pSupUSBComps
USBCompParams, 1392
pSupportEmergencyCalls
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 369
Profile3GPP, 816
pSupportedMsgList
IMSASupportedMsgInfo, 335
pSuppressSysInfoInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pSuppsNotifEvents
pack_voice_SLQSVoiceIndicationRegister_t, 752
voiceIndicationRegisterInfo, 1474
pSvUsedforFix
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pSvcClass
pack_voice_SLQSVoiceGetCallBarring_t, 748
pack_voice_SLQSVoiceGetCallForwardingStatus-_t, 749
pack_voice_SLQSVoiceGetCallWaiting_t, 750
unpack_voice_SLQSVoiceGetCallBarring_t, 1323
unpack_voice_SLQSVoiceGetCallWaiting_t, 1329
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceGetCallBarringReq, 1459
voiceGetCallBarringResp, 1460
voiceGetCallFWReq, 1461
voiceGetCallWaitInfo, 1464
voiceSUPSInfo, 1494
pSvcType
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
pSwiGetResetInd
dmsIndicationRegisterReq, 241
pSysInfoInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pSysInfoNoChange
nasSysInfo, 608
unpack_nas_SLQSSysInfoCallback_ind_t, 1239
pSysSelectPrefParams
sysSelectPrefParamsExt, 1022
pSystemSelectionInd
nasIndicationRegisterReq, 581
pack_nas_SLQSNasIndicationRegisterExt_t, 680
pTCPDstPort
swiQosFilter, 1010
pTCPSrcPort
swiQosFilter, 1010
pTDSCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMAECIOThreshList
nas_TDSCDMAECIOThresh, 549
TDSCDMAECIOThresh, 1023
pTDSCDMARSCPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMARSCPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMARSCPThreshList
nas_TDSCDMARSCPThresh, 549
TDSCDMARSCPThresh, 1024
pTDSCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMARSSIThreshList
nas_TDSCDMARSSIThresh, 550
TDSCDMARSSIThresh, 1024
pTDSCDMASINRCONFThresh
sigInfo, 936
pTDSCDMASINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMASINRThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pTDSCDMASINRThreshList
nas_TDSCDMASINRThresh, 551
TDSCDMASINRThresh, 1027
pTDSCDMASigInfoExt

nasGetSigInfoResp, 574
nasSigInfo, 603
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1227
pTDSCDMASigInfoRscp
nasGetSigInfoResp, 574
pTFTID1Params
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 369
Profile3GPP, 816
pTFTID2Params
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 369
Profile3GPP, 816
pTTYConfigStatus
unpack_voice_SLQSVoiceSetConfig_t, 1342
voiceSetConfigResp, 1487
pTTYMode
pack_voice_SLQSVoiceGetConfig_t, 751
pack_voice_SLQSVoiceSetConfig_t, 757
voiceGetConfigReq, 1472
voiceSetConfigReq, 1486
pTXAGCList
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pTXAIG
TXAGCList, 1041
pTXAVCSwitch
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pTXComprSlope
TXAGCList, 1041
pTXComprThres
TXAGCList, 1041
pTXDroppedCount
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1541
pTXExpSlope
TXAGCList, 1041
pTXExpThres
TXAGCList, 1041
pTXGain
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pTXInfo
nasGetTxRxInfoResp, 578
unpack_nas_SLQSNasGetTxRxInfo_t, 1225
pTXOKBytesCount
getDUNCallInfoResp, 282
pTXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1541
pTXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1541
pTXPCMIIRFltr
GetAudioPathConfigResp, 273
SetAudioPathConfigReq, 912
pTXPacketErrors
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1541
pTXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1541
pTXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1362
WdsPktStatisticsElmnts, 1541
pTXStaticGain
TXAGCList, 1041
pTXTotalBytes
unpack_wds_GetByteTotals_t, 1353
WdsByteTotalsElmnts, 1530
pTdsBandCapability
BandCapabilityResp, 142
pTdscdmaBandPref
_sysSelectPrefParams, 113
pack_nas_SLQSSetSysSelectionPref_t, 691
pack_nas_SLQSSetSysSelectionPrefExt_t, 700
pTech
pack_wds_SLQSStartDataSession_t, 782
pTechnology
qmiWdsRunTimeSettings, 865
ssdatasession_params, 992
pTechnologyMask
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pTempReport
pack_dms_SwiSetEventReport_t, 642
pTextMsg
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
cdmaMsgEncodingParams, 174
wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
wcdmaMsgEncodingParams, 1502
pTextMsgLength
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
pTgtRAT
imsaRatStatusInfo, 330
pTime
_SLQSOMADMSessionInfo, 93
SwiOTAMsg_s, 1004
pTimeLength
_SLQSOMADMSessionInfo, 93
pTimeSrc
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pTimeStamp
getDyingGaspStatistics, 284

pack_pds_SLQSPDSInjectPositionData_t, 708
packgetDyingGaspStatistics, 787
PDSPositionData, 791
pTimeType
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
pTimeUnc
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pTimeZone
nasNetworkTime, 588
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1226
pTimeout
unpack_pds_GetPDSDefaults_t, 1243
pTimer
NasSwiIndReg, 605
pack_nas_SLQSNasSwiIndicationRegister_t, 682
pTimerSIPReg
GetSIPConfigResp, 297
imsSIPConfigInfo, 339
SetSIPConfigReq, 933
pTimerT1
GetSIPConfigResp, 297
imsSIPConfigInfo, 339
SetSIPConfigReq, 933
pTimerT2
GetSIPConfigResp, 297
imsSIPConfigInfo, 339
SetSIPConfigReq, 933
pTimerTf
GetSIPConfigResp, 297
imsSIPConfigInfo, 339
SetSIPConfigReq, 933
pTimerVal
pack_voice_SLQSVoiceSetSUPSService_t, 759
voiceSetSUPSServiceReq, 1490
pTimestampAge
LocInjectPositionReq, 408
pTimestampUtc
LocInjectPositionReq, 408
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pTokenBucket
swiQosFlow, 1013
pTos
swiQosFilter, 1010
pTotalBytesRX
QosEventInfo, 868
slqsWdsEventInfo, 958
pTotalBytesTX
QosEventInfo, 868
slqsWdsEventInfo, 958
pTotalNum
cdmaMsgDecodingParamsExt, 172
wcdmaLongMsgDecodingParams, 1500
pTrackingStatus
unpack_pds_GetPDSState_t, 1244
pTrafficClass
swiQosFlow, 1013
pTranDstPort
swiQosFilter, 1010
pTranSrcPort
swiQosFilter, 1010
pTransLayerInfo
_getTransLayerInfoResp, 68
_transLayerInfoNotification, 114
sms_getTransLayerInfo, 961
unpack_sms_SLQSTransLayerInfoCallback_ind_t, 1286
pTransferRouteMTMessageInfo
SMSEventInfo_s, 978
pTransferStatInd
getDUNCallInfoReq, 279
pack_wds_SLQSGetDUNCallInfo_t, 774
pack_wds_SLQSWdsSetEventReport_t, 785
wdsSetEventReportReq, 1545
pTransferStatusReport
sms_setRoutesReq, 970
smsSetRoutesReq, 985
pTrueIMSI
nasGet3GPP2SubscriptionInfoResp, 569
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1220
pTxFilter
swiQosModifyReq, 1014
swiQosReq, 1015
pTxFlow
swiQosGranted, 1013
swiQosModifyReq, 1015
swiQosReq, 1015
pTxQFilter
QosFlowInfo, 869
pTxQFlowGranted
QosFlowInfo, 869
pTypeCode
USSDRespFNetwork, 1394
pUATI
GetHRPDStatsResp, 285
unpack_nas_SLQSSwiGetHRPDStats_t, 1233
pUDPDstPort
swiQosFilter, 1010
pUDPSrcPort
swiQosFilter, 1010
pUMTSCellID
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1222
pUMTSGrantedQoS
qmiWdsRunTimeSettings, 865
pUMTSInfo
nasCellLocationInfoResp, 567

unpack_nas_SLQSNasGetCellLocationInfo_t, 1222
pUMTSMinQoS
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 369
Profile3GPP, 816
pUMTSMinQosSigInd
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 369
Profile3GPP, 816
pUMTSReqQoS
LibpackProfile3GPP, 357
LibPackprofile_3GPP, 369
Profile3GPP, 816
pUMTSReqQoSSigInd
LibpackProfile3GPP, 358
LibPackprofile_3GPP, 369
Profile3GPP, 816
pURL
QmiCbkLocGetServerInd, 841
unpack_loc_GetServer_Ind_t, 1168
pURLAddr
LocSetServerReq, 411
pack_loc_SLQSLOCSetServer_t, 663
pUSBComp
USBCompConfig, 1390
USBCompParams, 1392
pUSSDData
USSDNoWaitIndicationInfo, 1393
pUSSDInfo
unpack_voice_SLQSOriginateUSSD_t, 1316
USSResp, 1395
pUSSInfo
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
pUTSvcRAT
imsaSvcStatusInfo, 335
pUTSvcStatus
imsaSvcStatusInfo, 335
pUUSInFo
pack_voice_SLQSVoiceDialCall_t, 747
voiceCallRequestParams, 1452
pUUSInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1451
pUimSlotsStatus
UIMGetSlotsStatusResp, 1075
unpack_uim_SLQSUIMGetSlotsStatus_t, 1305
pUname
_SLQSAVMSSettingsAPNInfo, 88
pUpdateCompleteStatus
_SLQSOMADMSessionInfo, 93
pUsbMTUSize
SwiDmsGetMTUResp, 999
unpack_swidms_SLQSSwiDmsGetMTU_t, 1290
pUser
pack_wds_SLQSStartDataSession_t, 782
pUserAcknowledgementReq
cdmaMsgDecodingParams, 169
cdmaMsgDecodingParamsExt, 172
pUserConfigEvents
imsCfgIndRegisterInfo, 337
pUserData
slqssendasyncsmsparams_s, 948
sms_sendAsyncsmsParams, 968
pUserId
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pUserIdSize
LibpackProfile3GPP2, 363
LibPackprofile_3GPP2, 374
Profile3GPP2, 821
pUsername
LibpackProfile3GPP, 358
LibPackprofile_3GPP, 369
pack_wds_SetDefaultProfile_t, 767
Profile3GPP, 816
qmiWdsRunTimeSettings, 865
ssdatasession_params, 992
pUsernameSize
LibpackProfile3GPP, 358
LibPackprofile_3GPP, 369
Profile3GPP, 816
pUtServiceRat
IMSAServiceStatus, 333
pUtServiceStatus
IMSAServiceStatus, 333
pV4sessionId
WdsByteTotals, 1529
WdsConnectionRate, 1531
WdsPktStatisticsResp, 1542
pV6sessionId
WdsByteTotals, 1529
WdsConnectionRate, 1531
WdsPktStatisticsResp, 1542
pVOIPSvcRAT
imsaSvcStatusInfo, 335
pVOIPSvcStatus
imsaSvcStatusInfo, 336
pVTSvcRAT
imsaSvcStatusInfo, 336
pVTSvcStatus
imsaSvcStatusInfo, 336
pVerboseFailReasonType
unpack_wds_SLQSStartDataSession_t, 1385
pVerboseFailureReason
unpack_wds_SLQSStartDataSession_t, 1385
pVersionString
_SLQSSwiGetOSInfoParams, 99
_SLQSSwiSetOSInfoParams, 101
pVertConfidence
LocInjectPositionReq, 408
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850

unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pVertReliability
LocInjectPositionReq, 408
QmiCbkLocBestAvailPosInd, 838
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1162
unpack_loc_PositionRpt_Ind_t, 1178
pVertUnc
LocInjectPositionReq, 408
QmiCbkLocBestAvailPosInd, 839
QmiCbkLocPositionReportInd, 850
unpack_loc_BestAvailPos_Ind_t, 1163
unpack_loc_PositionRpt_Ind_t, 1178
pVerticalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
pVerticalUnc
pack_pds_SLQSPDSInjectPositionData_t, 708
PDSPositionData, 791
pVoIPConfigEvents
imsCfgIndRegisterInfo, 337
pVoiceDomainPref
pack_voice_SLQSVoiceGetConfig_t, 751
voiceGetConfigReq, 1472
pVoiceDomainPrefStatus
unpack_voice_SLQSVoiceSetConfig_t, 1342
voiceSetConfigResp, 1487
pVoicePrivacy
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1321
unpack_voice_SLQSVoiceGetCallInfo_t, 1327
voiceCallInfoResp, 1451
voiceGetAllCallInfo, 1458
pVoipServiceRat
IMSAServiceStatus, 333
pVoipServiceStatus
IMSAServiceStatus, 333
pVoltReport
pack_dms_SwiSetEventReport_t, 642
pVolume
SetM2MAudioProfileReq, 923
pVsServiceRat
IMSAServiceStatus, 333
pVsServiceStatus
IMSAServiceStatus, 333
pVtServiceRat
IMSAServiceStatus, 333
pVtServiceStatus
IMSAServiceStatus, 334
pWCDMABER
GetErrRateResp, 284
unpack_nas_SLQSGetErrorRate_t, 1200
pWCDMACallBarringSysInfo
nasGetSysInfoResp, 577
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1239
pWCDMACipherDomainSysInfo
nasGetSysInfoResp, 577
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1214
unpack_nas_SLQSSysInfoCallback_ind_t, 1239
pWCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pWCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pWCDMAECIOThreshList
nas_WCDMAECIOThresh, 560
WCDMAECIOThresh, 1497
pWCDMAInfoLTENeighborCell
nasCellLocationInfoResp, 567
unpack_nas_SLQSNasGetCellLocationInfo_t, 1222
pWCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pWCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 676
setSignalStrengthInfo, 932
pWCDMARSSIThreshList
nas_WCDMARSSIThresh, 562
WCDMARSSIThresh, 1503
pWCDMASSInfo
nasGetSigInfoResp, 574
pWCDMASigInfo
nasSigInfo, 603
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1228
pWCDMASrvStatusInfo
nasGetSysInfoResp, 577
nasSysInfo, 608
unpack_nas_SLQSGetSysInfo_t, 1215
unpack_nas_SLQSSysInfoCallback_ind_t, 1239
pWCDMASysInfo
nasGetSysInfoResp, 577
nasSysInfo, 609
unpack_nas_SLQSGetSysInfo_t, 1215
unpack_nas_SLQSSysInfoCallback_ind_t, 1239
pWcdmaUARFCN
nasSwiGetChannelLockResp, 604
nasSwiSetChannelLockReq, 606
pack_nas_SLQSNASSwiSetChannelLock_t, 683
unpack_nas_SLQSNASSwiGetChannelLock_t, 1228
pWdisable
dmsSwiGetPCInfo, 244
pWifiState
pack_pds_SLQSSetPositionMethodState_t, 710
PDSPosMethodStateReq, 792
pXid
QmiCbkLocBestAvailPosInd, 839
unpack_loc_BestAvailPos_Ind_t, 1163
pXtraDataState
pack_pds_SLQSSetPositionMethodState_t, 710

PDSPosMethodStateReq, 792
pXtraTimeState
pack_pds_SLQSSetPositionMethodState_t, 710
PDSPosMethodStateReq, 792
pack_dms_ActivateAutomatic
dms.h, 1561
pack_dms_ActivateAutomatic_t, 630
actCode, 630
pack_dms_GetActivationState
dms.h, 1562
pack_dms_GetBandCapability
dms.h, 1562
pack_dms_GetCrashAction
dms.h, 1562
pack_dms_GetCustFeature
dms.h, 1563
pack_dms_GetCustFeaturesV2
dms.h, 1563
pack_dms_GetCustFeaturesV2_t, 630
cust_id, 631
list_type, 631
Tlvresult, 631
pack_dms_GetDeviceCap
dms.h, 1564
pack_dms_GetDeviceCapabilities
dms.h, 1564
pack_dms_GetDeviceHardwareRev
dms.h, 1564
pack_dms_GetDeviceMfr
dms.h, 1565
pack_dms_GetDeviceSerialNumbers
dms.h, 1565
pack_dms_GetFSN
dms.h, 1566
pack_dms_GetFirmwareInfo
dms.h, 1565
pack_dms_GetFirmwareRevision
dms.h, 1566
pack_dms_GetFirmwareRevisions
dms.h, 1566
pack_dms_GetHardwareRevision
dms.h, 1567
pack_dms_GetIMSI
dms.h, 1567
pack_dms_GetManufacturer
dms.h, 1567
pack_dms_GetModelID
dms.h, 1568
pack_dms_GetNetworkTime
dms.h, 1568
pack_dms_GetOfflineReason
dms.h, 1568
pack_dms_GetPRLVersion
dms.h, 1569
pack_dms_GetPower
dms.h, 1569
pack_dms_GetSerialNumbers
dms.h, 1569
pack_dms_GetUSBComp
dms.h, 1570
pack_dms_GetVoiceNumber
dms.h, 1570
pack_dms_ResetToFactoryDefaults
dms.h, 1570
pack_dms_ResetToFactoryDefaults_t, 631
spc, 631
pack_dms_SLQSDmsSwiGetPCInfo
dms.h, 1574
pack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1575
pack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1575
pack_dms_SLQSDmsSwiIndicationRegister_t, 638
resetInfoInd, 639
pack_dms_SLQSGetBandCapability
dms.h, 1575
pack_dms_SLQSGetERIFile
dms.h, 1576
pack_dms_SLQSGetPowerSaveModeConfig
dms.h, 1576
pack_dms_SLQSSetPowerSaveModeConfig
dms.h, 1576
pack_dms_SLQSSetPowerSaveModeConfig_t, 639
pActiveTimer, 639
pPeriodicUpdateTimer, 639
pPsmEnableState, 639
pack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1577
pack_dms_SLQSSwiGetCrashInfo
dms.h, 1577
pack_dms_SLQSSwiGetCrashInfo_t, 639
clear, 640
pack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1578
pack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1578
pack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1578
pack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1579
pack_dms_SLQSSwiGetHostDevInfo
dms.h, 1579
pack_dms_SLQSSwiGetOSInfo
dms.h, 1579
pack_dms_SLQSSwiGetSerialNoExt
dms.h, 1580
pack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1580
pack_dms_SLQSSwiSetDyingGaspCfg_t, 640
pDestSMSContent, 640
pDestSMSNum, 640
pack_dms_SLQSSwiSetHostDevInfo
dms.h, 1580
pack_dms_SLQSSwiSetHostDevInfo_t, 640
manString, 641
modelString, 641

plasmaIDString, 641
swVerString, 641
pack_dms_SLQSSwiSetOSInfo
dms.h, 1581
pack_dms_SLQSSwiSetOSInfo_t, 641
nameString, 642
versionString, 642
pack_dms_SLQSUIMGetState
dms.h, 1581
pack_dms_SetActivationStatusCallback
dms.h, 1571
pack_dms_SetActivationStatusCallback_t, 631
activationState, 632
pack_dms_SetCrashAction
dms.h, 1571
pack_dms_SetCrashAction_t, 632
crashAction, 632
pack_dms_SetCustFeature
dms.h, 1572
pack_dms_SetCustFeature_t, 632
DHCPRelayEnabled, 634
DisableIMSI, 634
GPSLPM, 634
GPSSel, 634
GpsEnable, 634
IPFamSupport, 634
IsVoiceEnabled, 634
RMAutoConnect, 634
SMSSupport, 634
pack_dms_SetCustFeaturesV2
dms.h, 1572
pack_dms_SetCustFeaturesV2_t, 634
cust_id, 635
cust_value, 635
Tlvresult, 635
value_length, 635
pack_dms_SetEventReport
dms.h, 1572
pack_dms_SetEventReport_t, 635
mode, 635
pack_dms_SetFirmwarePreference
dms.h, 1573
pack_dms_SetIndicationRegister
dms.h, 1573
pack_dms_SetIndicationRegister_t, 636
PSMCfgChangeInfo, 636
PSMStatus, 636
RptIMSCapability, 636
pack_dms_SetPower
dms.h, 1573
pack_dms_SetPower_t, 636
mode, 637
Tlvresult, 637
pack_dms_SetUSBComp
dms.h, 1574
pack_dms_SetUSBComp_t, 637
Tlvresult, 638
USBComp, 638
pack_dms_SwiSetEventReport
dms.h, 1581
pack_dms_SwiSetEventReport_t, 642
pTempReport, 642
pVoltReport, 642
pack_dms_UIMChangePIN
dms.h, 1582
pack_dms_UIMChangePIN_t, 642
id, 643
newValue, 643
oldValue, 643
pack_dms_UIMGetControlKeyStatus
dms.h, 1582
pack_dms_UIMGetControlKeyStatus_t, 643
facility, 643
pack_dms_UIMGetICCID
dms.h, 1582
pack_dms_UIMGetICCID_t, 643
Tlvresult, 644
pack_dms_UIMGetPINStatus
dms.h, 1583
pack_dms_UIMSetControlKeyProtection
dms.h, 1583
pack_dms_UIMSetControlKeyProtection_t, 644
facility, 644
facilityCk, 644
facilityState, 644
pack_dms_UIMSetPINProtection
dms.h, 1584
pack_dms_UIMSetPINProtection_t, 645
bEnable, 645
id, 645
value, 645
pack_dms_UIMUnblockControlKey
dms.h, 1584
pack_dms_UIMUnblockControlKey_t, 645
facility, 646
facilityCk, 646
pack_dms_UIMUnblockPIN
dms.h, 1584
pack_dms_UIMUnblockPIN_t, 646
id, 646
newPin, 646
pukValue, 646
pack_dms_UIMVerifyPIN
dms.h, 1585
pack_dms_UIMVerifyPIN_t, 647
id, 647
value, 647
pack_dms_ValidateSPC
dms.h, 1585
pack_fms_GetImagesPreference
fms.h, 1612
pack_fms_GetImagesPreference_t, 647
Tlvresult, 647
pack_fms_GetStoredImages
fms.h, 1613
pack_fms_GetStoredImages_t, 647

Tlvresult, 648
pack_fms_SetImagesPreference
fms.h, 1613
pack_fms_SetImagesPreference_t, 648
bForceDownload, 649
imageListSize, 649
modemindex, 649
pImageList, 649
Tlvresult, 649
pack_loc_Delete_Assist_Data_t, 649
pBdsSVInfo, 649
pCellDb, 649
pClkInfo, 649
pGnssData, 650
pSVInfo, 650
Tlvresult, 650
pack_loc_DeleteAssistData
loc.h, 1620
pack_loc_EventRegister
loc.h, 1620
pack_loc_EventRegister_t, 650
eventRegister, 652
Tlvresult, 652
pack_loc_SLQSLOCGetBestAvailPos
loc.h, 1621
pack_loc_SLQSLOCGetBestAvailPos_t, 653
Tlvresult, 653
xid, 653
pack_loc_SLQSLOCGetServer
loc.h, 1622
pack_loc_SLQSLOCGetServer_t, 654
has_serverAddrTypeMask, 654
serverAddrTypeMask, 654
serverType, 654
pack_loc_SLQSLOCInjectPosition
loc.h, 1622
pack_loc_SLQSLOCInjectPosition_t, 654
altitudeSrcInfo, 658
altitudeWrtEllipsoid, 658
altitudeWrtMeanSeaLevel, 658
has_altitudeSrcInfo, 658
has_altitudeWrtEllipsoid, 658
has_horConfidence, 658
has_horReliability, 658
has_horUncCircular, 658
has_latitude, 658
has_longitude, 658
has_positionSrc, 658
has_rawHorConfidence, 658
has_rawHorUncCircular, 658
has_timestampAge, 658
has_timestampUtc, 658
has_vertConfidence, 658
has_vertRelicability, 658
has_vertUnc, 659
horConfidence, 659
horReliability, 659
horUncCircular, 659
latitude, 659
longitude, 659
positionSrc, 659
rawHorConfidence, 659
rawHorUncCircular, 659
timestampAge, 659
timestampUtc, 659
vertConfidence, 659
vertReliability, 659
vertUnc, 659
pack_loc_SLQSLOCInjectSensorData
loc.h, 1622
pack_loc_SLQSLOCInjectSensorData_t, 659
accelTemp, 660
acceleroData, 660
acceleroTimeSrc, 660
gyroData, 660
gyroTemp, 660
gyroTimeSrc, 661
has_accelTemp, 661
has_acceleroTimeSrc, 661
has_accleroData, 661
has_gyroData, 661
has_gyroTemp, 661
has_gyroTimeSrc, 661
has_opaqueId, 661
opaqueId, 661
pack_loc_SLQSLOCInjectUTCTime
loc.h, 1623
pack_loc_SLQSLOCInjectUTCTime_t, 661
timeMsec, 661
timeUncMsec, 661
pack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1623
pack_loc_SLQSLOCSetCradleMountConfig_t, 661
confidence, 662
has_confidence, 662
state, 662
pack_loc_SLQSLOCSetServer
loc.h, 1624
pack_loc_SLQSLOCSetServer_t, 662
pIPv4Config, 663
pIPv6Config, 663
pURLAddr, 663
serverType, 663
pack_loc_SetExtPowerState
loc.h, 1620
pack_loc_SetExtPowerState_t, 652
extPowerState, 652
Tlvresult, 652
pack_loc_SetOperationMode
loc.h, 1621
pack_loc_SetOperationMode_t, 652
mode, 653
Tlvresult, 653
pack_loc_Start
loc.h, 1624
pack_loc_Start_t, 663

pApplicationInfo, 664
pConfigAltitudeAssumed, 664
pHorizontalAccuracyLvl, 664
pIntermediateReportState, 664
pMinIntervalTime, 664
pRecurrenceType, 664
SessionId, 665
Tlvresult, 665
pack_loc_Stop
loc.h, 1624
pack_loc_Stop_t, 665
SessionId, 665
Tlvresult, 665
pack_nas_GetACCOLC
nas.h, 1645
pack_nas_GetANAAAAuthenticationStatus
nas.h, 1645
pack_nas_GetCDMANetworkParameters
nas.h, 1645
pack_nas_GetHomeNetwork
nas.h, 1646
pack_nas_GetHomeNetwork3GPP2
nas.h, 1646
pack_nas_GetNetworkPreference
nas.h, 1646
pack_nas_GetRFInfo
nas.h, 1646
pack_nas_GetServingNetwork
nas.h, 1647
pack_nas_GetServingNetworkCapabilities
nas.h, 1647
pack_nas_GetSignalStrengths
nas.h, 1647
pack_nas_InitiateDomainAttach
nas.h, 1648
pack_nas_InitiateDomainAttach_t, 665
action, 665
pack_nas_PerformNetworkScan
nas.h, 1648
pack_nas_SLQSConfigSigInfo
nas.h, 1650
pack_nas_SLQSConfigSigInfo_t, 668
pECIOThresh, 670
pIOThresh, 670
pLTESNRThresh, 670
pLTESigRptCfg, 670
pRSRPThresh, 670
pRSRQThresh, 670
pRSSIThresh, 670
pack_nas_SLQSGetErrorRate
nas.h, 1651
pack_nas_SLQSGetNetworkTime
nas.h, 1651
pack_nas_SLQSGetOperatorNameData
nas.h, 1651
pack_nas_SLQSGetPLMNName
nas.h, 1652
pack_nas_SLQSGetPLMNName_t, 670
mcc, 671
mnc, 671
pMncPcsStatus, 671
pack_nas_SLQSGetServingSystem
nas.h, 1652
pack_nas_SLQSGetSignalStrength
nas.h, 1652
pack_nas_SLQSGetSysInfo
nas.h, 1653
pack_nas_SLQSGetSysSelectionPref
nas.h, 1653
pack_nas_SLQSInitiateNetworkRegistration
nas.h, 1654
pack_nas_SLQSInitiateNetworkRegistration_t, 671
pChangeDuration, 671
pMNRInfo, 671
pMncPcsDigitStatus, 671
regAction, 672
pack_nas_SLQSNASGeteDRXParams
nas.h, 1655
pack_nas_SLQSNASSeteDRXParams
nas.h, 1657
pack_nas_SLQSNASSeteDRXParams_t, 680
pCycleLen, 681
pEdrxEnable, 681
pack_nas_SLQSNASSwiGetChannelLock
nas.h, 1657
pack_nas_SLQSNASSwiSetChannelLock
nas.h, 1658
pack_nas_SLQSNASSwiSetChannelLock_t, 682
pack_nas_SLQSNasConfigSigInfo2
nas.h, 1654
pack_nas_SLQSNasConfigSigInfo2_t, 672
pHDRIODelta, 675
pHDRIOThresh, 675
pLTESigRptConfig, 676
pack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1654
pack_nas_SLQSNasGet3GPP2Subscription_t, 676
namID, 677
pack_nas_SLQSNasGetCellLocationInfo
nas.h, 1655
pack_nas_SLQSNasGetHDRColorCode
nas.h, 1655
pack_nas_SLQSNasGetRFInfo
nas.h, 1656
pack_nas_SLQSNasGetSigInfo
nas.h, 1656
pack_nas_SLQSNasGetTxRxInfo
nas.h, 1656
pack_nas_SLQSNasGetTxRxInfo_t, 677
radio_if, 677
pack_nas_SLQSNasIndicationRegisterExt
nas.h, 1657
pack_nas_SLQSNasIndicationRegisterExt_t, 677
pDDTMInd, 679
pDualStandByPrefInd, 679
pErrorRateInd, 680

pHDRSessionCloseInd, 680
pLTECphyCa, 680
pManagedRoamingInd, 680
pNetworkRejectInd, 680
pNetworkTimeInd, 680
pServingSystemInd, 680
pSignalStrengthInd, 680
pSubscriptionInfoInd, 680
pSuppressSysInfoInd, 680
pSysInfoInd, 680
pSystemSelectionInd, 680
pack_nas_SLQSNasSwiIndicationRegister
nas.h, 1658
pack_nas_SLQSNasSwiIndicationRegister_t, 681
gsmUmtsDl, 682
gsmUmtsUl, 682
lteEmmDl, 682
lteEmmUl, 682
lteEsmDl, 682
lteEsmUl, 682
pRankIndicatorInd, 682
pTimer, 682
pack_nas_SLQSNasSwiModemStatus
nas.h, 1658
pack_nas_SLQSSetBandPreference
nas.h, 1659
pack_nas_SLQSSetBandPreference_t, 683
bandPref, 684
pack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1659
pack_nas_SLQSSetSignalStrengthsCallback_t, 685
bEnable, 685
pSigIndReq, 685
pack_nas_SLQSSetSysSelectionPref
nas.h, 1660
pack_nas_SLQSSetSysSelectionPref_t, 685
pAcqOrderPref, 690
pBandPref, 690
pCSGID, 690
pChgDuration, 690
pEmerMode, 690
pGWAcqOrderPref, 690
pLTEBandPref, 690
pModePref, 690
pNetSelPref, 690
pPRLPref, 690
pRAT, 690
pRoamPref, 690
pSrvDomainPref, 690
pSrvRegRestriction, 690
pTdscdmaBandPref, 691
pack_nas_SLQSSetSysSelectionPrefExt
nas.h, 1660
pack_nas_SLQSSetSysSelectionPrefExt_t, 691
pAcqOrderPref, 699
pBandPref, 699
pCSGID, 699
pChgDuration, 699
pCiotAcqOrderPref, 699
pCiotLteOpMode, 699
pCiotLteOpModePref, 699
pEmerMode, 699
pGWAcqOrderPref, 699
pLTEBandPref, 699
pLteM1BandPref, 699
pLteNb1BandPref, 699
pModePref, 699
pNetSelPref, 699
pNr5gBandPref, 699
pPRLPref, 699
pRAT, 699
pRoamPref, 700
pSrvDomainPref, 700
pSrvRegRestriction, 700
pTdscdmaBandPref, 700
pack_nas_SLQSSwiGetHDRPersonality
nas.h, 1660
pack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1661
pack_nas_SLQSSwiGetHRPDStats
nas.h, 1661
pack_nas_SLQSSwiGetLteCQI
nas.h, 1661
pack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1662
pack_nas_SLQSSwiNetworkDebug
nas.h, 1662
pack_nas_SLQSSwiPSDetach
nas.h, 1662
pack_nas_SLQSSwiPSDetach_t, 700
pDetachAction, 700
pack_nas_SetACCOLC
nas.h, 1649
pack_nas_SetACCOLC_t, 666
accolc, 666
spc, 666
pack_nas_SetCDMANetworkParameters
nas.h, 1649
pack_nas_SetCDMANetworkParameters_t, 666
pApplication, 667
pBroadcast, 667
pCustomSCP, 667
pForceRev0, 667
pProtocol, 667
pRoaming, 667
pSPC, 667
pack_nas_SetLURejectCallback
nas.h, 1649
pack_nas_SetNetworkPreference
nas.h, 1650
pack_nas_SetNetworkPreference_t, 668
Duration, 668
TechnologyPref, 668
Tlvresult, 668
pack_nas_SetRFInfoCallback
nas.h, 1650

pack_nas_SlqsGetLTECphyCAInfo
nas.h, 1651
pack_pds_ForceXTRADownload
pds.h, 1690
pack_pds_GetPDSDefaults
pds.h, 1690
pack_pds_GetPDSState
pds.h, 1691
pack_pds_GetPortAutomaticTracking
pds.h, 1691
pack_pds_GetServiceAutomaticTracking
pds.h, 1691
pack_pds_GetXTRAAutomaticDownload
pds.h, 1692
pack_pds_GetXTRANetwork
pds.h, 1692
pack_pds_GetXTRAValidity
pds.h, 1692
pack_pds_PDSInjectTimeReference
pds.h, 1693
pack_pds_PDSInjectTimeReference_t, 700
systemDiscontinuities, 700
systemTime, 701
pack_pds_ResetPDSData
pds.h, 1693
pack_pds_ResetPDSData_t, 701
pCellDataMask, 701
pGPSDataMask, 701
pack_pds_SLQSGetAGPSConfig
pds.h, 1696
pack_pds_SLQSGetAGPSConfig_t, 705
pNetworkMode, 706
pack_pds_SLQSGetGPSStateInfo
pds.h, 1696
pack_pds_SLQSPDSDeterminePosition
pds.h, 1697
pack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1697
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 706
forceFlag, 706
timeBase, 706
timeMsec, 706
timeUncMsec, 707
pack_pds_SLQSPDSInjectPositionData
pds.h, 1697
pack_pds_SLQSPDSInjectPositionData_t, 707
pAltitudeWrtEllipsoid, 708
pAltitudeWrtSealevel, 708
pHorizontalConfidence, 708
pHorizontalUncCircular, 708
pLatitude, 708
pLongitude, 708
pPositionSource, 708
pTimeStamp, 708
pTimeType, 708
pVerticalConfidence, 708
pVerticalUnc, 708
pack_pds_SLQSSetAGPSConfig
pds.h, 1698
pack_pds_SLQSSetAGPSConfig_t, 709
pNetworkMode, 709
pServerAddress, 709
pServerPort, 709
pServerURL, 709
pack_pds_SLQSSetPositionMethodState
pds.h, 1698
pack_pds_SLQSSetPositionMethodState_t, 709
pWifiState, 710
pXtraDataState, 710
pXtraTimeState, 710
pack_pds_SetEventReportCallback
pds.h, 1694
pack_pds_SetEventReportCallback_t, 702
posDataNmea, 702
rptPosData, 702
pack_pds_SetPDSDefaults
pds.h, 1694
pack_pds_SetPDSDefaults_t, 702
accuracy, 703
interval, 703
operation, 703
timeout, 703
pack_pds_SetPDSState
pds.h, 1694
pack_pds_SetPDSState_t, 703
enable, 703
pack_pds_SetPortAutomaticTracking
pds.h, 1695
pack_pds_SetPortAutomaticTracking_t, 703
bAuto, 704
pack_pds_SetServiceAutomaticTracking
pds.h, 1695
pack_pds_SetServiceAutomaticTracking_t, 704
bAuto, 704
pack_pds_SetXTRAAutomaticDownload
pds.h, 1695
pack_pds_SetXTRAAutomaticDownload_t, 704
bEnabled, 705
interval, 705
pack_pds_SetXTRANetwork
pds.h, 1696
pack_pds_SetXTRANetwork_t, 705
preference, 705
pack_pds_StartPDSTrackingSessionExt
pds.h, 1699
pack_pds_StartPDSTrackingSessionExt_t, 710
fixAccuracy, 711
fixCount, 711
fixInterval, 711
fixTimeout, 711
sessionControl, 711
sessionOperation, 711
sessionServerOption, 711
sessionType, 711
pack_pds_StopPDSTrackingSession

pds.h, 1699
pack_qmi_t, 711
msgid, 712
svc, 712
timeout, 712
xid, 712
pack_qos_BindDataPort
qos.h, 2152
pack_qos_BindDataPort_t, 712
pMuxID, 713
pPeripheralEndPointID, 713
pSIODataPort, 713
pack_qos_SLQSQosGetNetworkStatus
qos.h, 2152
pack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 2153
pack_qos_SLQSQosSwiReadApnExtraParams_t, 713
apnId, 713
pack_qos_SLQSQosSwiReadDataStats
qos.h, 2154
pack_qos_SLQSQosSwiReadDataStats_t, 713
apnId, 713
pack_qos_SLQSSetQosEventCallback
qos.h, 2154
pack_qos_SLQSSetQosEventCallback_t, 713
enable, 714
pack_rms_GetSMSWake
rms.h, 2160
pack_rms_SetSMSWake
rms.h, 2160
pack_rms_SetSMSWake_t, 714
enabled, 714
wake_mask, 714
pack_sar_SLQSGetRfSarState
sar.h, 2162
pack_sar_SLQSSetRfSarState
sar.h, 2162
pack_sar_SLQSSetRfSarState_t, 714
RfSarState, 715
pack_sms_GetSMSCAddress
sms.h, 2167
pack_sms_SLQSDeleteSMS
sms.h, 2169
pack_sms_SLQSDeleteSMS_t, 717
pMessageIndex, 718
pMessageMode, 718
pMessageTag, 718
storageType, 718
pack_sms_SLQSGetIndicationRegister
sms.h, 2169
pack_sms_SLQSGetMessageWaiting
sms.h, 2170
pack_sms_SLQSGetSMS
sms.h, 2170
pack_sms_SLQSGetSMS_t, 718
messageIndex, 719
pMessageMode, 719
storageType, 719
pack_sms_SLQSGetSMSList
sms.h, 2171
pack_sms_SLQSGetSMSList_t, 719
pMessageMode, 720
pRequestedTag, 720
storageType, 720
pack_sms_SLQSGetSmsBroadcastConfig
sms.h, 2170
pack_sms_SLQSGetSmsBroadcastConfig_t, 719
mode, 719
pack_sms_SLQSGetTransLayerInfo
sms.h, 2171
pack_sms_SLQSGetTransNWRegInfo
sms.h, 2172
pack_sms_SLQSModifySMSStatus
sms.h, 2172
pack_sms_SLQSModifySMSStatus_t, 720
messageIndex, 721
messageTag, 721
pMessageMode, 721
storageType, 721
pack_sms_SLQSSendAsyncSMS
sms.h, 2172
pack_sms_SLQSSendAsyncSMS_t, 721
pSendSmsParams, 721
pack_sms_SLQSSetIndicationRegister
sms.h, 2173
pack_sms_SLQSSetIndicationRegister_t, 721
pSetIndicationRegReq, 722
pack_sms_SLQSSetSmsBroadcastActivation
sms.h, 2173
pack_sms_SLQSSetSmsBroadcastActivation_t, 722
broadcastActivate, 722
mode, 722
pack_sms_SLQSSetSmsBroadcastConfig
sms.h, 2173
pack_sms_SLQSSetSmsBroadcastConfig_t, 722
mode, 723
pBroadcastConfig, 723
pCDMABroadcastConfig, 723
pack_sms_SLQSSetSmsStorage
sms.h, 2174
pack_sms_SLQSSetSmsStorage_t, 723
smsStorage, 723
pack_sms_SLQSSmsGetMaxStorageSize
sms.h, 2174
pack_sms_SLQSSmsGetMaxStorageSize_t, 724
pMaxStorageSizeReq, 724
pack_sms_SLQSSmsGetMessageProtocol
sms.h, 2174
pack_sms_SLQSSmsSetRoutes
sms.h, 2175
pack_sms_SLQSSmsSetRoutes_t, 724
pSetRoutesReq, 724
pack_sms_SLQSSwiGetSMSStorage
sms.h, 2175
pack_sms_SaveSMS
sms.h, 2168

pack_sms_SaveSMS_t, 715
messageFormat, 716
messageSize, 716
pMessage, 716
storageType, 716
pack_sms_SendSMS
sms.h, 2168
pack_sms_SendSMS_t, 716
messageFormat, 716
messageSize, 716
pLinktimer, 716
pMessage, 717
pack_sms_SetNewSMSCallback
sms.h, 2168
pack_sms_SetNewSMSCallback_t, 717
status, 717
pack_sms_SetSMSCAddress
sms.h, 2169
pack_sms_SetSMSCAddress_t, 717
pSMSCAddress, 717
pSMSCType, 717
pack_swidms_SLQSSwiDmsGetMTU
swidms.h, 2187
pack_swidms_SLQSSwiDmsGetUsbNetNum
swidms.h, 2188
pack_swidms_SLQSSwiDmsSetMTU
swidms.h, 2188
pack_swidms_SLQSSwiDmsSetMTU_t, 724
MTUSize, 725
pack_swidms_SLQSSwiDmsSetUsbNetNum
swidms.h, 2188
pack_swidms_SLQSSwiDmsSetUsbNetNum_t, 725
nUsbNetNum, 725
pack_swiloc_SwiLocGetAutoStart
swiloc.h, 2191
pack_swiloc_SwiLocSetAutoStart
swiloc.h, 2191
pack_swiloc_SwiLocSetAutoStart_t, 725
fix_rate, 727
fix_type, 727
function, 727
max_dist, 727
max_time, 727
set_fix_rate, 727
set_fix_type, 727
set_function, 727
set_max_dist, 727
set_max_time, 727
pack_swioma_SLQSOMADMAlertCallback
swioma.h, 2193
pack_swioma_SLQSOMADMCancelSession
swioma.h, 2193
pack_swioma_SLQSOMADMCancelSession_t, 727
sessionType, 727
pack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 2194
pack_swioma_SLQSOMADMGetSessionInfo_t, 727
SessionType, 728
pack_swioma_SLQSOMADMGetSettings
swioma.h, 2195
pack_swioma_SLQSOMADMSendSelection
swioma.h, 2195
pack_swioma_SLQSOMADMSendSelection_t, 728
pDeferTime, 728
pRejectReason, 728
selection, 728
pack_swioma_SLQSOMADMSetSettings
swioma.h, 2196
pack_swioma_SLQSOMADMSetSettings_t, 728
FOTAdownload, 729
pAutosdm, 729
pFwAutoCheck, 729
pack_swioma_SLQSOMADMStartSession
swioma.h, 2197
pack_swioma_SLQSOMADMStartSession_t, 729
sessionType, 730
pack_uim_ChangePin
uim.h, 2204
pack_uim_ChangePin_t, 730
changePIN, 731
EncryptedPIN1, 731
pIndicationToken, 731
pKeyReferenceID, 731
sessionInfo, 731
Tlvresult, 731
pack_uim_GetCardStatus
uim.h, 2204
pack_uim_ReadTransparent
uim.h, 2205
pack_uim_ReadTransparent_t, 731
fileIndex, 732
pEncryptData, 732
pIndicationToken, 732
readTransparent, 732
sessionInfo, 732
Tlvresult, 732
pack_uim_SLQSUIMAuthenticate
uim.h, 2205
pack_uim_SLQSUIMAuthenticate_t, 733
authData, 734
pIndicationToken, 734
sessionInfo, 734
pack_uim_SLQSUIMDepersonalization
uim.h, 2206
pack_uim_SLQSUIMDepersonalization_t, 734
depersonilisationInfo, 734
pack_uim_SLQSUIMEventRegister
uim.h, 2206
pack_uim_SLQSUIMEventRegister_t, 734
eventMask, 734
pack_uim_SLQSUIMGetConfiguration
uim.h, 2207
pack_uim_SLQSUIMGetConfiguration_t, 735
pConfigurationMask, 735
pack_uim_SLQSUIMGetFileAttributes
uim.h, 2207

pack_uim_SLQSUIMGetFileAttributes_t, 735
fileIndex, 736
pIndicationToken, 736
sessionInfo, 736
pack_uim_SLQSUIMGetSlotsStatus
uim.h, 2207
pack_uim_SLQSUIMPowerDown
uim.h, 2208
pack_uim_SLQSUIMPowerDown_t, 736
slot, 736
pack_uim_SLQSUIMPowerUp
uim.h, 2208
pack_uim_SLQSUIMPowerUp_t, 736
pIgnoreHotSwapSwitch, 737
slot, 737
pack_uim_SLQSUIMRefreshComplete
uim.h, 2208
pack_uim_SLQSUIMRefreshComplete_t, 737
refreshComplete, 737
sessionInfo, 737
pack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 2209
pack_uim_SLQSUIMRefreshGetLastEvent_t, 737
sessionInfo, 738
pack_uim_SLQSUIMRefreshOK
uim.h, 2209
pack_uim_SLQSUIMRefreshOK_t, 738
OKtoRefresh, 738
sessionInfo, 738
pack_uim_SLQSUIMRefreshRegister
uim.h, 2210
pack_uim_SLQSUIMRefreshRegister_t, 738
regRefresh, 739
sessionInfo, 739
pack_uim_SLQSUIMReset
uim.h, 2210
pack_uim_SLQSUIMSwitchSlot
uim.h, 2210
pack_uim_SLQSUIMSwitchSlot_t, 739
bLogicalSlot, 739
ulPhysicalSlot, 739
pack_uim_SetPinProtection
uim.h, 2205
pack_uim_SetPinProtection_t, 732
EncryptedPIN1, 733
pIndicationToken, 733
pKeyReferenceID, 733
pinProtection, 733
sessionInfo, 733
Tlvresult, 733
pack_uim_UnblockPin
uim.h, 2211
pack_uim_UnblockPin_t, 740
EncryptedPIN1, 740
pIndicationToken, 740
pKeyReferenceID, 740
pinProtection, 740
sessionInfo, 740
Tlvresult, 740
pack_uim_VerifyPin
uim.h, 2211
pack_uim_VerifyPin_t, 741
pEncryptedPIN1, 741
pIndicationToken, 741
pKeyReferenceID, 741
sessionInfo, 742
Tlvresult, 742
verifyPIN, 742
pack_voice_AnswerUSSD
voice.h, 2225
pack_voice_AnswerUSSD_t, 742
pInfo, 742
pack_voice_CancelUSSD
voice.h, 2225
pack_voice_OriginateUSSD
voice.h, 2226
pack_voice_OriginateUSSD_t, 742
pInfo, 742
pack_voice_SLQSOriginateUSSD
voice.h, 2226
pack_voice_SLQSOriginateUSSD_t, 742
ussDCS, 743
ussData, 743
ussLen, 743
pack_voice_SLQSVoiceALSSelectLine
voice.h, 2226
pack_voice_SLQSVoiceALSSelectLine_t, 743
lineValue, 743
pack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 2227
pack_voice_SLQSVoiceALSSetLineSwitching_t, 744
switchOption, 744
pack_voice_SLQSVoiceAnswerCall
voice.h, 2227
pack_voice_SLQSVoiceAnswerCall_t, 744
pCallId, 744
pack_voice_SLQSVoiceBindSubscription
voice.h, 2228
pack_voice_SLQSVoiceBindSubscription_t, 744
subsType, 745
pack_voice_SLQSVoiceBurstDTMF
voice.h, 2228
pack_voice_SLQSVoiceBurstDTMF_t, 745
pack_voice_SLQSVoiceDialCall
voice.h, 2228
pack_voice_SLQSVoiceDialCall_t, 745
callNumber, 747
pCLIRType, 747
pCUGInfo, 747
pCallPartySubAdd, 747
pCallType, 747
pEmergencyCategory, 747
pSvcType, 747
pUUSInFo, 747
pack_voice_SLQSVoiceEndCall
voice.h, 2229

pack_voice_SLQSVoiceEndCall_t, 747
pCallId, 747
pack_voice_SLQSVoiceGetAllCallInfo
voice.h, 2229
pack_voice_SLQSVoiceGetCLIP
voice.h, 2231
pack_voice_SLQSVoiceGetCLIR
voice.h, 2231
pack_voice_SLQSVoiceGetCNAP
voice.h, 2232
pack_voice_SLQSVoiceGetCOLP
voice.h, 2232
pack_voice_SLQSVoiceGetCOLR
voice.h, 2232
pack_voice_SLQSVoiceGetCallBarring
voice.h, 2229
pack_voice_SLQSVoiceGetCallBarring_t, 747
pSvcClass, 748
reason, 748
pack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 2230
pack_voice_SLQSVoiceGetCallForwardingStatus_t, 748
pSvcClass, 749
Reason, 749
pack_voice_SLQSVoiceGetCallInfo
voice.h, 2230
pack_voice_SLQSVoiceGetCallInfo_t, 749
callID, 749
pack_voice_SLQSVoiceGetCallWaiting
voice.h, 2231
pack_voice_SLQSVoiceGetCallWaiting_t, 749
pSvcClass, 750
pack_voice_SLQSVoiceGetConfig
voice.h, 2233
pack_voice_SLQSVoiceGetConfig_t, 750
pAMRStatus, 751
pAirTimer, 751
pAutoAnswer, 751
pNamID, 751
pPrefVoicePrivacy, 751
pPrefVoiceSO, 751
pRoamTimer, 751
pTTYMode, 751
pVoiceDomainPref, 751
pack_voice_SLQSVoiceIndicationRegister
voice.h, 2233
pack_voice_SLQSVoiceIndicationRegister_t, 752
pRegDTMFEvents, 752
pRegVoicePrivacyEvents, 752
pSuppsNotifEvents, 752
pack_voice_SLQSVoiceManageCalls
voice.h, 2233
pack_voice_SLQSVoiceManageCalls_t, 752
pCallID, 753
SUPSType, 753
pack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 2234
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 753
pack_voice_SLQSVoiceSendFlash
voice.h, 2234
pack_voice_SLQSVoiceSendFlash_t, 754
pCallID, 754
pFlashPayLd, 754
pFlashType, 754
pack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 2234
pack_voice_SLQSVoiceSetCallBarringPassword_t, 754
newPasswd, 755
newPasswdAgain, 755
oldPasswd, 755
Reason, 755
pack_voice_SLQSVoiceSetConfig
voice.h, 2235
pack_voice_SLQSVoiceSetConfig_t, 755
pAirTimerConfig, 757
pAutoAnswer, 757
pPrefVoiceDomain, 757
pPrefVoiceSO, 757
pRoamTimerConfig, 757
pTTYMode, 757
pack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 2235
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 757
privacyPref, 757
pack_voice_SLQSVoiceSetSUPSService
voice.h, 2236
pack_voice_SLQSVoiceSetSUPSService_t, 757
pCallBarringPasswd, 759
pCallForwardingNumber, 759
pServiceClass, 759
pTimerVal, 759
reason, 759
voiceSvc, 759
pack_voice_SLQSVoiceStartContDTMF
voice.h, 2236
pack_voice_SLQSVoiceStartContDTMF_t, 759
DTMFdigit, 760
pCallID, 760
pack_voice_SLQSVoiceStopContDTMF
voice.h, 2236
pack_voice_SLQSVoiceStopContDTMF_t, 760
callID, 760
pack_wds_DHCPv4ClientLeaseChange
wds.h, 2257
pack_wds_DHCPv4ClientLeaseChange_t, 760
pEnableNotification, 761
pack_wds_GetAutoconnect
wds.h, 2258
pack_wds_GetByteTotals
wds.h, 2258
pack_wds_GetConnectionRate
wds.h, 2258
pack_wds_GetDataBearerTechnology
wds.h, 2259
pack_wds_GetDefaultProfile
wds.h, 2259

pack_wds_GetDefaultProfile_t, 761
profiletype, 761
pack_wds_GetDefaultProfileNum
wds.h, 2260
pack_wds_GetDefaultProfileNum_t, 761
family, 761
type, 761
pack_wds_GetDormancyState
wds.h, 2260
pack_wds_GetDormancyState_t, 762
pack_wds_GetLastMobileIPError
wds.h, 2260
pack_wds_GetLastMobileIPError_t, 762
pack_wds_GetMobileIP
wds.h, 2261
pack_wds_GetMobileIP_t, 762
pack_wds_GetMobileIPProfile
wds.h, 2261
pack_wds_GetMobileIPProfile_t, 762
index, 763
pack_wds_GetPacketStatistics
wds.h, 2262
pack_wds_GetPacketStatistics_t, 763
pStatMask, 763
pack_wds_GetPacketStatus
wds.h, 2262
pack_wds_GetPacketStatus_t, 763
statmask, 764
pack_wds_GetSessionDuration
wds.h, 2263
pack_wds_GetSessionDuration_t, 764
pack_wds_GetSessionState
wds.h, 2263
pack_wds_RMSetTransferStatistics
wds.h, 2263
pack_wds_RMSetTransferStatistics_t, 764
RmTrasnferStaticsReq, 764
pack_wds_SLQSCreateProfile
wds.h, 2267
pack_wds_SLQSCreateProfile_t, 771
pCurProfile, 772
pProfileId, 772
pProfileType, 772
pack_wds_SLQSDeleteProfile
wds.h, 2267
pack_wds_SLQSDeleteProfile_t, 772
profileIndex, 772
profileType, 772
pack_wds_SLQSGet3GPPConfigItem
wds.h, 2268
pack_wds_SLQSGetCurrDataSystemStat
wds.h, 2268
pack_wds_SLQSGetCurrDataSystemStat_t, 772
pack_wds_SLQSGetCurrentChannelRate
wds.h, 2269
pack_wds_SLQSGetDUNCallInfo
wds.h, 2269
pack_wds_SLQSGetDUNCallInfo_t, 773
Mask, 774
pReportChannelRate, 774
pReportConnStatus, 774
pReportDataBearerTech, 774
pReportDormStatus, 774
pTransferStatInd, 774
pack_wds_SLQSGetDataBearerTechnology
wds.h, 2269
pack_wds_SLQSGetDataBearerTechnology_t, 773
pack_wds_SLQSGetProfileSettings
wds.h, 2270
pack_wds_SLQSGetProfileSettings_t, 774
ProfileId, 775
ProfileType, 775
pack_wds_SLQSGetRuntimeSettings
wds.h, 2270
pack_wds_SLQSGetRuntimeSettings_t, 775
pReqSettings, 776
pack_wds_SLQSModifyProfile
wds.h, 2271
pack_wds_SLQSModifyProfile_t, 776
curProfile, 777
pProfileId, 777
pProfileType, 777
pack_wds_SLQSResetPacketStatics
wds.h, 2271
pack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 2273
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 780
pProfileId, 780
pack_wds_SLQSSGetLoopback
wds.h, 2273
pack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 2274
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 780
pHwConfig, 781
pProfileId, 781
pack_wds_SLQSSSetLoopback
wds.h, 2274
pack_wds_SLQSSSetLoopback_t, 781
loopbackMode, 781
loopbackMultiplier, 781
pack_wds_SLQSSet3GPPConfigItem
wds.h, 2272
pack_wds_SLQSSet3GPPConfigItem_t, 777
p3gppRelease, 778
pProfileList, 778
pack_wds_SLQSSetIPFamilyPreference
wds.h, 2272
pack_wds_SLQSSetIPFamilyPreference_t, 779
IPFamilyPreference, 779
pack_wds_SLQSSetWdsEventCallback
wds.h, 2272
pack_wds_SLQSSetWdsEventCallback_t, 779
currentDataBearer, 780
dataBearer, 780
dataBearerTechExt, 780
dataSystemStatus, 780

dormancyStatus, 780
interval, 780
mobileIP, 780
transferStats, 780
pack_wds_SLQSStartDataSession
wds.h, 2274
pack_wds_SLQSStartDataSession_t, 781
pAuth, 782
pPass, 782
pTech, 782
pUser, 782
pprofileid3gpp, 782
pprofileid3gpp2, 782
pack_wds_SLQSStopDataSession
wds.h, 2275
pack_wds_SLQSStopDataSession_t, 783
psid, 783
pack_wds_SLQSWdsGoActive
wds.h, 2275
pack_wds_SLQSWdsGoDormant
wds.h, 2276
pack_wds_SLQSWdsSetEventReport
wds.h, 2276
pack_wds_SLQSWdsSetEventReport_t, 783
pCurrChannelRateInd, 784
pCurrDataBearerTechInd, 784
pCurrPrefDataSysInd, 784
pDataBearerTechInd, 784
pDataCallStatusChangeInd, 784
pDataSystemStatusChangeInd, 784
pDormancyStatusInd, 785
pMIPStatusInd, 785
pTransferStatInd, 785
pack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 2277
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 785
contextId, 785
contextType, 785
pack_wds_SetAutoconnect
wds.h, 2264
pack_wds_SetAutoconnect_t, 765
acroamsetting, 765
acsetting, 765
pack_wds_SetDefaultProfile
wds.h, 2264
pack_wds_SetDefaultProfile_t, 765
authentication, 766
ipAddress, 767
pApnname, 767
pName, 767
pPassword, 767
pUsername, 767
pdpType, 767
primaryDNS, 767
profileType, 767
secondaryDNS, 767
pack_wds_SetDefaultProfileNum
wds.h, 2265
pack_wds_SetDefaultProfileNum_t, 767
family, 767
index, 767
type, 767
pack_wds_SetMobileIP
wds.h, 2265
pack_wds_SetMobileIP_t, 768
mode, 768
pack_wds_SetMobileIPParameters
wds.h, 2265
pack_wds_SetMobileIPParameters_t, 768
pHA2002bis, 769
pHAAuthenticator, 769
pMode, 769
pReRegPeriod, 769
pReRegTraffic, 769
pRetryInterval, 769
pRetryLimit, 769
pSPC, 769
pack_wds_SetMobileIPProfile
wds.h, 2266
pack_wds_SetMobileIPProfile_t, 769
index, 770
pAAASPI, 771
pAddress, 771
pEnabled, 771
pHASPI, 771
pMNAAA, 771
pMNHA, 771
pNAI, 771
pPrimaryHA, 771
pRevTunneling, 771
pSecondaryHA, 771
spc, 771
pack_wds_SetMuxID
wds.h, 2266
PackCreateProfileOut, 785
ExtErrorCode, 786
ProfileIndex, 786
ProfileType, 786
package_name
omaDmFotaTlv, 625
omaDmFotaTlvExt, 627
unpack_omaDmFotaTlv_t, 1241
PackageID
_SwiAvmsEventInd, 102
packageSize
omaDmFotaTlvExt, 627
packageid_str
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
packetSrvStatus
qaGobiApiCbk.h, 1730
packetZone
CDMASysInfo, 179
nas_CDMASysInfo, 451
packetZoneValid
CDMASysInfo, 179

nas_CDMASysInfo, 451
packgetDyingGaspCfg, 786
pDestSMSContent, 786
pDestSMSNum, 786
packgetDyingGaspStatistics, 786
pSMSAttemptedFlag, 787
pTimeStamp, 787
path
fileInfo, 260
uim_fileInfo, 1057
pathLen
fileInfo, 260
uim_fileInfo, 1057
pbAuto
unpack_pds_GetPortAutomaticTracking_t, 1244
unpack_pds_GetServiceAutomaticTracking_t, 1245
pbEnabled
unpack_pds_GetXTRAAutomaticDownload_t, 1245
pbIMSRegistered
imsaRegStatusInfo, 331
pbPlatform
unpack_dms_GetOfflineReason_t, 1120
pci
cellParams, 181
ltePCI, 423
nas_cellParams, 453
nas_ltePCI, 491
nas_PhyCaAggPcellInfo, 511
nas_PhyCaAggScellIndType, 513
nas_PhyCaAggScellInfo, 516
nas_umtsLTENbrCell, 558
NASPhyCaAggPcellInfo, 590
NASPhyCaAggScellIndType, 592
NASPhyCaAggScellInfo, 593
PhyCaAggPcellInfo, 795
PhyCaAggScellIndType, 797
PhyCaAggScellInfo, 800
umtsLTENbrCell, 1093
pcsDigit
PlmnID, 802
pcsfFQDNAddress
PCSCFFQDNAddressList, 788
wds_PCSCFFQDNAddressList, 1522
pdpType
pack_wds_SetDefaultProfile_t, 767
pdptype
unpack_wds_GetDefaultProfile_t, 1357
pds.h, 1688
pack_pds_ForceXTRADownload, 1690
pack_pds_GetPDSDefaults, 1690
pack_pds_GetPDSState, 1691
pack_pds_GetPortAutomaticTracking, 1691
pack_pds_GetServiceAutomaticTracking, 1691
pack_pds_GetXTRAAutomaticDownload, 1692
pack_pds_GetXTRANetwork, 1692
pack_pds_GetXTRAValidity, 1692
pack_pds_PDSInjectTimeReference, 1693
pack_pds_ResetPDSData, 1693
pack_pds_SLQSGetAGPSConfig, 1696
pack_pds_SLQSGetGPSStateInfo, 1696
pack_pds_SLQSPDSDeterminePosition, 1697
pack_pds_SLQSPDSInjectAbsoluteTimeReference, 1697
pack_pds_SLQSPDSInjectPositionData, 1697
pack_pds_SLQSSetAGPSConfig, 1698
pack_pds_SLQSSetPositionMethodState, 1698
pack_pds_SetEventReportCallback, 1694
pack_pds_SetPDSDefaults, 1694
pack_pds_SetPDSState, 1694
pack_pds_SetPortAutomaticTracking, 1695
pack_pds_SetServiceAutomaticTracking, 1695
pack_pds_SetXTRAAutomaticDownload, 1695
pack_pds_SetXTRANetwork, 1696
pack_pds_StartPDSTrackingSessionExt, 1699
pack_pds_StopPDSTrackingSession, 1699
unpack_pds_ForceXTRADownload, 1699
unpack_pds_GetPDSDefaults, 1700
unpack_pds_GetPDSState, 1700
unpack_pds_GetPortAutomaticTracking, 1700
unpack_pds_GetServiceAutomaticTracking, 1701
unpack_pds_GetXTRAAutomaticDownload, 1701
unpack_pds_GetXTRANetwork, 1701
unpack_pds_GetXTRAValidity, 1702
unpack_pds_PDSInjectTimeReference, 1702
unpack_pds_ResetPDSData, 1702
unpack_pds_SLQSGetAGPSConfig, 1706
unpack_pds_SLQSGetGPSStateInfo, 1706
unpack_pds_SLQSPDSDeterminePosition, 1706
unpack_pds_SLQSPDSInjectAbsoluteTimeReference, 1707
unpack_pds_SLQSPDSInjectPositionData, 1707
unpack_pds_SLQSSetAGPSConfig, 1707
unpack_pds_SLQSSetPositionMethodState, 1708
unpack_pds_SetEventReport_Ind, 1703
unpack_pds_SetEventReportCallback, 1703
unpack_pds_SetPDSDefaults, 1703
unpack_pds_SetPDSState, 1704
unpack_pds_SetPdsState_Ind, 1704
unpack_pds_SetPortAutomaticTracking, 1704
unpack_pds_SetServiceAutomaticTracking, 1705
unpack_pds_SetXTRAAutomaticDownload, 1705
unpack_pds_SetXTRANetwork, 1705
unpack_pds_StartPDSTrackingSessionExt, 1708
unpack_pds_StopPDSTrackingSession, 1708
peakRate
tokenBucket, 1037
unpack_qos_tokenBucket_t, 1273
peakThroughputClass
GPRSQoS, 301
GPRSRequestedQoS, 302
LibPackGPRSRequestedQoS, 351
wds_GPRSQoS, 1519
peerNumberInfo, 792
callID, 793

numLen, 793
numPI, 793
numPlan, 793
numSI, 793
numType, 794
number, 793
PerformNetworkScan
qaGobiApiNas.h, 1919
PeriodicUpdateTimer
unpack_dms_PSMCfgChange_ind_t, 1127
periodicUpdateTimer
dms_PSMPeriodicUpdateTimerTlv, 236
PSMPeriodicUpdateTimerTlv, 827
PeriodicUpdateTimerInd
dms_PSMPeriodicUpdateTimerIndTlv, 235
PersistentLpm
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
PersistentTechPref
unpack_nas_GetNetworkPreference_t, 1190
persoFeature
appStats, 126
appStatus, 129
uim_appStatus, 1046
persoRetries
appStats, 126
appStatus, 129
uim_appStatus, 1046
persoState
appStats, 126
appStatus, 129
uim_appStatus, 1046
persoUnblockRetries
appStats, 126
appStatus, 129
uim_appStatus, 1046
personalizationStatus, 794
feature, 794
numFeatures, 794
unblockLeft, 794
verifyLeft, 794
pfailureCause
unpack_voice_SLQSOriginateUSSD_t, 1316
USSResp, 1395
phase
nas_rxInfo, 536
rxInfo, 891
PhyCaAggPcellInfo, 795
dl_bw_value, 795
freq, 795
iLTEbandValue, 795
NasGetLTECphyCaInfo, 571
pci, 795
TlvPresent, 795
PhyCaAggScellDlBw, 795
dl_bw_value, 796
NasGetLTECphyCaInfo, 571
TlvPresent, 796
PhyCaAggScellIndType, 796
freq, 797
NasGetLTECphyCaInfo, 572
pci, 797
scell_state, 797
TlvPresent, 797
PhyCaAggScellIndex, 796
NasGetLTECphyCaInfo, 571
scell_idx, 796
TlvPresent, 796
PhyCaAggScellInfo, 797
dl_bw_value, 800
freq, 800
iLTEbandValue, 800
NasGetLTECphyCaInfo, 572
pci, 800
scell_state, 800
TlvPresent, 800
PhysicalLayer
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
PilotEnergy
nas_NetworkStatEVDO, 507
NetworkStatEVDO, 616
PilotPN
nas_PilotSetParams, 517
PilotSetParams, 801
PilotSetData, 800
NumPilots, 800
pPilotSetInfo, 800
PilotSetParams, 801
PilotPN, 801
PilotStrength, 801
PilotType, 801
PilotStrength
nas_PilotSetParams, 517
PilotSetParams, 801
PilotType
nas_PilotSetParams, 517
PilotSetParams, 801
pin1Len
encryptedPIN1, 252
uim_encryptedPIN1, 1053
pin1Retries
appStats, 126
appStatus, 129
uim_appStatus, 1046
pin1State
appStats, 126
appStatus, 129
uim_appStatus, 1046
pin1Val
encryptedPIN1, 252
uim_encryptedPIN1, 1053
pin2Retries
appStats, 126
appStatus, 129
uim_appStatus, 1046
pin2State

appStats, 126
appStatus, 129
uim_appStatus, 1046
pinID
changeUIMPIN, 182
setPINProtection, 926
uim_changeUIMPIN, 1051
uim_setPINProtection, 1064
uim_unblockUIMPIN, 1067
uim_verifyUIMPIN, 1068
unblockUIMPIN, 1100
verifyUIMPIN, 1397
pinLen
changeUIMPIN, 182
uim_changeUIMPIN, 1051
uim_verifyUIMPIN, 1068
verifyUIMPIN, 1397
pinLength
setPINProtection, 926
uim_setPINProtection, 1064
pinOperation
setPINProtection, 926
uim_setPINProtection, 1064
pinProtection
pack_uim_SetPinProtection_t, 733
pack_uim_UnblockPin_t, 740
UIMSetPinProtectionReq, 1085
pinVal
changeUIMPIN, 182
uim_changeUIMPIN, 1051
uim_verifyUIMPIN, 1068
verifyUIMPIN, 1397
pinValue
setPINProtection, 926
uim_setPINProtection, 1064
PkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 2142
PkQmiNasPerformNetworkScan
qaNasPerformNetworkScan.h, 2143
PkgDescLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
PkgDescription
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
PkgName
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
PkgNameLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
pkgver
CurrentImgList, 204
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1144
PktErrRate
unpack_qos_swiQosFlow_t, 1272
pktErrRate, 801
exponent, 801
multiplier, 801
PktStatElmntsV4
WdsPktStatisticsResp, 1542
PktStatElmntsV6
WdsPktStatisticsResp, 1542
plasmaIDString
pack_dms_SLQSSwiSetHostDevInfo_t, 641
unpack_dms_SLQSSwiGetHostDevInfo_t, 1147
plmn
GERANInfo, 268
LTEInfoIntrafreq, 420
nas_GERANInfo, 468
nas_LTEInfoIntrafreq, 488
nas_UMTSInfo, 556
UMTSInfo, 1091
PlmnID, 802
mcc, 802
mnc, 802
pcsDigit, 802
TlvPresent, 802
polarityIncluded
lineCtrlInfo, 381
voice_lineCtrlInfo, 1433
Port, 805
port, 805
range, 805
port
IPv4Info, 346
IPv6Info, 349
loc_IPv4Info, 390
loc_IPv6Info, 391
Port, 805
unpack_qos_Port_t, 1257
posDataNmea
pack_pds_SetEventReportCallback_t, 702
posSrc
unpack_pds_SetEventReport_Ind_t, 1248
Position Determination Service (PDS), 45
PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 1248
positionSrc
pack_loc_SLQSLOCInjectPosition_t, 659
PowerOffMode
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
PowerSaveModeConfigReq, 805
pActiveTimer, 806
pPeriodicUpdateTimer, 806
pPsmEnableState, 806
PowerSaveModeConfigResp, 806
pActiveTimer, 807
pDurationDueToOOS, 807
pDurationThreshold, 807
pEarlyWakeupTime, 807
pPeriodicUpdateTimer, 807
pPsmEnableState, 807
pRandomizationWindow, 807
pprofileid3gpp
pack_wds_SLQSStartDataSession_t, 782

pprofileid3gpp2
pack_wds_SLQSStartDataSession_t, 782
prDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
prDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
prPCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
prPCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
Precedence
unpack_qos_swiQosFilter_t, 1268
precedenceClass
GPRSQoS, 301
GPRSRequestedQoS, 302
LibPackGPRSRequestedQoS, 351
wds_GPRSQoS, 1519
precisionDilution
qaGobiApiCbk.h, 1731
precisionDilution_s, 807
HDOP, 807
PDOP, 807
VDOP, 807
PrefImageList, 808
listEntries, 808
listSize, 808
prefNetwork
unpack_wds_SLQSGetCurrDataSystemStat_t, 1370
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
prefVoiceSO, 808
evrcCapability, 810
homeOrigVoiceSO, 810
homePageVoiceSO, 810
namID, 810
roamOrigVoiceSO, 810
preference
pack_pds_SetXTRANetwork_t, 705
Preferred
nas_QmiNas3GppNetworkInfo, 523
SlqsNas3GppNetworkInfo, 945
prefixLen
IPv6Addr, 347
unpack_qos_IPv6Addr_t, 1256
presentationInd
ECTNum, 251
remotePartyNum, 877
voice_ECTNum, 1429
voice_remotePartyNum, 1441
priChA
CDMAChannel, 164
nas_CDMAChannel, 446
priChB
CDMAChannel, 164
nas_CDMAChannel, 446
priSize
unpack_dms_GetFirmwareRevisions_t, 1116
pridns
unpack_wds_GetDefaultProfile_t, 1357
pridnsv6
unpack_wds_GetDefaultProfile_t, 1357
primaryDNS
pack_wds_SetDefaultProfile_t, 767
PrimaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1378
PrimaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1378
primaryHA
unpack_wds_GetMobileIPProfile_t, 1360
privacyPref
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 757
voiceSetPrefPrivacy, 1488
priver
CurrentImgList, 204
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1144
priversion_str
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
Profile
GetAudioPathConfigReq, 271
GetAudioProfileResp, 274
GetAudioVolTLBConfigReq, 275
GetM2MAudioProfileResp, 291
GetM2MAudioVolumeReq, 292
GetM2MAVMuteReq, 292
GetM2MSpkrGainReq, 294
SetAudioPathConfigReq, 912
SetAudioProfileReq, 913
SetAudioVolTLBConfigReq, 914
SetM2MAudioAVCFGReq, 921
SetM2MAudioProfileReq, 923
SetM2MAudioVolumeReq, 924
SetM2MAVMuteReq, 925
SetM2MSpkrGainReq, 925
Profile3GPP, 810
pAPNClass, 815
pAPNDisabledFlag, 815
pAPNName, 815
pAPNnameSize, 815
pAddrAllocPref, 815
pAuthenticationPref, 815
pGPRSMinimumQoS, 815
pGPRSRequestedQos, 815
pIPv4AddrPref, 815
pIPv6AddPref, 815
pImCnFlag, 815
pPDNInactivTimeout, 815
pPDPtype, 816
pPassword, 815
pPasswordSize, 815

pPcscfAddrUsingDhcp, 815
pPcscfAddrUsingPCO, 815
pPdpAccessConFlag, 815
pPdpContext, 815
pPdpDataCompType, 815
pPdpHdrCompType, 815
pPriDNSIPv4AddPref, 816
pPriDNSIPv6addpref, 816
pPrimaryID, 816
pProfilename, 816
pProfilenameSize, 816
pQosClassID, 816
pSecDNSIPv4AddPref, 816
pSecDNSIPv6addpref, 816
pSecondaryFlag, 816
pSupportEmergencyCalls, 816
pTFTID1Params, 816
pTFTID2Params, 816
pUMTSMinQoS, 816
pUMTSMinQosSigInd, 816
pUMTSReqQoS, 816
pUMTSReqQoSSigInd, 816
pUsername, 816
pUsernameSize, 816
Profile3GPP2, 816
pAPNClass3GPP2, 820
pAPNEnabled3GPP2, 820
pAllowLinger, 820
pApnString, 820
pApnStringSize, 820
pAppPriority, 820
pAppType, 820
pAuthPassword, 820
pAuthPasswordSize, 821
pAuthProtocol, 821
pAuthRetryCount, 821
pAuthTimeout, 821
pDataMode, 821
pDataRate, 821
pIpcpAckTimeout, 821
pIpcpCreqRetryCount, 821
pIsPcscfAddressNedded, 821
pLcpAckTimeout, 821
pLcpCreqRetryCount, 821
pNegoDnsSrvrPref, 821
pPDNInactivTimeout3GPP2, 821
pPdnType, 821
pPppSessCloseTimer1x, 821
pPppSessCloseTimerDO, 821
pPriV6DnsAddress, 821
pPrimaryV4DnsAddress, 821
pRATType, 821
pSecV6DnsAddress, 821
pSecondaryV4DnsAddress, 821
pUserId, 821
pUserIdSize, 821
ProfileID
_GetProfileSettingIn, 66
unpack_wds_SLQSGetRuntimeSettings_t, 1378
ProfileId
pack_wds_SLQSGetProfileSettings_t, 775
profileId
wds_DHCPProfileIdTlv, 1514
wds_DHCPv4ProfileId, 1516
WdsDHCPv4ProfileId, 1537
wdsDhcpv4ProfileId, 1537
ProfileId3GPP2
unpack_qos_swiQosFlow_t, 1272
ProfileIdTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1352
ProfileIdentifier, 821
profileIndex, 822
profileType, 822
ProfileIndex
PackCreateProfileOut, 786
profileIndex
pack_wds_SLQSDeleteProfile_t, 772
ProfileIdentifier, 822
SLQSDeleteProfileParams, 943
wds_ProfileIdentifier, 1523
profileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1370
ProfileName
unpack_wds_SLQSGetRuntimeSettings_t, 1378
ProfileType
_GetProfileSettingIn, 66
pack_wds_SLQSGetProfileSettings_t, 775
PackCreateProfileOut, 786
unpack_wds_SLQSGetProfileSettings_t, 1375
profileType
pack_wds_SetDefaultProfile_t, 767
pack_wds_SLQSDeleteProfile_t, 772
ProfileIdentifier, 822
SLQSDeleteProfileParams, 943
wds_DHCPProfileIdTlv, 1514
wds_DHCPv4ProfileId, 1517
wds_ProfileIdentifier, 1523
WdsDHCPv4ProfileId, 1538
wdsDhcpv4ProfileId, 1537
profiletype
pack_wds_GetDefaultProfile_t, 761
Protocol
unpack_nas_GetCDMANetworkParameters_t, 1186
protocolSubtypeElement, 822
AccessMac, 823
AuthProt, 823
ControlMac, 823
EncryptProt, 823
ForwardMac, 823
IdleState, 823
KeyExchange, 823
MultDisc, 823
PhysicalLayer, 823
ReverseMac, 823
SecProt, 823

VirtStream, 824
ProvisionStatus
CLIPResp, 185
CLIRResp, 186
CNAPResp, 188
COLPResp, 189
COLRResp, 189
voice_CLIPResp, 1420
voice_CLIRResp, 1420
voice_CNAPResp, 1421
voice_COLPResp, 1422
voice_COLRResp, 1423
psAttachState
nas_servSystem, 541
NASServingSystemInfo, 601
ServingSystemInfo, 906
servSystem, 908
psBarStatus
CallBarringSysInfo, 145
callBarStatus, 146
nas_CallBarringSysInfo, 444
nas_callBarStatus, 445
psState
CommInfo, 191
nas_CommInfo, 456
psc
nas_UMTSInfo, 556
nas_wcdmaCellInfo, 560
nas_WCDMASysInfo, 565
UMTSInfo, 1092
wcdmaCellInfo, 1496
WCDMASysInfo, 1506
pscValid
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
pscsfIPv4Addr
PCSCFIPv4ServerAddressList, 789
wds_PCSCFIPv4ServerAddressList, 1523
psetting
unpack_wds_GetAutoconnect_t, 1352
psid
pack_wds_SLQSStopDataSession_t, 783
unpack_wds_SLQSStartDataSession_t, 1385
puk1Retries
appStats, 126
appStatus, 129
uim_appStatus, 1046
puk2Retries
appStats, 126
appStatus, 129
uim_appStatus, 1046
pukLen
uim_unblockUIMPIN, 1067
unblockUIMPIN, 1100
pukVal
uim_unblockUIMPIN, 1068
unblockUIMPIN, 1100
pukValue
pack_dms_UIMUnblockPIN_t, 646
pv4sessionId
WdsIpAddressInfoReq, 1539
pv6sessionId
WdsIpAddressInfoReq, 1539
pwrDenialTime
lineCtrlInfo, 381
voice_lineCtrlInfo, 1433

QMI_LITE_WDS_CURRENT_CALL_DB_MASK
wds.h, 2257
QMI_LITE_WDS_LAST_CALL_DB_MASK
wds.h, 2257
QMI_SAR_RF_STATE_1
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_2
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_3
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_4
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_5
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_6
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_7
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_8
qaGobiApiSar.h, 1967
QMI_SAR_RF_STATE_DEFAULT
qaGobiApiSar.h, 1967
QMI_WDS_CURRENT_CALL_DB_MASK
qaGobiApiWds.h, 2105
QMI_WDS_LAST_CALL_DB_MASK
qaGobiApiWds.h, 2105
QCI
LibPackQosClassID, 375
QosClassID, 867
QCWWAN2kConnect
qaGobiApiDcs.h, 1806
QCWWAN2kEnumerateDevices
qaGobiApiDcs.h, 1806
QCWWAN2kGetConnectedDeviceID
qaGobiApiDcs.h, 1807
QCWWANConnect
qaGobiApiDcs.h, 1807
QCWWANDisconnect
qaGobiApiDcs.h, 1808
QCWWANEnumerateDevices
qaGobiApiDcs.h, 1808
QFlowState
unpack_qos_QosFlowInfo_t, 1258
QLIC
DeviceConfigDetail, 228
nas_DeviceConfigDetail, 463
qaCbkCatEventReportInd.h
eTLV_CBK_ALPHA_IDENTIFIER, 1710
eTLV_CBK_DISPLAY_TEXT, 1710
eTLV_CBK_END_PROACTIVE_SESSION, 1710

eTLV_CBK_GET_IN_KEY, 1710
eTLV_CBK_GET_INPUT, 1710
eTLV_CBK_LANGUAGE_NOTIFICATION, 1710
eTLV_CBK_REFRESH, 1710
eTLV_CBK_SELECT_ITEM, 1710
eTLV_CBK_SETUP_EVENT_LIST, 1710
eTLV_CBK_SETUP_IDLE_MODE_TEXT, 1710
eTLV_CBK_SETUP_MENU, 1710
eTLV_END_PROACTIVE_SESSION_LENGTH, 1710
eTLV_REFRESH_LENGTH, 1710
eTLV_SETUP_EVENT_LIST_LENGTH, 1710
qaCbkSwiOmaDmEventReportInd.h
eTLV_IND_OMA_DM_CONFIG, 1711
eTLV_IND_OMA_DM_FOTA, 1711
eTLV_IND_OMA_DM_NOT, 1711
qaGobiApiCbk.h
DEVICE_STATE_BOOT, 1766
DEVICE_STATE_DISCONNECTED, 1766
DEVICE_STATE_READY, 1766
DEVICE_STATE_SERIAL_ONLY, 1766
eQA_QMI_SVC_NA, 1766
eQA_QMI_SVC_NAS, 1766
eQA_QMI_SVC_WDS, 1766
SMS_EVENT_ETWS, 1766
SMS_EVENT_ETWS_PLMN, 1767
SMS_EVENT_MESSAGE_MODE, 1766
SMS_EVENT_MT_MESSAGE, 1766
SMS_EVENT_SMS_ON_IMS, 1767
SMS_EVENT_SMSC_ADDRESS, 1767
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE, 1766
SYS_SRV_DOMAIN_CAMPED, 1767
SYS_SRV_DOMAIN_CS_ONLY, 1767
SYS_SRV_DOMAIN_CS_PS, 1767
SYS_SRV_DOMAIN_NO_SRV, 1767
SYS_SRV_DOMAIN_PS_ONLY, 1767
qaGobiApiFms.h
eGOBI_DEV_SERIES_9X07_9X50, 1858
eGOBI_DEV_SERIES_9X15, 1858
eGOBI_DEV_SERIES_9X30, 1858
eGOBI_DEV_SERIES_G3K, 1858
eGOBI_DEV_SERIES_NON_GOBI, 1858
eGOBI_DEV_SERIES_SIERRA_GOBI, 1858
eGOBI_DEV_SERIES_UNKNOWN, 1858
eGOBI_IMG_CAR_3, 1859
eGOBI_IMG_CAR_AERIS, 1859
eGOBI_IMG_CAR_ALLTEL, 1858
eGOBI_IMG_CAR_AMX_TELCEL, 1859
eGOBI_IMG_CAR_ATT, 1858
eGOBI_IMG_CAR_BELL, 1858
eGOBI_IMG_CAR_BHARTI, 1859
eGOBI_IMG_CAR_BRASIL_VIVO, 1859
eGOBI_IMG_CAR_CHINA_MOBILE, 1859
eGOBI_IMG_CAR_CHINA_TELECOM, 1858
eGOBI_IMG_CAR_CHINA_UNICOM, 1858
eGOBI_IMG_CAR_EMOBILE, 1859
eGOBI_IMG_CAR_FACTORY, 1858
eGOBI_IMG_CAR_GENERIC, 1858
eGOBI_IMG_CAR_GENERIC_CDMA, 1858
eGOBI_IMG_CAR_IUSACELL, 1858
eGOBI_IMG_CAR_KDDI, 1858
eGOBI_IMG_CAR_KT_FREETEL, 1859
eGOBI_IMG_CAR_LEAP, 1858
eGOBI_IMG_CAR_METROPCS, 1858
eGOBI_IMG_CAR_NETCOM, 1859
eGOBI_IMG_CAR_NORF, 1858
eGOBI_IMG_CAR_NTT_DOCOMO, 1859
eGOBI_IMG_CAR_O2, 1859
eGOBI_IMG_CAR_OMH, 1858
eGOBI_IMG_CAR_ORANGE, 1859
eGOBI_IMG_CAR_RELIANCE1, 1858
eGOBI_IMG_CAR_RELIANCE2, 1859
eGOBI_IMG_CAR_ROGERS, 1859
eGOBI_IMG_CAR_SFR, 1859
eGOBI_IMG_CAR_SINGTEL_OPTUS, 1859
eGOBI_IMG_CAR_SK_TELCOM1, 1858
eGOBI_IMG_CAR_SK_TELCOM2, 1859
eGOBI_IMG_CAR_SOFTBANK, 1859
eGOBI_IMG_CAR_SPRINT, 1858
eGOBI_IMG_CAR_SWISSCOM, 1859
eGOBI_IMG_CAR_TATA, 1858
eGOBI_IMG_CAR_TELCOM_ITALIA, 1859
eGOBI_IMG_CAR_TELCOM_NZ, 1858
eGOBI_IMG_CAR_TELEFONICA, 1859
eGOBI_IMG_CAR_TELENOR, 1859
eGOBI_IMG_CAR_TELIASONERA, 1859
eGOBI_IMG_CAR_TELSTRA1, 1858
eGOBI_IMG_CAR_TELSTRA2, 1859
eGOBI_IMG_CAR_TELUS, 1858
eGOBI_IMG_CAR_TMOBILE, 1858
eGOBI_IMG_CAR_US, 1858
eGOBI_IMG_CAR_VERIZON, 1858
eGOBI_IMG_CAR_VODAFONE, 1858
eGOBI_IMG_GPS_ASSISTED, 1859
eGOBI_IMG_GPS_NO_XTRA, 1859
eGOBI_IMG_GPS_NONE, 1859
eGOBI_IMG_GPS_STAND_ALONE, 1859
eGOBI_IMG_REG_ASIA, 1859
eGOBI_IMG_REG_AUS, 1859
eGOBI_IMG_REG_EU, 1859
eGOBI_IMG_REG_GLOBAL, 1859
eGOBI_IMG_REG_LA, 1859
eGOBI_IMG_REG_NA, 1859
eGOBI_IMG_TECH_CDMA, 1860
eGOBI_IMG_TECH_UMTS, 1860
eGobi_DEV_SERIES_MC83, 1858
qaGobiApiNas.h
eNAS_LTE_CPHY_CA_BW_NRB_100, 1909
eNAS_LTE_CPHY_CA_BW_NRB_15, 1909
eNAS_LTE_CPHY_CA_BW_NRB_25, 1909
eNAS_LTE_CPHY_CA_BW_NRB_50, 1909
eNAS_LTE_CPHY_CA_BW_NRB_6, 1909
eNAS_LTE_CPHY_CA_BW_NRB_75, 1909
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1909

eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1909
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1909
eNAS_RADIO_IF_GSM, 1908
eNAS_RADIO_IF_LTE, 1908
eNAS_RADIO_IF_TDSCDMA, 1908
eNAS_RADIO_IF_UMTS, 1908
eSYS_SRV_DOMAIN_CAMPED, 1908
eSYS_SRV_DOMAIN_CS_ONLY, 1908
eSYS_SRV_DOMAIN_CS_PS, 1908
eSYS_SRV_DOMAIN_NO_SRV, 1908
eSYS_SRV_DOMAIN_PS_ONLY, 1908
eSYS_SRV_DOMAIN_UNKNOWN, 1908
qaGobiApiPds.h
eSetServiceAutomaticTrackingDisable, 1945
eSetServiceAutomaticTrackingEnable, 1945
qaGobiApiSar.h
QMI_SAR_RF_STATE_1, 1967
QMI_SAR_RF_STATE_2, 1967
QMI_SAR_RF_STATE_3, 1967
QMI_SAR_RF_STATE_4, 1967
QMI_SAR_RF_STATE_5, 1967
QMI_SAR_RF_STATE_6, 1967
QMI_SAR_RF_STATE_7, 1967
QMI_SAR_RF_STATE_8, 1967
QMI_SAR_RF_STATE_DEFAULT, 1967
qaGobiApiVoice.h
VOICE_SUPS_SRV_CLASS_DATA, 2081
VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC, 2081
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC, 2081
VOICE_SUPS_SRV_CLASS_FAX, 2081
VOICE_SUPS_SRV_CLASS_NONE, 2081
VOICE_SUPS_SRV_CLASS_PACKETACCESS, 2081
VOICE_SUPS_SRV_CLASS_PADACCESS, 2081
VOICE_SUPS_SRV_CLASS_SMS, 2081
VOICE_SUPS_SRV_CLASS_VOICE, 2081
qaGobiApiWds.h
QMI_WDS_CURRENT_CALL_DB_MASK, 2105
QMI_WDS_LAST_CALL_DB_MASK, 2105
qaNasGetRFBandInfo.h
eTLV_LTE_OPERATIONAL_MODE, 2142
eTLV_RF_BAND_INFO, 2142
eTLV_RF_BANDWIDTH_INFO, 2142
eTLV_RF_DEDICATED_BAND_INFO, 2142
eTLV_RF_EXT_FORMAT_BAND_INFO, 2142
qaNasPerformNetworkScan.h
eTLV_3GPP_NETWORK_INFO, 2143
qaCbkCatEventReportInd.h, 1709
UpkQmiCbkCatEventReportInd, 1710
qaCbkSwiOmaDmEventReportInd.h, 1710
UpkQmiCbkSwiOmaDmEventReportInd, 1711
UpkQmiCbkSwiOmaDmEventReportIndExt, 1711
qaGobiApiAudio.h, 1711
SLQSGetAudioPathConfig, 1712
SLQSGetAudioProfile, 1712
SLQSGetAudioVolTLBConfig, 1713
SLQSSetAudioPathConfig, 1713
SLQSSetAudioProfile, 1714
SLQSSetAudioVolTLBConfig, 1714
qaGobiApiCat.h, 1715
CATSendEnvelopeCommand, 1715
CATSendTerminalResponse, 1716
qaGobiApiCbk.h, 1716
accelAcceptReady, 1727
accelTempAcceptReady, 1727
CBK_ENABLE_EVENT, 1725
CBK_NOCHANGE, 1725
DEREGISTER_EVENT, 1725
DEREGISTER_SRV, 1725
device_state_enum, 1766
eDevState, 1727
eQaQMIService, 1766
eSMSEventType, 1728
EVENT_MASK_CARD, 1725
FIRST_INSTANCE, 1725
gpsTime, 1728
gyroAcceptReady, 1728
gyroTempAcceptReady, 1728
INVALID_INSTACNE, 1725
IPV4, 1725
IPV4V6, 1725
IPV6, 1725
iSLQSSetDUNCallInfoCallback, 1767
iSLQSSetSignalStrengthsCallback, 1767
iSLQSSetWdsFirstInstEventCallback, 1767
iSLQSSetWdsSecondInstEventCallback, 1767
iSLQSSetWdsThirdInstEventCallback, 1767
iSLQSSetWdsXferStatsFirstInstCallback, 1767
iSLQSSetWdsXferStatsSecondInstCallback, 1767
iSetCATEventCallback, 1767
iSetSignalStrengthCallback, 1767
LteNasReleaseInfo, 1729
MAX_NO_OF_CALLS, 1726
MAX_NO_OF_FILES, 1726
MAX_NO_OF_SLOTS, 1726
MAX_PATH_LENGTH, 1726
MAXUSSDLENGTH, 1726
modemTempNotification, 1729
NAS_SRV, 1726
NUM_OF_SET, 1726
PDS_SRV, 1726
packetSrvStatus, 1730
precisionDilution, 1731
REGISTER_EVENT, 1726
REGISTER_SRV, 1726
ResetInfoNotification, 1731
SECOND_INSTANCE, 1726
SLQSNasNetworkRejectCallback, 1782
SLQSNasNetworkTimeCallBack, 1783
SLQSNasSigInfo2CallBack, 1783
SLQSNasSigInfoCallBack, 1783
SLQSNasSwiOTAMessageCallback, 1784

SLQSNasSysInfoCallBack, 1784
SLQSNasTimerCallback, 1785
SLQSSetBandPreferenceCbk, 1785
SLQSSetDHCPv4ClientLeaseStatusCallback, 1786
SLQSSetDUNCallInfoCallback, 1786
SLQSSetDataSystemStatusCallback, 1786
SLQSSetIMSAPdpStatusCallback, 1787
SLQSSetIMSARatStatusCallback, 1787
SLQSSetIMSARegStatusCallback, 1788
SLQSSetIMSASvcStatusCallback, 1788
SLQSSetIMSSMSConfigCallback, 1788
SLQSSetIMSUserConfigCallback, 1789
SLQSSetIMSVoIPConfigCallback, 1789
SLQSSetLocInjectPositionCallback, 1790
SLQSSetLocInjectUTCTimeCallback, 1790
SLQSSetModemTempCallback, 1790
SLQSSetPacketSrvStatusCallback, 1791
SLQSSetQosEventCallback, 1791
SLQSSetQosNWStatusCallback, 1791
SLQSSetQosPriEventCallback, 1792
SLQSSetQosStatusCallback, 1792
SLQSSetRegMgrConfigCallback, 1793
SLQSSetSDKTerminatedCallback, 1793
SLQSSetSIPConfigCallback, 1795
SLQSSetSMSEventCallback, 1795
SLQSSetServingSystemCallback, 1794
SLQSSetSessionStateCallback, 1794
SLQSSetSignalStrengthsCallback, 1794
SLQSSetSwiGetResetInfoCallback, 1796
SLQSSetSwiHDRPersCallback, 1796
SLQSSetSysSelectionPrefCallBack, 1796
SLQSSetTransLayerInfoCallback, 1797
SLQSSetTransNWRegInfoCallback, 1797
SLQSSetWdsEventCallback, 1798
SLQSSetWdsTransferStatisticCallback, 1798
SLQSSwiAVMSEventReportCallback, 1799
SLQSTmdMitigationLvlRptCallback, 1799
SLQSUIMSetRefreshCallBack, 1799
SLQSUIMSetStatusChangeCallBack, 1800
SLQSVoiceInfoRecCallback, 1800
SLQSVoiceSetAllCallStatusCallBack, 1801
SLQSVoiceSetDTMFEventCallBack, 1801
SLQSVoiceSetOTASPStatusCallBack, 1801
SLQSVoiceSetPrivacyChangeCallBack, 1802
SLQSVoiceSetSUPSCallBack, 1802
SLQSVoiceSetSUPSNotificationCallback, 1803
SLQSWmsAsyncRawSendCallBack, 1803
SLQSWmsMemoryFullCallBack, 1804
SLQSWmsMessageWaitingCallBack, 1804
SMSAsyncRawSend, 1733
SMSCAddressInfo, 1734
SMSEtwsMessageInfo, 1734
SMSEtwsPlmnInfo, 1734
SMSEventInfo, 1734
SMSEventType, 1766
SMSMTMessageInfo, 1735
SMSMessageModeInfo, 1735
SMSOnIMSInfo, 1736
SMSTransferRouteMTMessageInfo, 1736
sensorDataUsage, 1732
sessionInformation, 1732
sessionInformationExt, 1733
SetActivationStatusCallback, 1767
SetCATEventCallback, 1767
SetCfgParamChangeCallback, 1768
SetDataCapabilitiesCallback, 1769
SetDeviceStateChangeCbk, 1769
SetFwDldCompletionCbk, 1769
SetGPSCallback, 1770
SetLURejectCallback, 1775
SetLocBestAvailPosCallback, 1770
SetLocCradleMountCallback, 1770
SetLocDeleteAssistDataCallback, 1771
SetLocEngineStateCallback, 1771
SetLocEventMaskNMEACallback, 1771
SetLocEventPositionCallback, 1772
SetLocEventTimeSyncCallback, 1772
SetLocGetServerCallback, 1772
SetLocGnssSvInfoCallback, 1772
SetLocInjectSensorDataCallback, 1773
SetLocInjectTimeCallback, 1773
SetLocOpModeCallback, 1773
SetLocSensorStreamingCallback, 1774
SetLocSetExtPowerConfigCallback, 1774
SetLocSetServerCallback, 1774
SetMobileIPStatusCallback, 1775
SetNMEACallback, 1777
SetNasLTECphyCaIndCallback, 1775
SetNetChangeCbk, 1776
SetNewSMSCallback, 1776
SetOMADMStateCallback, 1777
SetPDSStateCallback, 1778
SetPowerCallback, 1778
SetRFInfoCallback, 1778
SetRMTransferStatisticsCallback, 1779
SetRankIndicatorCallback, 1778
SetRoamingIndicatorCallback, 1779
SetSLQSOMADMAlertCallback, 1780
SetSLQSOMADMAlertCallbackExt, 1781
SetSignalStrengthCallback, 1779
SetUSSDNoWaitIndicationCallback, 1782
SetUSSDNotificationCallback, 1781
SetUSSDReleaseCallback, 1782
SetUimSlotStatusChangeCallback, 1781
svUsedforFix, 1736
SwiAvmsEventInd, 1737
SwiOTAMsg, 1737
tFNASwiLTECphyCaInfo, 1738
tFNASwiOTAMsg, 1738
tFNActivationStatus, 1737
tFNAllCallStatus, 1738
tFNAsyncRawSend, 1738
tFNBandPreference, 1739
tFNBestAvailPos, 1740
tFNCATEvent, 1741

tFNCbkUimSlotStatusChangeInd, 1741
tFNDHCPv4ClientLeaseStatus, 1743
tFNDTMFEvent, 1743
tFNDUNCallInfo, 1743
tFNDataCapabilities, 1741
tFNDataSysStatus, 1742
tFNDelAssistData, 1742
tFNDeviceStateChange, 1742
tFNEventNmea, 1743
tFNEventPosition, 1743
tFNFwDldCompletion, 1744
tFNGetServer, 1744
tFNGnssSvInfo, 1744
tFNHDRPersonaity, 1744
tFNImsRegMgrConfig, 1745
tFNImsSIPConfig, 1746
tFNImsSMSConfig, 1746
tFNImsUserConfig, 1746
tFNImsVoIPConfig, 1746
tFNImsaPdpStatus, 1745
tFNImsaRatStatus, 1745
tFNImsaRegStatus, 1745
tFNImsaSvcStatus, 1745
tFNInfoRec, 1746
tFNInjectPosition, 1747
tFNInjectSensorData, 1747
tFNInjectTimeStatus, 1747
tFNInjectUTCTime, 1747
tFNLUReject, 1748
tFNMemoryFull, 1750
tFNMessageWaiting, 1750
tFNMitiLvlRpt, 1750
tFNMobileIPStatus, 1750
tFNModemTempInfo, 1750
tFNNasTimer, 1751
tFNNet, 1751
tFNNetworkReject, 1751
tFNNetworkTime, 1751
tFNNewGPS, 1752
tFNNewNMEA, 1752
tFNNewRMTransferStatistics, 1752
tFNNewSMS, 1753
tFNOMADMState, 1753
tFNOTASPStatus, 1755
tFNOpMode, 1754
tFNPDSState, 1755
tFNPSMCfgParamChange, 1756
tFNPacketSrvState, 1755
tFNPower, 1756
tFNPrivacyChange, 1756
tFNQosNWStatus, 1756
tFNQosPriEvent, 1757
tFNQosStatus, 1757
tFNRFInfo, 1758
tFNRankIndicator, 1758
tFNResetInfo, 1758
tFNRoamingIndicator, 1759
tFNSDKTerminated, 1759
tFNSLQSOMADMAlert, 1761
tFNSLQSQOSEvent, 1761
tFNSLQSSessionState, 1762
tFNSLQSSignalStrengths, 1762
tFNSLQSWDSEvent, 1762
tFNSMSEvents, 1762
tFNSUPSInfo, 1762
tFNSUPSNotification, 1763
tFNSensorStreaming, 1759
tFNServingSystem, 1759
tFNSetCradleMount, 1760
tFNSetEngineState, 1760
tFNSetEventTimeSync, 1760
tFNSetExtPowerConfig, 1760
tFNSetServer, 1760
tFNSigInfo, 1761
tFNSignalStrength, 1761
tFNSwiAvmEventReport, 1763
tFNSysInfo, 1763
tFNSysSelectionPref, 1763
tFNUIMRefresh, 1764
tFNUIMStatusChangeInfo, 1764
tFNUSSDNoWaitIndication, 1765
tFNUSSDNotification, 1764
tFNUSSDRelease, 1765
tFNtransLayerInfo, 1764
tFNtransNWRegInfo, 1764
THIRD_INSTANCE, 1726
transLayerNotification, 1765
transNWRegInfoNotification, 1765
USSD_DCS_8BIT, 1726
USSD_DCS_ASCII, 1726
USSD_DCS_UCS2, 1726
VOICE_SRV, 1726
WDS_SRV, 1727
qaGobiApiDcs.h, 1804
GetQmapIsSupported, 1806
LEN, 1805
PORTNAM_LEN, 1805
QCWWAN2kConnect, 1806
QCWWAN2kEnumerateDevices, 1806
QCWWAN2kGetConnectedDeviceID, 1807
QCWWANConnect, 1807
QCWWANDisconnect, 1808
QCWWANEnumerateDevices, 1808
SLQSGetDeviceMode, 1810
SLQSGetNetStatistic, 1810
SLQSGetUsbPortNames, 1810
SLQSKillSDKProcess, 1811
SLQSSetLoggingMask, 1811
SLQSStart, 1812
SLQSStart_AVAgent, 1812
SLQSStartSrv, 1813
SetQmapIPTable, 1809
SetQmapIPv6Table, 1809
SetSDKImagePath, 1809
qaGobiApiDms.h, 1813
ActivateAutomatic, 1823

custFeaturesInfo, 1816
custFeaturesSetting, 1818
dmsCurrentPRLInfo, 1819
ERIFileparams, 1819
GetActivationState, 1823
GetDeviceCapabilities, 1824
GetFirmwareRevision, 1825
GetFirmwareRevisions, 1825
GetHardwareRevision, 1826
GetIMSI, 1827
GetManufacturer, 1827
GetModelID, 1828
GetNetworkTime, 1828
GetOfflineReason, 1829
GetPRLVersion, 1830
GetPower, 1829
GetSerialNumbers, 1830
GetVoiceNumber, 1831
MAX_CUST_ID_LEN, 1816
MAX_FSN_LENGTH, 1816
ResetToFactoryDefaults, 1832
SLQSDmsSwiGetPCInfo, 1833
SLQSDmsSwiGetResetInfo, 1833
SLQSDmsSwiIndicationRegister, 1833
SLQSGetBandCapabilities, 1833
SLQSGetBandCapability, 1833
SLQSGetCurrentPRLInfo, 1835
SLQSGetCustFeatures, 1835
SLQSGetCustFeaturesV2, 1836
SLQSGetERIFile, 1836
SLQSGetPowerSaveModeConfig, 1836
SLQSGetSerialNumbers, 1837
SLQSSetCustFeatures, 1837
SLQSSetCustFeaturesV2, 1838
SLQSSetPowerSaveModeConfig, 1838
SLQSSwiClearDyingGaspStatistics, 1839
SLQSSwiGetCrashAction, 1839
SLQSSwiGetCrashInfo, 1839
SLQSSwiGetDyingGaspCfg, 1840
SLQSSwiGetDyingGaspStatistics, 1840
SLQSSwiGetFSN, 1840
SLQSSwiGetFwUpdateStatus, 1841
SLQSSwiGetHostDevInfo, 1841
SLQSSwiGetHostDevInfoParams, 1820
SLQSSwiGetOSInfo, 1842
SLQSSwiGetOSInfoParams, 1821
SLQSSwiGetSerialNoExt, 1842
SLQSSwiGetSerialNoExtParams, 1821
SLQSSwiGetUSBComp, 1842
SLQSSwiSetCrashAction, 1843
SLQSSwiSetDyingGaspCfg, 1843
SLQSSwiSetHostDevInfo, 1844
SLQSSwiSetHostDevInfoParams, 1822
SLQSSwiSetOSInfo, 1844
SLQSSwiSetOSInfoParams, 1822
SLQSSwiSetUSBComp, 1844
SLQSUIMGetState, 1845
serialNumbersInfo, 1820
SetPower, 1832
UIMChangePIN, 1845
UIMGetControlKeyStatus, 1846
UIMGetICCID, 1847
UIMGetPINStatus, 1848
UIMSetControlKeyProtection, 1849
UIMSetPINProtection, 1849
UIMUnblockControlKey, 1850
UIMUnblockPIN, 1851
UIMVerifyPIN, 1852
ValidateSPC, 1852
qaGobiApiFms.h, 1853
BUILD_ID_LEN, 1857
DEVICE_OFFLINE, 1857
DEVICE_RESET, 1857
DEVICE_SHUTDOWN, 1857
DeleteStoredImage, 1860
DownloadToSlot, 1860
eGetDeviceSeries, 1861
eGobiDeviceSeries, 1858
eGobiImageCarrier, 1858
eGobiImageGPS, 1859
eGobiImageRegion, 1859
eGobiImageTech, 1859
GetImageStore, 1862
GetImagesPreference, 1861
GetStoredImages, 1862
IMG_ID_LEN, 1857
IMGDETAILS_LEN, 1857
PRI_UPDATE_FAIL, 1857
SLQSDownloadFirmwareToSlot, 1864
SLQSGetBootVersionNumber, 1865
SLQSGetFirmwareInfo, 1865
SLQSGetImageInfo, 1866
SLQSGetImageInfo_9x15, 1866
SLQSGetImageInfoMC77xx, 1867
SLQSGetImageInfoMC83xx, 1867
SLQSGetValidFwPriCombinations, 1868
SLQSIsSpkgFormatRequired, 1868
SLQSSetCrashStateCheckIgnore, 1869
SLQSSetSIMBasedImageSwitching, 1869
SLQSSetSpkgFormatRequired, 1869
SLQSSwiGetAllCarrierImages, 1870
SLQSSwiGetFirmwareCurr, 1870
SLQSUpgradeFirmware9x07, 1871
SLQSUpgradeFirmware9x15, 1871
SetImagePath, 1862
SetImagesPreference, 1863
UNIQUE_ID_LEN, 1857
upgrade_9x07_9x50_fw, 1872
upgrade_mc77xx_fw, 1872
UpgradeFirmware2k, 1873
qaGobiApiIms.h, 1873
SLQSGetIMSSMSConfig, 1874
SLQSGetIMSUserConfig, 1875
SLQSGetIMSVoIPConfig, 1875
SLQSGetRegMgrConfig, 1876
SLQSGetSIPConfig, 1876

SLQSImsConfigIndicationRegister, 1876
SLQSSetIMSSMSConfig, 1877
SLQSSetIMSUserConfig, 1877
SLQSSetIMSVoIPConfig, 1878
SLQSSetRegMgrConfig, 1878
SLQSSetSIPConfig, 1879
qaGobiApiImsa.h, 1880
SLQSGetIMSARegStatus, 1880
SLQSGetIMSAServiceStatus, 1881
SLQSGetIMSASupportedFields, 1881
SLQSGetIMSASupportedMsg, 1882
SLQSRegisterIMSAIndication, 1882
qaGobiApiLoc.h, 1883
SLQSLOCDelAssData, 1884
SLQSLOCEventRegister, 1884
SLQSLOCGetBestAvailPos, 1885
SLQSLOCGetServer, 1885
SLQSLOCInjectPosition, 1886
SLQSLOCInjectSensorData, 1886
SLQSLOCInjectUTCTime, 1887
SLQSLOCSetCradleMountConfig, 1887
SLQSLOCSetExtPowerState, 1887
SLQSLOCSetOpMode, 1888
SLQSLOCSetServer, 1888
SLQSLOCStart, 1889
SLQSLOCStop, 1889
SwiLocGetAutoStart, 1890
SwiLocSetAutoStart, 1890
qaGobiApiNas.h, 1890
eSYS_SRV_DOMAIN, 1908
GetACCOLC, 1909
GetANAAAAuthenticationStatus, 1909
GetCDMANetworkParameters, 1910
GetHomeNetwork, 1911
GetHomeNetwork3GPP2, 1912
GetNetworkPreference, 1914
GetRFInfo, 1915
GetServingNetwork, 1915
GetServingNetworkCapabilities, 1916
GetSignalStrengths, 1917
IMSI_M_S1_LENGTH, 1896
IMSI_M_S2_LENGTH, 1896
InitiateDomainAttach, 1918
InitiateNetworkRegistration, 1918
MAX_PILOT_SETS, 1896
NAM_NAME_LENGTH, 1896
PLMN_LENGTH, 1896
PerformNetworkScan, 1919
SLQSConfigSigInfo, 1922
SLQSGetErrorRate, 1923
SLQSGetNetworkTime, 1923
SLQSGetOperatorNameData, 1923
SLQSGetPLMNName, 1924
SLQSGetServingSystem, 1924
SLQSGetSignalStrength, 1925
SLQSGetSysSelectionPref, 1925
SLQSInitiateNetworkRegistration, 1926
SLQSNASGetLTECPHYCaInfo, 1928
SLQSNASGeteDRXParams, 1928
SLQSNASSeteDRXParams, 1932
SLQSNASSwiGetChannelLock, 1933
SLQSNASSwiSetChannelLock, 1934
SLQSNasConfigSigInfo2, 1926
SLQSNasGet3GPP2Subscription, 1927
SLQSNasGetCellLocationInfo, 1927
SLQSNasGetHDRColorCode, 1928
SLQSNasGetRFInfo, 1929
SLQSNasGetSigInfo, 1929
SLQSNasGetSysInfo, 1930
SLQSNasGetTxRxInfo, 1930
SLQSNasIndicationRegister, 1931
SLQSNasIndicationRegisterExt, 1931
SLQSNasIndicationRegisterLTECphyCa, 1932
SLQSNasSwiIndicationRegister, 1933
SLQSNasSwiModemStatus, 1933
SLQSPerformNetworkScan, 1934
SLQSSetBandPreference, 1935
SLQSSetSysSelectionPref, 1937
SLQSSetSysSelectionPrefExt, 1937
SLQSSwiGetHDRPersonality, 1937
SLQSSwiGetHDRProtSubtype, 1938
SLQSSwiGetHRPDStats, 1938
SLQSSwiGetLteCQI, 1939
SLQSSwiGetLteSccRxInfo, 1939
SLQSSwiNetworkDebug, 1939
SLQSSwiPSDetach, 1940
SetACCOLC, 1919
SetCDMANetworkParameters, 1920
SetNetworkPreference, 1921
SlqsNas3GppNetworkRAT, 1897
slqsNetworkScanInfo, 1897
sysSelectPrefInfo, 1898
sysSelectPrefParams, 1902
UATISIZE, 1897
qaGobiApiOmadm.h, 1940
OMADMCancelSession, 1941
OMADMGetPendingNIA, 1941
OMADMGetSessionInfo, 1942
OMADMStartSession, 1943
qaGobiApiPds.h, 1943
DEFAULTBYTEVALUE, 1945
DEFAULTLONGVALUE, 1945
DEFAULTWORDVALUE, 1945
ForceXTRADownload, 1945
GetPDSDefaults, 1945
GetPDSState, 1946
GetPortAutomaticTracking, 1946
GetServiceAutomaticTracking, 1947
GetXTRAAutomaticDownload, 1947
GetXTRANetwork, 1948
GetXTRAValidity, 1948
PDSInjectTimeReference, 1949
ResetPDSData, 1949
SLQSGetAGPSConfig, 1953
SLQSGetGPSStateInfo, 1954
SLQSPDSDeterminePosition, 1954

SLQSPDSInjectAbsoluteTimeReference, 1955
SLQSPDSInjectPositionData, 1955
SLQSSetAGPSConfig, 1956
SLQSSetPositionMethodState, 1956
SetPDSDefaults, 1950
SetPDSState, 1951
SetPortAutomaticTracking, 1951
SetServiceAutomaticTracking, 1952
SetXTRAAutomaticDownload, 1952
SetXTRANetwork, 1953
StartPDSTrackingSessionExt, 1957
StopPDSTrackingSession, 1958
qaGobiApiQos.h, 1958
SLQSQosGetFlowStatus, 1959
SLQSQosGetGranted, 1960
SLQSQosGetNWProf, 1961
SLQSQosGetNetworkStatus, 1960
SLQSQosModify, 1961
SLQSQosRel, 1962
SLQSQosReq, 1962
SLQSQosReset, 1963
SLQSQosResume, 1963
SLQSQosSuspend, 1964
SLQSQosSwiReadApnExtraParams, 1964
SLQSQosSwiReadDataStats, 1964
SetQosMuxID, 1959
qaGobiApiRms.h, 1965
GetSMSWake, 1965
SetSMSWake, 1966
qaGobiApiSar.h, 1966
eQMISARRFState, 1967
SLQSGetRfSarState, 1968
SLQSSetRfSarState, 1968
qaGobiApiSms.h, 1969
ABSOLUTE_VALIDITY, 1971
CONFIG_LEN, 1971
getIndicationRegResp, 1971
GetSMSCAddress, 1974
getTransLayerInfoResp, 1972
getTransNWRegInfoResp, 1972
MAX_SMS_ROUTES, 1971
NUM_OF_SET, 1971
qaQmi3GPP2BroadcastCfgInfo, 1972
qaQmi3GPPBroadcastCfgInfo, 1973
SLQSCDMADecodeMTTextMsg, 1977
SLQSCDMADecodeMTTextMsgExt, 1977
SLQSCDMAEncodeMOTextMsg, 1978
SLQSDeleteSMS, 1978
SLQSGetIndicationRegister, 1979
SLQSGetMessageWaiting, 1980
SLQSGetSMS, 1980
SLQSGetSMSList, 1982
SLQSGetSmsBroadcastConfig, 1981
SLQSGetTransLayerInfo, 1983
SLQSGetTransNWRegInfo, 1983
SLQSModifySMSStatus, 1983
SLQSSendAsyncSMS, 1984
SLQSSendLongSMS, 1984
SLQSSendSMS, 1985
SLQSSetIndicationRegister, 1986
SLQSSetSmsBroadcastActivation, 1986
SLQSSetSmsBroadcastConfig, 1987
SLQSSetSmsStorage, 1987
SLQSSmsGetMaxStorageSize, 1988
SLQSSmsGetMessageProtocol, 1988
SLQSSmsSetRoutes, 1989
SLQSSwiGetSMSStorage, 1989
SLQSWCDMADecodeLongTextMsg, 1990
SLQSWCDMADecodeMTTextMsg, 1990
SLQSWCDMAEncodeMOTextMsg, 1990
SaveSMS, 1975
SendSMS, 1976
setIndicationRegReq, 1973
SetSMSCAddress, 1976
TIME_DATE_BUF, 1971
TIME_STAMP_BUF, 1971
transLayerInfo, 1974
qaGobiApiSwi.h, 1991
SLQSGetPidof, 1991
SLQSGetSdkVersion, 1992
SLQSSendRawQMI, 1992
qaGobiApiSwiAudio.h, 1992
SLQSGetM2MAVMute, 1994
SLQSGetM2MAudioProfile, 1993
SLQSGetM2MAudioVolume, 1994
SLQSGetM2MSpkrGain, 1994
SLQSSetM2MAVMute, 1997
SLQSSetM2MAudioAVCFG, 1995
SLQSSetM2MAudioLPBK, 1995
SLQSSetM2MAudioNVDef, 1996
SLQSSetM2MAudioProfile, 1996
SLQSSetM2MAudioVolume, 1996
SLQSSetM2MSpkrGain, 1997
qaGobiApiSwiAvms.h, 1998
QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 2000
QmiSwiAvmsEventReportConfigResp, 2001
QmiSwiAvmsEventReportConnectionRequestResp, 2001
QmiSwiAvmsEventReportDataSessionStatusResp, 2002
QmiSwiAvmsEventReportHTTPStatusResp, 2002
QmiSwiAvmsEventReportNotificationResp, 2003
QmiSwiAvmsEventReportPackageIDResp, 2004
QmiSwiAvmsEventReportRegStatusResp, 2004
QmiSwiAvmsEventReportSessionTypeResp, 2004
QmiSwiAvmsEventReportWAMSParamChangeResp, 2005
QmiSwiAvmsSetSettingsAPNInfo, 2005
QmiSwiAvmsSetSettingsConnectionRetryTimers, 2005
QmiSwiAvmsSetSettingsPeriodInfo, 2005
SLQSAVMSGetSessionInfo, 2007
SLQSAVMSGetSettings, 2007
SLQSAVMSSelection, 2007
SLQSAVMSSetEventReport, 2008

SLQSAVMSSetSettings, 2008
SLQSAVMSSetSettingsIgnoreAutoReboot, 2009
SLQSAVMSSetSettingsReq, 2005
SLQSAVMSSettings, 2006
SLQSAVMSSettingsAPNInfo, 2006
SLQSAVMSSettingsConnectionRetryTimers, 2006
SLQSAVMSSettingsPeriodsInfo, 2006
SLQSAVMSStartSession, 2009
SLQSAVMSStopSession, 2010
qaGobiApiSwiDms.h, 2010
SLQSSwiDmsGetMTU, 2011
SLQSSwiDmsGetUsbNetNum, 2011
SLQSSwiDmsSetMTU, 2012
SLQSSwiDmsSetUsbNetNum, 2012
qaGobiApiSwiOmadms.h, 2012
SLQSOMADMCancelSession, 2018
SLQSOMADMGetSessionInfo, 2018
SLQSOMADMGetSettings, 2019
SLQSOMADMGetSettings2, 2019
SLQSOMADMSendSelection, 2020
SLQSOMADMSendSelection2, 2020
SLQSOMADMSessionInfo, 2013
SLQSOMADMSetSettings, 2021
SLQSOMADMSetSettings2, 2022
SLQSOMADMSetSettings3, 2022
SLQSOMADMSettings, 2015
SLQSOMADMSettingsReqParams, 2016
SLQSOMADMSettingsReqParams3, 2017
SLQSOMADMStartSession, 2022
SLQSOMADMStartSession2, 2023
qaGobiApiTableBandClasses.h, 2023
qaGobiApiTableCallControlReturnReasons.h, 2026
qaGobiApiTableCallEndReasons.h, 2027
qaGobiApiTableCarrierCodes.h, 2043
qaGobiApiTableCodingScheme.h, 2045
qaGobiApiTableGpsCapabilityCodes.h, 2048
qaGobiApiTablePowerModes.h, 2049
qaGobiApiTableRadioInterfaces.h, 2049
qaGobiApiTableRegionCodes.h, 2050
qaGobiApiTableServiceOptions.h, 2050
qaGobiApiTableSupServiceInfoClasses.h, 2053
qaGobiApiTableSwiAudio.h, 2053
qaGobiApiTableSwiOMADMSessionStatus.h, 2054
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 2055
qaGobiApiTableVoiceCallEndReasons.h, 2056
qaGobiApiTmd.h, 2063
SLQSTmdDeRegNotMitigationLvl, 2063
SLQSTmdGetMitigationDevList, 2064
SLQSTmdGetMitigationLvl, 2064
SLQSTmdRegNotMitigationLvl, 2064
qaGobiApiUim.h, 2065
MAX_ICCID_LENGTH, 2067
MAX_NO_OF_SLOTS, 2067
MAX_PATH_LENGTH, 2067
MAX_PUK_LENGTH, 2067
MAX_SLOTS_STATUS, 2067
SLQSUIMAuthenticate, 2067
SLQSUIMChangePin, 2068
SLQSUIMDepersonalization, 2068
SLQSUIMEventRegister, 2069
SLQSUIMGetCardStatus, 2069
SLQSUIMGetConfiguration, 2070
SLQSUIMGetFileAttributes, 2070
SLQSUIMGetSlotsStatus, 2071
SLQSUIMPowerDown, 2071
SLQSUIMPowerUp, 2072
SLQSUIMReadTransparent, 2072
SLQSUIMRefreshComplete, 2073
SLQSUIMRefreshGetLastEvent, 2073
SLQSUIMRefreshOK, 2074
SLQSUIMRefreshRegister, 2074
SLQSUIMReset, 2075
SLQSUIMSetPinProtection, 2075
SLQSUIMSwitchSlot, 2076
SLQSUIMUnblockPin, 2076
SLQSUIMVerifyPin, 2077
qaGobiApiVoice.h, 2077
AnswerUSSD, 2081
CancelUSSD, 2081
MAX_CALL_NO_LEN, 2080
MAX_NO_OF_CALLS, 2080
MAXUSSDLENGTH, 2080
OriginateUSSD, 2082
PASSWORD_LENGTH, 2081
SLQSOriginateUSSD, 2082
SLQSVoiceALSSelectLine, 2083
SLQSVoiceALSSetLineSwitching, 2083
SLQSVoiceAnswerCall, 2084
SLQSVoiceBindSubscription, 2084
SLQSVoiceBurstDTMF, 2085
SLQSVoiceDialCall, 2085
SLQSVoiceEndCall, 2086
SLQSVoiceGetAllCallInfo, 2086
SLQSVoiceGetCLIP, 2089
SLQSVoiceGetCLIR, 2090
SLQSVoiceGetCNAP, 2090
SLQSVoiceGetCOLP, 2091
SLQSVoiceGetCOLR, 2091
SLQSVoiceGetCallBarring, 2087
SLQSVoiceGetCallForwardingStatus, 2087
SLQSVoiceGetCallInfo, 2088
SLQSVoiceGetCallWaiting, 2088
SLQSVoiceGetConfig, 2092
SLQSVoiceIndicationRegister, 2092
SLQSVoiceManageCalls, 2093
SLQSVoiceOrigUSSDNoWait, 2093
SLQSVoiceSendFlash, 2094
SLQSVoiceSetCallBarringPassword, 2094
SLQSVoiceSetConfig, 2095
SLQSVoiceSetPreferredPrivacy, 2095
SLQSVoiceSetSUPSService, 2096
SLQSVoiceStartContDTMF, 2097
SLQSVoiceStopContDTMF, 2097
serviceClassInformation, 2081
qaGobiApiWds.h, 2098

GetAutoconnect, 2105
GetByteTotals, 2105
GetConnectionRate, 2106
GetDataBearerTechnology, 2107
GetDefaultProfile, 2107
GetDefaultProfileLTE, 2109
GetDefaultProfileNum, 2111
GetDormancyState, 2111
GetIPAddressLTE, 2112
GetLastMobileIPError, 2112
GetMobileIP, 2113
GetMobileIPProfile, 2113
GetPacketStatistics, 2115
GetPacketStatus, 2115
GetProfileSettingIn, 2101
GetProfileSettingOut, 2102
GetSessionDuration, 2116
GetSessionState, 2116
iGetByteTotals, 2117
iGetConnectionRate, 2117
iGetPacketStatistics, 2117
iSLQSMISetIPFamilyPreference, 2117
qmiDataBearerMasks, 2105
QmiProfileInfo, 2102
QmiWDSDataBearerTechnology, 2103
QmiWDSDataBearers, 2102
RMSetTransferStatistics, 2117
SLQSAutoConnect, 2126
SLQSCreateProfile, 2127
SLQSDeleteProfile, 2127
SLQSGet3GPPConfigItem, 2128
SLQSGetByteTotals, 2128
SLQSGetConnectionRate, 2129
SLQSGetCurrDataSystemStat, 2129
SLQSGetCurrentChannelRate, 2130
SLQSGetDUNCallInfo, 2131
SLQSGetDataBearerTechnology, 2130
SLQSGetDataBearerTechnologyExt, 2131
SLQSGetPacketStatistics, 2131
SLQSGetProfile, 2132
SLQSGetProfileSettings, 2134
SLQSGetRuntimeSettings, 2134
SLQSGetSessionState, 2134
SLQSModifyProfile, 2135
SLQSResetPacketStatics, 2136
SLQSSGetDHCPv4ClientConfig, 2138
SLQSSGetLoopback, 2138
SLQSSSetDHCPv4ClientConfig, 2139
SLQSSSetLoopback, 2139
SLQSSet3GPPConfigItem, 2136
SLQSSetProfile, 2136
SLQSStartStopDataSession, 2139
SLQSWdsGoActive, 2140
SLQSWdsGoDormant, 2140
SLQSWdsSetEventReport, 2140
SLQSWdsSwiPDPRuntimeSettings, 2141
SetActiveMobileIPProfile, 2118
SetAutoconnect, 2118
SetDefaultProfile, 2118
SetDefaultProfileLTE, 2120
SetDefaultProfileLTEV2, 2121
SetDefaultProfileNum, 2123
SetMobileIP, 2123
SetMobileIPParameters, 2124
SetMobileIPProfile, 2125
SetMuxID, 2126
slqs3GPPConfigItem, 2103
WDS_IsGobiDevice, 2141
qaNasGetRFBandInfo.h, 2142
PkQmiNasGetRFBandInfo, 2142
UpkQmiNasGetRFBandInfo, 2142
qaNasPerformNetworkScan.h, 2142
FORBIDDEN_INDEX, 2143
INDEX_ZERO, 2143
PREFERRED_INDEX, 2143
PkQmiNasPerformNetworkScan, 2143
ROAMING_INDEX, 2143
UpkQmiNasPerformNetworkScan, 2143
qaQmi3GPP2BroadcastCfgInfo
qaGobiApiSms.h, 1972
qaQmi3GPPBroadcastCfgInfo
qaGobiApiSms.h, 1973
qaQmi3Gpp2TimeZone, 828
daylightSavings, 828
leapSeconds, 828
localTimeOffset, 828
qaQmiInterfaceInfo, 828
qaQmiinstanceid, 829
qaQmisvctype, 829
v4sessionId, 829
v6sessionId, 829
qaQmiServingSystemParam, 829
BasestationID, 832
BasestationLatitude, 832
BasestationLongitude, 832
CDMA_P_Rev, 832
CDMASystemInfoExt, 832
CallBarStatus, 832
CellID, 832
concSvcInfo, 832
CurrentPLMN, 832
DTMInd, 832
DataSrvCapabilities, 832
defaultRoamInd, 832
DetailedSvcInfo, 832
Gpp2TimeZone, 832
GppNetworkDSTAdjustment, 832
GppTimeZone, 832
hdrPersonality, 832
Lac, 833
NetworkID, 833
PRLInd, 833
roamIndicatorVal, 833
RoamingIndicatorList, 833
ServingSystem, 833
SystemID, 833

trackAreaCode, 833
qaQmiinstanceid
qaQmiInterfaceInfo, 829
qaQmisvctype
qaQmiInterfaceInfo, 829
qm_wds_ds_profile_extended_err_codes
qmerrno.h, 2149
qmerrno.h
eQCWWAN_ERR_API_MUTEX_TIMEOUT, 2146
eQCWWAN_ERR_BUFFER_SZ, 2145
eQCWWAN_ERR_CANCEL_OP, 2146
eQCWWAN_ERR_DRIVER, 2146
eQCWWAN_ERR_ENUM_BEGIN, 2145
eQCWWAN_ERR_ENUM_END, 2146
eQCWWAN_ERR_FILE_COPY, 2145
eQCWWAN_ERR_FILE_OPEN, 2145
eQCWWAN_ERR_GENERAL, 2145
eQCWWAN_ERR_INTERNAL, 2145
eQCWWAN_ERR_INVALID_ARG, 2145
eQCWWAN_ERR_INVALID_DEVID, 2145
eQCWWAN_ERR_INVALID_FILE, 2145
eQCWWAN_ERR_INVALID_QMI_RSP, 2145
eQCWWAN_ERR_INVALID_XID, 2146
eQCWWAN_ERR_MALFORMED_QMI_RSP, 2145
eQCWWAN_ERR_MEMORY, 2145
eQCWWAN_ERR_MULTIPLE_DEVICES, 2146
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED, 2146
eQCWWAN_ERR_NO_CANCELABLE_OP, 2146
eQCWWAN_ERR_NO_CONNECTION, 2145
eQCWWAN_ERR_NO_DEVICE, 2145
eQCWWAN_ERR_NO_SIGNAL, 2146
eQCWWAN_ERR_NONE, 2145
eQCWWAN_ERR_NULL_TLV, 2149
eQCWWAN_ERR_OFFLINE, 2146
eQCWWAN_ERR_PDU_GENERATION, 2146
eQCWWAN_ERR_QMI_ABORTED, 2146
eQCWWAN_ERR_QMI_ACCESS_DENIED, 2148
eQCWWAN_ERR_QMI_ACK_NOT_SENT, 2148
eQCWWAN_ERR_QMI_ARG_TOO_LONG, 2146
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, 2147
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, 2147
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, 2148
eQCWWAN_ERR_QMI_CALL_FAILED, 2146
eQCWWAN_ERR_QMI_CARD_BUSY_RSP, 2149
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, 2148
eQCWWAN_ERR_QMI_CAT_END, 2149
eQCWWAN_ERR_QMI_CAT_START, 2149
eQCWWAN_ERR_QMI_CAUSE_CODE, 2147
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, 2146
eQCWWAN_ERR_QMI_CONNECT, 2145
eQCWWAN_ERR_QMI_DEVICE_IN_USE, 2146
eQCWWAN_ERR_QMI_DEVICE_NOT_READY, 2147
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, 2147
eQCWWAN_ERR_QMI_DISABLED, 2147
eQCWWAN_ERR_QMI_ENCODING, 2147
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, 2149
eQCWWAN_ERR_QMI_EVENT_REG_FAILED, 2149
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, 2148
eQCWWAN_ERR_QMI_FDN_RESTRICT, 2148
eQCWWAN_ERR_QMI_FLOW_SUSPENDED, 2147
eQCWWAN_ERR_QMI_GENERAL, 2147
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, 2148
eQCWWAN_ERR_QMI_IFACE, 2145
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, 2148
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, 2147
eQCWWAN_ERR_QMI_INCORRECT_PIN, 2146
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, 2148
eQCWWAN_ERR_QMI_INJECT_TIMEOUT, 2148
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, 2147
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, 2147
eQCWWAN_ERR_QMI_INTERNAL, 2146
eQCWWAN_ERR_QMI_INVALID_ARG, 2147
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, 2146
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, 2147
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, 2149
eQCWWAN_ERR_QMI_INVALID_HANDLE, 2146
eQCWWAN_ERR_QMI_INVALID_ID, 2147
eQCWWAN_ERR_QMI_INVALID_INDEX, 2147
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, 2147
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, 2147
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, 2147
eQCWWAN_ERR_QMI_INVALID_OPERATION, 2148
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, 2146
eQCWWAN_ERR_QMI_INVALID_PINID, 2146
eQCWWAN_ERR_QMI_INVALID_PROFILE, 2146
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, 2146
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, 2147

eQCWWAN_ERR_QMI_INVALID_QMI_CMD, 2148
eQCWWAN_ERR_QMI_INVALID_QOS_ID, 2147
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, 2147
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, 2146
eQCWWAN_ERR_QMI_INVALID_TECH_PREF, 2146
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, 2149
eQCWWAN_ERR_QMI_INVALID_TRANSITION, 2147
eQCWWAN_ERR_QMI_INVALID_TX_ID, 2146
eQCWWAN_ERR_QMI_MALFORMED_MSG, 2146
eQCWWAN_ERR_QMI_MAX, 2148
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, 2147
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, 2147
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, 2147
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, 2147
eQCWWAN_ERR_QMI_MISSING_ARG, 2146
eQCWWAN_ERR_QMI_MSG_BLOCKED, 2148
eQCWWAN_ERR_QMI_NETWORK_ABORTED, 2148
eQCWWAN_ERR_QMI_NETWORK_NOT_READY, 2147
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, 2147
eQCWWAN_ERR_QMI_NO_EFFECT, 2146
eQCWWAN_ERR_QMI_NO_ENTRY, 2147
eQCWWAN_ERR_QMI_NO_FREE_PROFILE, 2146
eQCWWAN_ERR_QMI_NO_MEMORY, 2146
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, 2146
eQCWWAN_ERR_QMI_NO_RADIO, 2148
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, 2148
eQCWWAN_ERR_QMI_NO_THRESHOLDS, 2146
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, 2147
eQCWWAN_ERR_QMI_NOT_PROVISIONED, 2146
eQCWWAN_ERR_QMI_NOT_SUPPORTED, 2148
eQCWWAN_ERR_QMI_OFFSET, 2146
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, 2146
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, 2146
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, 2148
eQCWWAN_ERR_QMI_OUT_OF_CALL, 2146
eQCWWAN_ERR_QMI_PIN_BLOCKED, 2147
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, 2147
eQCWWAN_ERR_QMI_POLICY_MISMATCH, 2148
eQCWWAN_ERR_QMI_REQ, 2145
eQCWWAN_ERR_QMI_REQ_SCH, 2145
eQCWWAN_ERR_QMI_REQ_TO, 2145
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, 2147
eQCWWAN_ERR_QMI_RSP, 2145
eQCWWAN_ERR_QMI_RSP_TO, 2145
eQCWWAN_ERR_QMI_SEGMENT_ORDER, 2148
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, 2148
eQCWWAN_ERR_QMI_SESSION_INACTIVE, 2147
eQCWWAN_ERR_QMI_SESSION_INVALID, 2147
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, 2147
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, 2148
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, 2147
eQCWWAN_ERR_QMI_SMSC_ADDR, 2148
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, 2148
eQCWWAN_ERR_QMI_TPDU_TYPE, 2148
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, 2146
eQCWWAN_ERR_QMI_UNKNOWN, 2147
eQCWWAN_ERR_QMI_WIDTH, 2149
eQCWWAN_ERR_RESET, 2146
eQCWWAN_ERR_SWICM_AM_VERS_ERROR, 2148
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, 2148
eQCWWAN_ERR_SWICM_END, 2149
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, 2148
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, 2148
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, 2148
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, 2148
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, 2148
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, 2148
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, 2148
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, 2148

eQCWWAN_ERR_SWICM_SOCKET_IN_USE, 2148
eQCWWAN_ERR_SWICM_START, 2148
eQCWWAN_ERR_SWICM_TIMEOUT, 2148
eQCWWAN_ERR_SWICM_V4DWN_V6DWN, 2148
eQCWWAN_ERR_SWICM_V4DWN_V6UP, 2148
eQCWWAN_ERR_SWICM_V4UP_V6DWN, 2148
eQCWWAN_ERR_SWICM_V4UP_V6UP, 2148
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, 2149
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, 2149
eQCWWAN_ERR_SWIDCS_END, 2149
eQCWWAN_ERR_SWIDCS_FILEIO_ERR, 2149
eQCWWAN_ERR_SWIDCS_IOCTL_ERR, 2149
eQCWWAN_ERR_SWIDCS_START, 2149
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, 2149
eQCWWAN_ERR_SWIIM_END, 2149
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, 2149
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, 2149
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, 2149
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, 2149
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX, 2149
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, 2149
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE, 2149
eQCWWAN_ERR_SWIIM_FW_SECBOOT_IMAGE_NOT_SIGNED, 2149
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, 2149
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, 2149
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, 2149
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE, 2149
eQCWWAN_ERR_SWIIM_INVALID_PATH, 2149
eQCWWAN_ERR_SWIIM_OPENING_DIR, 2149
eQCWWAN_ERR_SWIIM_OPENING_FILE, 2149
eQCWWAN_ERR_SWIIM_START, 2149
eQCWWAN_ERR_SWISM_END, 2149
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, 2149
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, 2149
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, 2149
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, 2149
eQCWWAN_ERR_SWISMS_START, 2149
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, 2150
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, 2150
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, 2150
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES, 2150
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY, 2150
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET, 2150
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET, 2150
eWDS_ERR_PROFILE_REG_END, 2150
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID, 2150
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED, 2150
eWDS_ERR_PROFILE_REG_RESULT_FAIL, 2150
eWDS_ERR_PROFILE_REG_RESULT_LIST_END, 2150
qmerrno.h, 2143
eQCWWANError, 2145
qm_wds_ds_profile_extended_err_codes, 2149
QmiCbkCatEventStatusReportInd, 833
CCETlv, 833
event_Index, 833
QmiCbkLocBestAvailPosInd, 833
pAltitudeWrtEllipsoid, 838
pAltitudeWrtMeanSeaLevel, 838
pGpsTime, 838
pHeading, 838
pHeadingUnc, 838
pHorCirConf, 838
pHorEllpConf, 838
pHorReliability, 838
pHorUncCircular, 838
pHorUncEllipseOrientAzimuth, 838
pHorUncEllipseSemiMajor, 838
pHorUncEllipseSemiMinor, 838

pLatitude, 838
pLongitude, 838
pMagneticDeviation, 838
pPrecisionDilution, 838
pSensorDataUsage, 838
pSpeedHorizontal, 838
pSpeedUnc, 838
pSpeedVertical, 838
pSpeedVerticalUnc, 838
pSvUsedforFix, 838
pTechnologyMask, 838
pTimeSrc, 838
pTimeUnc, 838
pTimestampUtc, 838
pVertConfidence, 838
pVertReliability, 838
pVertUnc, 839
pXid, 839
status, 839
QmiCbkLocCradleMountInd, 839
cradleMountConfigStatus, 839
QmiCbkLocEngineStateInd, 839
engineState, 840
QmiCbkLocEventTimeSyncInd, 840
timeSyncRefCounter, 840
QmiCbkLocGetServerInd, 840
pIPv4AddrInfo, 841
pIPv6AddrInfo, 841
pURL, 841
serverStatus, 841
serverType, 841
QmiCbkLocInjectPositionInd, 842
status, 842
QmiCbkLocInjectSensorDataInd, 842
injectSensorDataStatus, 843
pAccelSamplesAccepted, 843
pAccelTempSamplesAccepted, 843
pGyroSamplesAccepted, 843
pGyroTempSamplesAccepted, 843
pOpaqueIdentifier, 843
QmiCbkLocInjectTimeInd, 844
injectTimeSyncStatus, 844
QmiCbkLocInjectUTCTimeInd, 844
status, 845
QmiCbkLocPositionReportInd, 845
pAltitudeAssumed, 849
pAltitudeWrtEllipsoid, 849
pAltitudeWrtMeanSeaLevel, 849
pFixId, 849
pGpsTime, 850
pHeading, 850
pHeadingUnc, 850
pHorConfidence, 850
pHorReliability, 850
pHorUncCircular, 850
pHorUncEllipseOrientAzimuth, 850
pHorUncEllipseSemiMajor, 850
pHorUncEllipseSemiMinor, 850
pLatitude, 850
pLeapSeconds, 850
pLongitude, 850
pMagneticDeviation, 850
pPrecisionDilution, 850
pSensorDataUsage, 850
pSpeedHorizontal, 850
pSpeedUnc, 850
pSpeedVertical, 850
pSvUsedforFix, 850
pTechnologyMask, 850
pTimeSrc, 850
pTimeUnc, 850
pTimestampUtc, 850
pVertConfidence, 850
pVertReliability, 850
pVertUnc, 850
sessionId, 850
sessionStatus, 850
QmiCbkLocSensorStreamingInd, 851
pAccelAcceptReady, 851
pAccelTempAcceptReady, 851
pGyroAcceptReady, 851
pGyroTempAcceptReady, 851
QmiCbkLocSetExtPowerConfigInd, 851
status, 852
QmiCbkLocSetServerInd, 852
serverStatus, 853
QmiCbkNasLTECphyCaInfo, 853
sPhyCaAggPcellInfo, 853
sPhyCaAggScellDlBw, 853
sPhyCaAggScellIndType, 853
sPhyCaAggScellIndex, 853
sPhyCaAggScellInfo, 853
QmiCbkSwiOmaDmEventStatusReportInd, 853
SITlv, 854
QmiCbkSwiOmaDmEventStatusReportIndExt, 854
SITlv, 854
QmiCbkTmdMitiLvlRptInd, 854
currentMitigationLvl, 854
MitigationDevInfo, 854
QmiCbkWdsStatisticsIndState, 854
RxDropConutTlv, 855
RxOkByteCountTlv, 855
RxOkConutTlv, 855
TxDropConutTlv, 855
TxOkByteCountTlv, 855
TxOkConutTlv, 855
qmiDataBearerMasks
qaGobiApiWds.h, 2105
QmiNas3GppNetworkInfo, 856
pDesription, 857
pForbidden, 857
pInUse, 857
pMCC, 857
pMNC, 857
pPreferred, 857
pRoaming, 857

QmiNasGetRFBandInfoResp, 857
pInstancesSize, 858
pLTEOperationMode, 858
pRFBandInfoElements, 858
pRfBandInfoExtFormat, 858
pRfBandwidthInfo, 858
pRfDedicatedBandInfo, 858
results, 858
QmiNasPerformNetworkScanResp, 858
pInstanceSize, 858
pInstances, 858
results, 858
QmiProfileInfo
qaGobiApiWds.h, 2102
qmiSmsMessageList, 858
messageIndex, 859
messageTag, 859
QmiSwiAvmsEventReportBinaryUpdateSessionInfo-Resp
qaGobiApiSwiAvms.h, 2000
QmiSwiAvmsEventReportConfigResp
qaGobiApiSwiAvms.h, 2001
QmiSwiAvmsEventReportConnectionRequestResp
qaGobiApiSwiAvms.h, 2001
QmiSwiAvmsEventReportDataSessionStatusResp
qaGobiApiSwiAvms.h, 2002
QmiSwiAvmsEventReportHTTPStatusResp
qaGobiApiSwiAvms.h, 2002
QmiSwiAvmsEventReportNotificationResp
qaGobiApiSwiAvms.h, 2003
QmiSwiAvmsEventReportPackageIDResp
qaGobiApiSwiAvms.h, 2004
QmiSwiAvmsEventReportRegStatusResp
qaGobiApiSwiAvms.h, 2004
QmiSwiAvmsEventReportSessionTypeResp
qaGobiApiSwiAvms.h, 2004
QmiSwiAvmsEventReportWAMSParamChangeResp
qaGobiApiSwiAvms.h, 2005
QmiSwiAvmsSetSettingsAPNInfo
qaGobiApiSwiAvms.h, 2005
QmiSwiAvmsSetSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 2005
QmiSwiAvmsSetSettingsPeriodInfo
qaGobiApiSwiAvms.h, 2005
QmiWDSDataBearerTechnology
qaGobiApiWds.h, 2103
qmiWDSDataBearerTechnology, 859
currentNetwork, 860
ratMask, 860
soMask, 860
QmiWDSDataBearers
qaGobiApiWds.h, 2102
QmiWdsIpAddressInfo, 860
pIPAddressV4, 861
pIPAddressV6, 861
pIPv6prefixlen, 861
qmiWdsRunTimeSettings, 861
pAPNName, 864
pAuthentication, 864
pDomainList, 864
pGPRSGrantedQoS, 864
pGWAddressV4, 864
pIMCNflag, 864
pIPAddressV4, 864
pIPFamilyPreference, 864
pIPV6AddrInfo, 864
pIPV6GWAddrInfo, 864
pMtu, 864
pPCSCFAddrPCO, 864
pPCSCFFQDNAddrList, 864
pPDPType, 864
pPrimaryDNSV4, 864
pPrimaryDNSV6, 864
pProfileID, 864
pProfileName, 864
pSecondaryDNSV4, 865
pSecondaryDNSV6, 865
pServerAddrList, 865
pSubnetMaskV4, 865
pTechnology, 865
pUMTSGrantedQoS, 865
pUsername, 865
qmifwinfo_s, 855
dev, 856
g, 856
s, 856
qos.h, 2150
pack_qos_BindDataPort, 2152
pack_qos_SLQSQosGetNetworkStatus, 2152
pack_qos_SLQSQosSwiReadApnExtraParams, 2153
pack_qos_SLQSQosSwiReadDataStats, 2154
pack_qos_SLQSSetQosEventCallback, 2154
unpack_qos_BindDataPort, 2155
unpack_qos_SLQSQosGetNetworkStatus, 2155
unpack_qos_SLQSQosSwiReadApnExtraParams, 2156
unpack_qos_SLQSQosSwiReadDataStats, 2156
unpack_qos_SLQSSetQosEventCallback, 2157
unpack_qos_SLQSSetQosEventCallback_ind, 2157
unpack_qos_SLQSSetQosNWStatusCallback_ind, 2158
unpack_qos_SLQSSetQosPriEventCallback_ind, 2158
unpack_qos_SLQSSetQosStatusCallback_ind, 2159
qos_BindDataPortMuxID_t, 865
MuxID, 865
qos_BindDataPortPeripheralEndPointID_t, 865
EndPointType, 866
IfaceID, 866
qos_BindDataPortSIODataPort_t, 866
SIODataPort, 866
qos_id
QosMap, 871

QosClassID, 866
gDlBitRate, 867
gUlBitRate, 867
maxDlBitRate, 867
maxUlBitRate, 867
QCI, 867
qosDeliveryOrder
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
QosEventInfo, 867
pDataBearer, 868
pPacketsCountRX, 868
pPacketsCountTX, 868
pTotalBytesRX, 868
pTotalBytesTX, 868
qosFlow
sQosStat, 988
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
QosFlowInfo, 869
pBearerID, 869
pQFlowState, 869
pRxQFilter, 869
pRxQFlowGranted, 869
pTxQFilter, 869
pTxQFlowGranted, 869
unpack_qos_SLQSSetQosEventCallback_ind_t, 1263
QosFlowInfoState, 870
id, 870
isNewFlow, 870
state, 870
QosMap, 870
dscp, 871
qos_id, 871
state, 871
Quality of Service (QOS), 56

RAN
unpack_nas_GetServingNetwork_t, 1193
RAT
_SlqsNas3GppNetworkRAT_, 89
nas_QmiNas3GppNetworkRAT, 524
RATMask
CurrNetworkInfo, 208
currNetworkInfo, 209
wds_currNetworkInfo, 1510
REGISTER_EVENT
qaGobiApiCbk.h, 1726
REGISTER_SRV
qaGobiApiCbk.h, 1726
RFBandInfoElements, 878
activeBandClass, 879
activeChannel, 879
radioInterface, 879
unpack_nas_GetRFInfo_t, 1191
RFTlv
unpack_nas_SetEventReportInd_t, 1197
RMAutoConnect
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1107
RMSetTransferStatistics
qaGobiApiWds.h, 2117
ROAMING_INDEX
qaNasPerformNetworkScan.h, 2143
RPCause
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
RPTlv
NASQmiCbkNasSystemSelPrefInd, 599
RRTlv
unpack_nas_SetEventReportInd_t, 1197
RSRPThresListLen
nas_RSRPThresh, 533
RSRPThresh, 887
RSRPThresh, 886
pRSRPThresList, 887
RSRPThresListLen, 887
RSRQThresListLen
nas_RSRQThresh, 534
RSRQThresh, 888
RSRQThresh, 887
pRSRQThresList, 888
RSRQThresListLen, 888
RSSIThresListLen
nas_RSSIThresh, 535
RSSIThresh, 889
RSSIThresh, 888
pRSSIThresList, 889
RSSIThresListLen, 889
RX_EC_IO
nas_NetworkStat1x, 505
NetworkStat1x, 614
RX_PWR
nas_NetworkStat1x, 505
nas_NetworkStatEVDO, 507
NetworkStat1x, 614
NetworkStatEVDO, 616
RXAGCList, 889
pRXAIG, 890
pRXComprSlope, 890
pRXComprThres, 890
pRXExpSlope, 890
pRXExpThres, 890
pRXStaticGain, 890
RXAVCList, 890
pAVRXAVCHeadroom, 890
pAVRXAVCSens, 890
RXBOTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 1368
RXChan
LTEInfo, 418
nas_LTEInfo, 486
rXDroppedCount

unpack_wds_GetPacketStatus_t, 1364
RXOKBytesCount
DUNCallInfoInd, 248
rXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1364
rXOkBytesCount
unpack_wds_GetPacketStatus_t, 1364
RXPCMIIRFltr, 892
pFlag, 893
pStage0Val, 893
pStage1Val, 893
pStage2Val, 893
pStage3Val, 893
pStage4Val, 893
pStageCnt, 893
rXPacketErrors
unpack_wds_GetPacketStatus_t, 1364
rXPacketOverflows
unpack_wds_GetPacketStatus_t, 1364
rXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1364
radio
unpack_nas_GetSignalStrengths_t, 1195
radio_if
nasGetTxRxInfoReq, 577
pack_nas_SLQSNasGetTxRxInfo_t, 677
radioIf
ecioListElement, 250
errorRateListElement, 254
nas_ecioListElement, 465
nas_errorRateListElement, 466
nas_rsrqInformation, 533
nas_rxSignalStrengthListElement, 538
nasNetworkReject, 587
rsrqInformation, 887
rxSignalStrengthListElement, 895
RadioIfaces
unpack_dms_GetDeviceCap_t, 1108
unpack_dms_GetDeviceCapabilities_t, 1110
unpack_nas_GetServingNetwork_t, 1193
RadioIfacesSize
unpack_dms_GetDeviceCap_t, 1108
unpack_nas_GetServingNetwork_t, 1193
radioIfacesSize
unpack_dms_GetDeviceCapabilities_t, 1110
radioInterface
nas_RFBandInfoElements, 526
nas_RfBandInfoExtFormatElements, 527
nas_RfBandwidthInfoElements, 529
nas_RfDedicatedBandInfoElements, 530
nas_RFInfoTlv, 531
nas_roamIndList, 532
nas_servSystem, 541
nas_SignalStrengthTlv, 542
nas_timeInfo, 553
RFBandInfoElements, 879
RfBandInfoExtFormatElements, 880
RfBandwidthInfoElements, 882
RfDedicatedBandInfoElements, 884
roamIndList, 885
servSystem, 908
timeInfo, 1033
radioInterfaceList
NASServingSystemInfo, 601
ServingSystemInfo, 906
radioInterfaceNo
NASServingSystemInfo, 601
ServingSystemInfo, 907
radioInterfaceSize
nas_RFInfoTlv, 531
randomizationWindow
dms_PSMRandomizationWindowTlv, 236
PSMRandomizationWindowTlv, 827
range
Port, 805
unpack_qos_Port_t, 1257
RankIndicatorInd, 871
Count1, 871
Count2, 871
rankIndicatorTlv
unpack_nas_SLQSSwiRandIndicatorCallback_-
Ind_t, 1236
rat
CSGID, 200
MNRInfo, 438
nas_CSGID, 457
nas_MNRInfo, 501
ratMask
dataBearerTechnology, 219
qmiWDSDataBearerTechnology, 860
unpack_wds_SLQSSetWdsEventCallback_ind_t,
1383
ratValue
DataBearerTech, 217
rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 659
rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 659
rawLen
fileAttributes, 259
uim_fileAttributes, 1056
rawValue
fileAttributes, 259
uim_fileAttributes, 1056
rcv4
ssdatasession_params, 992
rcv6
ssdatasession_params, 992
readResult, 871
content, 872
contentLen, 872
readTransparent
pack_uim_ReadTransparent_t, 732
UIMReadTransparentReq, 1078
readTransparentInfo, 872
length, 872

offset, 872
Reason
pack_voice_SLQSVoiceGetCallForwardingStatus-_t, 749
pack_voice_SLQSVoiceSetCallBarringPassword_t, 755
voiceGetCallFWReq, 1461
voiceSetCallBarringPwdInfo, 1483
reason
ccSUPSType, 163
pack_voice_SLQSVoiceGetCallBarring_t, 748
pack_voice_SLQSVoiceSetSUPSService_t, 759
redirNumInfo, 874
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1265
voice_ccSUPSType, 1419
voice_redirNumInfo, 1439
voiceGetCallBarringReq, 1459
voiceSetSUPSServiceReq, 1490
receiptAction
sms_routeEntry, 966
smsRouteEntry, 984
receivedBytes
omaDmFotaTlvExt, 627
reconfigReqd
_packetSrvStatus, 72
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1381
reconfiguration_required
slqsSessionStateInfo, 950
recordCount
fileAttributes, 259
uim_fileAttributes, 1056
recordSize
fileAttributes, 259
uim_fileAttributes, 1056
redirNumInfo, 872
numLen, 874
numPlan, 874
numType, 874
number, 874
PI, 874
reason, 874
SI, 874
RedirPartyNum
arrRedirPartyNum, 135
voice_arrRedirPartyNum, 1405
refData
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1145
refString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1146
ReferenceID
CatAlPhaIdentifierTlv, 160
CatEventIDDataTlv, 161
refpn
CDMAInfo, 166
nas_CDMAInfo, 447
refreshComplete
pack_uim_SLQSUIMRefreshComplete_t, 737
UIMRefreshCompleteReq, 1080
refreshEvent
unpack_uim_SLQSUIMRefreshCallback_Ind_t, 1306
RefreshMode
CatRefreshTlv, 162
RefreshStage
CatRefreshTlv, 162
regAction
nasInitNetworkReg, 582
pack_nas_SLQSInitiateNetworkRegistration_t, 672
RegForeignNID
unpack_nas_GetCDMANetworkParameters_t, 1186
RegForeignSID
unpack_nas_GetCDMANetworkParameters_t, 1186
RegHomeSID
unpack_nas_GetCDMANetworkParameters_t, 1186
regInd
_transLayerInfoNotification, 114
unpack_sms_SLQSTransLayerInfoCallback_ind_t, 1286
regPrd
AddCDMASysInfo, 118
nas_AddCDMASysInfo, 443
regRefresh
pack_uim_SLQSUIMRefreshRegister_t, 739
UIMRefreshRegisterReq, 1083
regRejectInfoValid
GSMSysInfo, 312
LTESysInfo, 433
nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
regState
nas_servSystem, 541
servSystem, 908
RegStatus
_SwiAvmsEventInd, 102
Region
fwinfo_s, 266
registerFlag
registerRefresh, 875
uim_registerRefresh, 1062
registerRefresh, 874
arrfileInfo, 875
numFiles, 875
registerFlag, 875
voteForInit, 875
RegistrationState
unpack_nas_GetServingNetwork_t, 1193
registrationState

NASServingSystemInfo, 601
ServingSystemInfo, 907
rejCause
GSMSysInfo, 312
LTESysInfo, 433
nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
rejectCause
nas_RejectReasonTlv, 525
nasNetworkReject, 587
rejectSrvDomain
GSMSysInfo, 312
LTESysInfo, 433
nas_GSMSysInfo, 474
nas_LTESysInfo, 499
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
reliabilityClass
GPRSQoS, 301
GPRSRequestedQoS, 302
LibPackGPRSRequestedQoS, 351
wds_GPRSQoS, 1519
remPartyNumber
remotePartyNum, 877
voice_remotePartyNum, 1441
remainingRetries, 875
unblockLeft, 875
verifyLeft, 875
Remote Management Service (RMS), 47
RemotePartyName
getAllCallRmtPtyName, 270
voice_getAllCallRmtPtyName, 1431
remotePartyName, 875
callerName, 876
codingScheme, 876
nameLen, 876
namePI, 876
RemotePartyNum
getAllCallRmtPtyNum, 270
voice_getAllCallRmtPtyNum, 1431
remotePartyNum, 876
numLen, 877
presentationInd, 877
remPartyNumber, 877
ReqFieldsList, 877
requestFields, 877
requestFieldsLen, 877
requestFields
ReqFieldsList, 877
requestFieldsLen
ReqFieldsList, 877
resBerRatio
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
ResCode
FirmwareUpdatStat, 262
GetAudioVolTLBConfigResp, 276
SetAudioVolTLBConfigResp, 914
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1146
reserved
omaDmFotaTlvExt, 627
resetInfoInd
pack_dms_SLQSDmsSwiIndicationRegister_t, 639
ResetInfoNotification
qaGobiApiCbk.h, 1731
ResetPDSData
qaGobiApiPds.h, 1949
ResetToFactoryDefaults
qaGobiApiDms.h, 1832
RespFieldsList, 878
responseFields, 878
responseFieldsLen, 878
responseFields
RespFieldsList, 878
responseFieldsLen
RespFieldsList, 878
results
QmiNasGetRFBandInfoResp, 858
QmiNasPerformNetworkScanResp, 858
RetryCount
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
revPolarity
lineCtrlInfo, 381
voice_lineCtrlInfo, 1433
revTunneling
unpack_wds_GetMobileIPProfile_t, 1360
ReverseMac
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
RfBandInfoExtFormat, 879
pInstancesSize, 880
pRfBandInfoExtFormatParam, 880
TlvPresent, 880
RfBandInfoExtFormatElements, 880
activeBand, 880
activeChannel, 880
radioInterface, 880
RfBandInfoList, 881
pInstanceSize, 881
pRFBandInfoParam, 881
RfBandwidthInfo, 881
pInstancesSize, 882
pRfBandwidthInfoParam, 882
TlvPresent, 882
RfBandwidthInfoElements, 882
bandwidth, 882
radioInterface, 882
RfDedicatedBandInfo, 882
pInstancesSize, 883
pRfDedicatedBandInfoParam, 883

TlvPresent, 883
RfDedicatedBandInfoElements, 883
dedicatedBand, 884
radioInterface, 884
RfSarState
pack_sar_SLQSSetRfSarState_t, 715
rfbandInfoList
NasGetRFInfoResp, 573
unpack_nas_SLQSNasGetRFInfo_t, 1224
RmTrasnferStaticsReq
pack_wds_RMSetTransferStatistics_t, 764
rmTrasnferStaticsReq, 884
bResetStatistics, 884
ulMask, 884
rms.h, 2160
pack_rms_GetSMSWake, 2160
pack_rms_SetSMSWake, 2160
unpack_rms_GetSMSWake, 2161
unpack_rms_SetSMSWake, 2161
RmtPtyNum
arrRemotePartyNum, 136
voice_arrRemotePartyNum, 1406
roamIndList, 884
numInstances, 885
radioInterface, 885
roamIndicator, 885
roamIndicator
nas_roamIndList, 532
roamIndList, 885
RoamIndicatorVal
unpack_nas_SLQSGetServingSystem_t, 1209
roamIndicatorVal
qaQmiServingSystemParam, 833
roamOrigVoiceSO
prefVoiceSO, 810
voice_prefVoiceSO, 1438
RoamPref
NASRoamPreferenceTlv, 599
roamStatus
nas_sysInfoCommon, 548
sysInfoCommon, 1018
roamStatusValid
nas_sysInfoCommon, 548
sysInfoCommon, 1018
roamTimer, 885
namID, 886
roamTimerValue, 886
roamTimerValue
roamTimer, 886
voice_roamTimer, 1442
Roaming
nas_QmiNas3GppNetworkInfo, 523
SlqsNas3GppNetworkInfo, 946
unpack_nas_GetCDMANetworkParameters_t, 1186
unpack_nas_GetServingNetwork_t, 1193
roaming
unpack_nas_SetRoamingIndicatorCallback_ind_t, 1198
roaming_ind
RoamingInfo, 885
RoamingIndicatorList
qaQmiServingSystemParam, 833
unpack_nas_SLQSGetServingSystem_t, 1209
RoamingInfo, 885
roaming_ind, 885
TlvPresent, 885
routeList
sms_setRoutesReq, 970
smsSetRoutesReq, 985
routeStorage
sms_routeEntry, 966
smsRouteEntry, 984
RptIMSCapability
pack_dms_SetIndicationRegister_t, 636
rptPosData
pack_pds_SetEventReportCallback_t, 702
rptRate
LTESigRptCfg, 427
LTESigRptConfig, 428
nas_LTESigRptCfg, 494
nas_LTESigRptConfig, 495
rscp
nas_rxInfo, 536
nas_UMTSInfo, 556
rxInfo, 891
TDSCDMASigInfoExt, 1025
tdscdmaSigInfoExt, 1026
UMTSInfo, 1092
rsrp
cellParams, 181
LTESSInfo, 430
lteSSInfo, 431
nas_cellParams, 453
nas_rxInfo, 536
nas_RxSigInfo, 537
nas_umtsLTENbrCell, 558
rxInfo, 891
RxSigInfo, 894
umtsLTENbrCell, 1093
rsrplevel
lteRsrpinformation, 424
nas_lteRsrpinformation, 491
rsrq
cellParams, 181
LTESSInfo, 430
lteSSInfo, 431
nas_cellParams, 453
nas_rsrqInformation, 533
nas_SccRxInfo, 539
nas_umtsLTENbrCell, 558
rsrqInformation, 887
SccRxInfo, 899
umtsLTENbrCell, 1093
rsrqDelta

nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
rsrqInfo
nas_SLQSSignalStrengthsInformation, 545
slqsSignalStrengthInfo, 952
SLQSSignalStrengthsInformation, 956
unpack_nas_SLQSGetSignalStrength_t, 1211
rsrqInformation, 887
radioIf, 887
rsrq, 887
rssi
CDMASSInfo, 175
cdmaSSInfo, 175
cellParams, 181
gsmCellInfo, 307
HDRSSInfo, 319
hdrSSInfo, 320
LTESSInfo, 430
lteSSInfo, 431
nas_cellParams, 453
nas_gsmCellInfo, 470
TDSCDMASigInfoExt, 1025
tdscdmaSigInfoExt, 1026
unpack_nas_GetSignalStrengths_t, 1195
rts
WdsRunTimeSettings, 1543
rx_bytes
NetStats, 611
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
rx_errors
NetStats, 611
rx_overflows
NetStats, 611
rx_packets
NetStats, 611
rx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
RxByteOKCnt
wds_RXBytesOKTlv, 1524
rxChainIndex
nas_RxSigInfo, 537
RxSigInfo, 894
RxDropConutTlv
QmiCbkWdsStatisticsIndState, 855
unpack_RMTransferStatistics_ind_t, 1276
rxInfo, 890
ecio, 891
isRadioTuned, 891
phase, 891
rscp, 891
rsrp, 891
rxPower, 891
rxLev
GERANInfo, 268
nas_GERANInfo, 468
rxOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1375
RxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 855
unpack_RMTransferStatistics_ind_t, 1276
RxOkConutTlv
QmiCbkWdsStatisticsIndState, 855
unpack_RMTransferStatistics_ind_t, 1276
rxPower
nas_rxInfo, 536
nas_RxSigInfo, 537
rxInfo, 891
RxSigInfo, 894
RxQFilter
unpack_qos_QosFlowInfo_t, 1258
RxQFlowGranted
unpack_qos_QosFlowInfo_t, 1259
RxSigInfo, 893
isRadioTuned, 894
rsrp, 894
rxChainIndex, 894
rxPower, 894
rxSignalStrength
nas_rxSignalStrengthListElement, 538
rxSignalStrengthListElement, 895
rxSignalStrengthDelta
nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
rxSignalStrengthInfo
nas_SLQSSignalStrengthsInformation, 545
SLQSSignalStrengthsInformation, 956
rxSignalStrengthList
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
rxSignalStrengthListElement, 894
radioIf, 895
rxSignalStrength, 895
rxSignalStrengthListLen
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211

s
qmifwinfo_s, 856
SMS_EVENT_ETWS
qaGobiApiCbk.h, 1766
SMS_EVENT_ETWS_PLMN
qaGobiApiCbk.h, 1767
SMS_EVENT_MESSAGE_MODE
qaGobiApiCbk.h, 1766
SMS_EVENT_MT_MESSAGE
qaGobiApiCbk.h, 1766
SMS_EVENT_SMS_ON_IMS
qaGobiApiCbk.h, 1767
SMS_EVENT_SMSC_ADDRESS
qaGobiApiCbk.h, 1767
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE
qaGobiApiCbk.h, 1766
SYS_SRV_DOMAIN_CAMPED
qaGobiApiCbk.h, 1767
SYS_SRV_DOMAIN_CS_ONLY

qaGobiApiCbk.h, 1767
SYS_SRV_DOMAIN_CS_PS
qaGobiApiCbk.h, 1767
SYS_SRV_DOMAIN_NO_SRV
qaGobiApiCbk.h, 1767
SYS_SRV_DOMAIN_PS_ONLY
qaGobiApiCbk.h, 1767
sApnExtraParams, 895
ambr_dl, 896
ambr_dl_ext, 896
ambr_dl_ext2, 896
ambr_ul, 896
ambr_ul_ext, 896
ambr_ul_ext2, 896
apnId, 896
SCI
unpack_nas_GetCDMANetworkParameters_t, 1186
SCM
unpack_nas_GetCDMANetworkParameters_t, 1186
SDPTlv
NASQmiCbkNasSystemSelPrefInd, 599
SDU_HDR_LEN
common.h, 1549
SECOND_INSTANCE
qaGobiApiCbk.h, 1726
sGetDeviceSeriesResult, 934
eDevice, 934
uResult, 934
SHORT
SwiDataTypes.h, 2186
SI
calledPartyInfo, 148
callFWExtInfo, 152
callingPartyInfo, 156
redirNumInfo, 874
voice_calledPartyInfo, 1410
voice_callFWExtInfo, 1414
voice_callingPartyInfo, 1418
voice_redirNumInfo, 1439
SIODataPort
qos_BindDataPortSIODataPort_t, 866
SITlv
QmiCbkSwiOmaDmEventStatusReportInd, 854
QmiCbkSwiOmaDmEventStatusReportIndExt, 854
sIntraSearch
LTEInfoIntrafreq, 420
nas_LTEInfoIntrafreq, 488
SLQSAVMSGetSessionInfo
qaGobiApiSwiAvms.h, 2007
SLQSAVMSGetSettings
qaGobiApiSwiAvms.h, 2007
SLQSAVMSSelection
qaGobiApiSwiAvms.h, 2007
SLQSAVMSSetEventReport
qaGobiApiSwiAvms.h, 2008
SLQSAVMSSetSettings
qaGobiApiSwiAvms.h, 2008
SLQSAVMSSetSettingsIgnoreAutoReboot
qaGobiApiSwiAvms.h, 2009
SLQSAVMSSetSettingsReq
qaGobiApiSwiAvms.h, 2005
SLQSAVMSSettings
qaGobiApiSwiAvms.h, 2006
SLQSAVMSSettingsAPNInfo
qaGobiApiSwiAvms.h, 2006
SLQSAVMSSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 2006
SLQSAVMSSettingsPeriodsInfo
qaGobiApiSwiAvms.h, 2006
SLQSAVMSStartSession
qaGobiApiSwiAvms.h, 2009
SLQSAVMSStopSession
qaGobiApiSwiAvms.h, 2010
SLQSAutoConnect
qaGobiApiWds.h, 2126
SLQSCDMADecodeMTTextMsg
qaGobiApiSms.h, 1977
SLQSCDMADecodeMTTextMsgExt
qaGobiApiSms.h, 1977
SLQSCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1978
SLQSConfigSigInfo
qaGobiApiNas.h, 1922
SLQSCreateProfile
qaGobiApiWds.h, 2127
SLQSDeleteProfile
qaGobiApiWds.h, 2127
SLQSDeleteProfileParams, 942
profileIndex, 943
profileType, 943
SLQSDeleteSMS
qaGobiApiSms.h, 1978
SLQSDmsSwiGetPCInfo
qaGobiApiDms.h, 1833
SLQSDmsSwiGetResetInfo
qaGobiApiDms.h, 1833
SLQSDmsSwiIndicationRegister
qaGobiApiDms.h, 1833
SLQSDownloadFirmwareToSlot
qaGobiApiFms.h, 1864
SLQSFWINFO_SKU_SZ
qaGobiApiFms.h, 1857
SLQSGet3GPPConfigItem
qaGobiApiWds.h, 2128
SLQSGetAGPSConfig
qaGobiApiPds.h, 1953
SLQSGetAudioPathConfig
qaGobiApiAudio.h, 1712
SLQSGetAudioProfile
qaGobiApiAudio.h, 1712
SLQSGetAudioVolTLBConfig
qaGobiApiAudio.h, 1713
SLQSGetBandCapabilities
qaGobiApiDms.h, 1833

SLQSGetBandCapability
qaGobiApiDms.h, 1833
SLQSGetBootVersionNumber
qaGobiApiFms.h, 1865
SLQSGetByteTotals
qaGobiApiWds.h, 2128
SLQSGetConnectionRate
qaGobiApiWds.h, 2129
SLQSGetCurrDataSystemStat
qaGobiApiWds.h, 2129
SLQSGetCurrentChannelRate
qaGobiApiWds.h, 2130
SLQSGetCurrentPRLInfo
qaGobiApiDms.h, 1835
SLQSGetCustFeatures
qaGobiApiDms.h, 1835
SLQSGetCustFeaturesV2
qaGobiApiDms.h, 1836
SLQSGetDUNCallInfo
qaGobiApiWds.h, 2131
SLQSGetDataBearerTechnology
qaGobiApiWds.h, 2130
SLQSGetDataBearerTechnologyExt
qaGobiApiWds.h, 2131
SLQSGetDeviceMode
qaGobiApiDcs.h, 1810
SLQSGetERIFile
qaGobiApiDms.h, 1836
SLQSGetErrorRate
qaGobiApiNas.h, 1923
SLQSGetFirmwareInfo
qaGobiApiFms.h, 1865
SLQSGetGPSStateInfo
qaGobiApiPds.h, 1954
SLQSGetIMSARegStatus
qaGobiApiImsa.h, 1880
SLQSGetIMSAServiceStatus
qaGobiApiImsa.h, 1881
SLQSGetIMSASupportedFields
qaGobiApiImsa.h, 1881
SLQSGetIMSASupportedMsg
qaGobiApiImsa.h, 1882
SLQSGetIMSSMSConfig
qaGobiApiIms.h, 1874
SLQSGetIMSUserConfig
qaGobiApiIms.h, 1875
SLQSGetIMSVoIPConfig
qaGobiApiIms.h, 1875
SLQSGetImageInfo
qaGobiApiFms.h, 1866
SLQSGetImageInfo_9x15
qaGobiApiFms.h, 1866
SLQSGetImageInfoMC77xx
qaGobiApiFms.h, 1867
SLQSGetImageInfoMC83xx
qaGobiApiFms.h, 1867
SLQSGetIndicationRegister
qaGobiApiSms.h, 1979
SLQSGetM2MAVMute
qaGobiApiSwiAudio.h, 1994
SLQSGetM2MAudioProfile
qaGobiApiSwiAudio.h, 1993
SLQSGetM2MAudioVolume
qaGobiApiSwiAudio.h, 1994
SLQSGetM2MSpkrGain
qaGobiApiSwiAudio.h, 1994
SLQSGetMessageWaiting
qaGobiApiSms.h, 1980
SLQSGetNetStatistic
qaGobiApiDcs.h, 1810
SLQSGetNetworkTime
qaGobiApiNas.h, 1923
SLQSGetOperatorNameData
qaGobiApiNas.h, 1923
SLQSGetPLMNName
qaGobiApiNas.h, 1924
SLQSGetPacketStatistics
qaGobiApiWds.h, 2131
SLQSGetPidof
qaGobiApiSwi.h, 1991
SLQSGetPowerSaveModeConfig
qaGobiApiDms.h, 1836
SLQSGetProfile
qaGobiApiWds.h, 2132
SLQSGetProfileSettings
qaGobiApiWds.h, 2134
SLQSGetRegMgrConfig
qaGobiApiIms.h, 1876
SLQSGetRfSarState
qaGobiApiSar.h, 1968
SLQSGetRuntimeSettings
qaGobiApiWds.h, 2134
SLQSGetSIPConfig
qaGobiApiIms.h, 1876
SLQSGetSMS
qaGobiApiSms.h, 1980
SLQSGetSMSList
qaGobiApiSms.h, 1982
SLQSGetSdkVersion
qaGobiApiSwi.h, 1992
SLQSGetSerialNumbers
qaGobiApiDms.h, 1837
SLQSGetServingSystem
qaGobiApiNas.h, 1924
SLQSGetSessionState
qaGobiApiWds.h, 2134
SLQSGetSignalStrength
qaGobiApiNas.h, 1925
SLQSGetSmsBroadcastConfig
qaGobiApiSms.h, 1981
SLQSGetSysSelectionPref
qaGobiApiNas.h, 1925
SLQSGetTransLayerInfo
qaGobiApiSms.h, 1983
SLQSGetTransNWRegInfo
qaGobiApiSms.h, 1983

SLQSGetUsbPortNames
qaGobiApiDcs.h, 1810
SLQSGetValidFwPriCombinations
qaGobiApiFms.h, 1868
SLQSImsConfigIndicationRegister
qaGobiApiIms.h, 1876
SLQSInitiateNetworkRegistration
qaGobiApiNas.h, 1926
SLQSIsSpkgFormatRequired
qaGobiApiFms.h, 1868
SLQSKillSDKProcess
qaGobiApiDcs.h, 1811
SLQSLOCDelAssData
qaGobiApiLoc.h, 1884
SLQSLOCEventRegister
qaGobiApiLoc.h, 1884
SLQSLOCGetBestAvailPos
qaGobiApiLoc.h, 1885
SLQSLOCGetServer
qaGobiApiLoc.h, 1885
SLQSLOCInjectPosition
qaGobiApiLoc.h, 1886
SLQSLOCInjectSensorData
qaGobiApiLoc.h, 1886
SLQSLOCInjectUTCTime
qaGobiApiLoc.h, 1887
SLQSLOCSetCradleMountConfig
qaGobiApiLoc.h, 1887
SLQSLOCSetExtPowerState
qaGobiApiLoc.h, 1887
SLQSLOCSetOpMode
qaGobiApiLoc.h, 1888
SLQSLOCSetServer
qaGobiApiLoc.h, 1888
SLQSLOCStart
qaGobiApiLoc.h, 1889
SLQSLOCStop
qaGobiApiLoc.h, 1889
SLQSModifyProfile
qaGobiApiWds.h, 2135
SLQSModifySMSStatus
qaGobiApiSms.h, 1983
SLQSNASGetLTECPHYCaInfo
qaGobiApiNas.h, 1928
SLQSNASGeteDRXParams
qaGobiApiNas.h, 1928
SLQSNASSeteDRXParams
qaGobiApiNas.h, 1932
SLQSNASSwiGetChannelLock
qaGobiApiNas.h, 1933
SLQSNASSwiSetChannelLock
qaGobiApiNas.h, 1934
SLQSNasConfigSigInfo2
qaGobiApiNas.h, 1926
SLQSNasGet3GPP2Subscription
qaGobiApiNas.h, 1927
SLQSNasGetCellLocationInfo
qaGobiApiNas.h, 1927
SLQSNasGetHDRColorCode
qaGobiApiNas.h, 1928
SLQSNasGetRFInfo
qaGobiApiNas.h, 1929
SLQSNasGetSigInfo
qaGobiApiNas.h, 1929
SLQSNasGetSysInfo
qaGobiApiNas.h, 1930
SLQSNasGetTxRxInfo
qaGobiApiNas.h, 1930
SLQSNasIndicationRegister
qaGobiApiNas.h, 1931
SLQSNasIndicationRegisterExt
qaGobiApiNas.h, 1931
SLQSNasIndicationRegisterLTECphyCa
qaGobiApiNas.h, 1932
SLQSNasNetworkRejectCallback
qaGobiApiCbk.h, 1782
SLQSNasNetworkTimeCallBack
qaGobiApiCbk.h, 1783
SLQSNasSigInfo2CallBack
qaGobiApiCbk.h, 1783
SLQSNasSigInfoCallBack
qaGobiApiCbk.h, 1783
SLQSNasSwiIndicationRegister
qaGobiApiNas.h, 1933
SLQSNasSwiModemStatus
qaGobiApiNas.h, 1933
SLQSNasSwiOTAMessageCallback
qaGobiApiCbk.h, 1784
SLQSNasSysInfoCallBack
qaGobiApiCbk.h, 1784
SLQSNasTimerCallback
qaGobiApiCbk.h, 1785
SLQSOMADMCancelSession
qaGobiApiSwiOmadms.h, 2018
SLQSOMADMGetSessionInfo
qaGobiApiSwiOmadms.h, 2018
SLQSOMADMGetSettings
qaGobiApiSwiOmadms.h, 2019
SLQSOMADMGetSettings2
qaGobiApiSwiOmadms.h, 2019
SLQSOMADMSendSelection
qaGobiApiSwiOmadms.h, 2020
SLQSOMADMSendSelection2
qaGobiApiSwiOmadms.h, 2020
SLQSOMADMSessionInfo
qaGobiApiSwiOmadms.h, 2013
SLQSOMADMSetSettings
qaGobiApiSwiOmadms.h, 2021
SLQSOMADMSetSettings2
qaGobiApiSwiOmadms.h, 2022
SLQSOMADMSetSettings3
qaGobiApiSwiOmadms.h, 2022
SLQSOMADMSettings
qaGobiApiSwiOmadms.h, 2015
SLQSOMADMSettingsReqParams
qaGobiApiSwiOmadms.h, 2016

SLQSOMADMSettingsReqParams3
qaGobiApiSwiOmadms.h, 2017
SLQSOMADMStartSession
qaGobiApiSwiOmadms.h, 2022
SLQSOMADMStartSession2
qaGobiApiSwiOmadms.h, 2023
SLQSOriginateUSSD
qaGobiApiVoice.h, 2082
SLQSPDSDeterminePosition
qaGobiApiPds.h, 1954
SLQSPDSInjectAbsoluteTimeReference
qaGobiApiPds.h, 1955
SLQSPDSInjectPositionData
qaGobiApiPds.h, 1955
SLQSPerformNetworkScan
qaGobiApiNas.h, 1934
SLQSQosGetFlowStatus
qaGobiApiQos.h, 1959
SLQSQosGetGranted
qaGobiApiQos.h, 1960
SLQSQosGetNWProf
qaGobiApiQos.h, 1961
SLQSQosGetNetworkStatus
qaGobiApiQos.h, 1960
SLQSQosModify
qaGobiApiQos.h, 1961
SLQSQosRel
qaGobiApiQos.h, 1962
SLQSQosReq
qaGobiApiQos.h, 1962
SLQSQosReset
qaGobiApiQos.h, 1963
SLQSQosResume
qaGobiApiQos.h, 1963
SLQSQosSuspend
qaGobiApiQos.h, 1964
SLQSQosSwiReadApnExtraParams
qaGobiApiQos.h, 1964
SLQSQosSwiReadDataStats
qaGobiApiQos.h, 1964
SLQSRegisterIMSAIndication
qaGobiApiImsa.h, 1882
SLQSResetPacketStatics
qaGobiApiWds.h, 2136
SLQSSGetDHCPv4ClientConfig
qaGobiApiWds.h, 2138
SLQSSGetLoopback
qaGobiApiWds.h, 2138
SLQSSSTlv
unpack_nas_SetEventReportInd_t, 1197
SLQSSSetDHCPv4ClientConfig
qaGobiApiWds.h, 2139
SLQSSSetLoopback
qaGobiApiWds.h, 2139
SLQSSendAsyncSMS
qaGobiApiSms.h, 1984
SLQSSendLongSMS
qaGobiApiSms.h, 1984
SLQSSendRawQMI
qaGobiApiSwi.h, 1992
SLQSSendSMS
qaGobiApiSms.h, 1985
SLQSSet3GPPConfigItem
qaGobiApiWds.h, 2136
SLQSSetAGPSConfig
qaGobiApiPds.h, 1956
SLQSSetAudioPathConfig
qaGobiApiAudio.h, 1713
SLQSSetAudioProfile
qaGobiApiAudio.h, 1714
SLQSSetAudioVolTLBConfig
qaGobiApiAudio.h, 1714
SLQSSetBandPreference
qaGobiApiNas.h, 1935
SLQSSetBandPreferenceCbk
qaGobiApiCbk.h, 1785
SLQSSetCrashStateCheckIgnore
qaGobiApiFms.h, 1869
SLQSSetCustFeatures
qaGobiApiDms.h, 1837
SLQSSetCustFeaturesV2
qaGobiApiDms.h, 1838
SLQSSetDHCPv4ClientLeaseStatusCallback
qaGobiApiCbk.h, 1786
SLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1786
SLQSSetDataSystemStatusCallback
qaGobiApiCbk.h, 1786
SLQSSetIMSAPdpStatusCallback
qaGobiApiCbk.h, 1787
SLQSSetIMSARatStatusCallback
qaGobiApiCbk.h, 1787
SLQSSetIMSARegStatusCallback
qaGobiApiCbk.h, 1788
SLQSSetIMSASvcStatusCallback
qaGobiApiCbk.h, 1788
SLQSSetIMSSMSConfig
qaGobiApiIms.h, 1877
SLQSSetIMSSMSConfigCallback
qaGobiApiCbk.h, 1788
SLQSSetIMSUserConfig
qaGobiApiIms.h, 1877
SLQSSetIMSUserConfigCallback
qaGobiApiCbk.h, 1789
SLQSSetIMSVoIPConfig
qaGobiApiIms.h, 1878
SLQSSetIMSVoIPConfigCallback
qaGobiApiCbk.h, 1789
SLQSSetIndicationRegister
qaGobiApiSms.h, 1986
SLQSSetLocInjectPositionCallback
qaGobiApiCbk.h, 1790
SLQSSetLocInjectUTCTimeCallback
qaGobiApiCbk.h, 1790
SLQSSetLoggingMask
qaGobiApiDcs.h, 1811

SLQSSetM2MAVMute
qaGobiApiSwiAudio.h, 1997
SLQSSetM2MAudioAVCFG
qaGobiApiSwiAudio.h, 1995
SLQSSetM2MAudioLPBK
qaGobiApiSwiAudio.h, 1995
SLQSSetM2MAudioNVDef
qaGobiApiSwiAudio.h, 1996
SLQSSetM2MAudioProfile
qaGobiApiSwiAudio.h, 1996
SLQSSetM2MAudioVolume
qaGobiApiSwiAudio.h, 1996
SLQSSetM2MSpkrGain
qaGobiApiSwiAudio.h, 1997
SLQSSetModemTempCallback
qaGobiApiCbk.h, 1790
SLQSSetPacketSrvStatusCallback
qaGobiApiCbk.h, 1791
SLQSSetPositionMethodState
qaGobiApiPds.h, 1956
SLQSSetPowerSaveModeConfig
qaGobiApiDms.h, 1838
SLQSSetProfile
qaGobiApiWds.h, 2136
SLQSSetQosEventCallback
qaGobiApiCbk.h, 1791
SLQSSetQosNWStatusCallback
qaGobiApiCbk.h, 1791
SLQSSetQosPriEventCallback
qaGobiApiCbk.h, 1792
SLQSSetQosStatusCallback
qaGobiApiCbk.h, 1792
SLQSSetRegMgrConfig
qaGobiApiIms.h, 1878
SLQSSetRegMgrConfigCallback
qaGobiApiCbk.h, 1793
SLQSSetRfSarState
qaGobiApiSar.h, 1968
SLQSSetSDKTerminatedCallback
qaGobiApiCbk.h, 1793
SLQSSetSIMBasedImageSwitching
qaGobiApiFms.h, 1869
SLQSSetSIPConfig
qaGobiApiIms.h, 1879
SLQSSetSIPConfigCallback
qaGobiApiCbk.h, 1795
SLQSSetSMSEventCallback
qaGobiApiCbk.h, 1795
SLQSSetServingSystemCallback
qaGobiApiCbk.h, 1794
SLQSSetSessionStateCallback
qaGobiApiCbk.h, 1794
SLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1794
SLQSSetSmsBroadcastActivation
qaGobiApiSms.h, 1986
SLQSSetSmsBroadcastConfig
qaGobiApiSms.h, 1987
SLQSSetSmsStorage
qaGobiApiSms.h, 1987
SLQSSetSpkgFormatRequired
qaGobiApiFms.h, 1869
SLQSSetSwiGetResetInfoCallback
qaGobiApiCbk.h, 1796
SLQSSetSwiHDRPersCallback
qaGobiApiCbk.h, 1796
SLQSSetSysSelectionPref
qaGobiApiNas.h, 1937
SLQSSetSysSelectionPrefCallBack
qaGobiApiCbk.h, 1796
SLQSSetSysSelectionPrefExt
qaGobiApiNas.h, 1937
SLQSSetTransLayerInfoCallback
qaGobiApiCbk.h, 1797
SLQSSetTransNWRegInfoCallback
qaGobiApiCbk.h, 1797
SLQSSetWdsEventCallback
qaGobiApiCbk.h, 1798
SLQSSetWdsTransferStatisticCallback
qaGobiApiCbk.h, 1798
SLQSSignalStrengthsIndReq, 953
ecioDelta, 954
ecioThresholdList, 954
ecioThresholdListLen, 954
ioDelta, 954
lteRsrpDelta, 954
lteSnrDelta, 954
rsrqDelta, 954
rxSignalStrengthDelta, 954
sinrDelta, 954
sinrThresholdList, 954
sinrThresholdListLen, 954
SLQSSignalStrengthsInformation, 955
ecioInfo, 955
errorRateInfo, 956
io, 956
lteRsrpinfo, 956
lteSnrinfo, 956
rsrqInfo, 956
rxSignalStrengthInfo, 956
sinr, 956
SLQSSmsGetMaxStorageSize
qaGobiApiSms.h, 1988
SLQSSmsGetMessageProtocol
qaGobiApiSms.h, 1988
SLQSSmsSetRoutes
qaGobiApiSms.h, 1989
SLQSStart
qaGobiApiDcs.h, 1812
SLQSStart_AVAgent
qaGobiApiDcs.h, 1812
SLQSStartSrv
qaGobiApiDcs.h, 1813
SLQSStartStopDataSession
qaGobiApiWds.h, 2139
SLQSSwiAVMSEventReportCallback

qaGobiApiCbk.h, 1799
SLQSSwiClearDyingGaspStatistics
qaGobiApiDms.h, 1839
SLQSSwiDmsGetMTU
qaGobiApiSwiDms.h, 2011
SLQSSwiDmsGetUsbNetNum
qaGobiApiSwiDms.h, 2011
SLQSSwiDmsSetMTU
qaGobiApiSwiDms.h, 2012
SLQSSwiDmsSetUsbNetNum
qaGobiApiSwiDms.h, 2012
SLQSSwiGetAllCarrierImages
qaGobiApiFms.h, 1870
SLQSSwiGetCrashAction
qaGobiApiDms.h, 1839
SLQSSwiGetCrashInfo
qaGobiApiDms.h, 1839
SLQSSwiGetDyingGaspCfg
qaGobiApiDms.h, 1840
SLQSSwiGetDyingGaspStatistics
qaGobiApiDms.h, 1840
SLQSSwiGetFSN
qaGobiApiDms.h, 1840
SLQSSwiGetFirmwareCurr
qaGobiApiFms.h, 1870
SLQSSwiGetFwUpdateStatus
qaGobiApiDms.h, 1841
SLQSSwiGetHDRPersonality
qaGobiApiNas.h, 1937
SLQSSwiGetHDRProtSubtype
qaGobiApiNas.h, 1938
SLQSSwiGetHRPDStats
qaGobiApiNas.h, 1938
SLQSSwiGetHostDevInfo
qaGobiApiDms.h, 1841
SLQSSwiGetHostDevInfoParams
qaGobiApiDms.h, 1820
SLQSSwiGetLteCQI
qaGobiApiNas.h, 1939
SLQSSwiGetLteSccRxInfo
qaGobiApiNas.h, 1939
SLQSSwiGetOSInfo
qaGobiApiDms.h, 1842
SLQSSwiGetOSInfoParams
qaGobiApiDms.h, 1821
SLQSSwiGetSMSStorage
qaGobiApiSms.h, 1989
SLQSSwiGetSerialNoExt
qaGobiApiDms.h, 1842
SLQSSwiGetSerialNoExtParams
qaGobiApiDms.h, 1821
SLQSSwiGetUSBComp
qaGobiApiDms.h, 1842
SLQSSwiNetworkDebug
qaGobiApiNas.h, 1939
SLQSSwiPSDetach
qaGobiApiNas.h, 1940
SLQSSwiSetCrashAction
qaGobiApiDms.h, 1843
SLQSSwiSetDyingGaspCfg
qaGobiApiDms.h, 1843
SLQSSwiSetHostDevInfo
qaGobiApiDms.h, 1844
SLQSSwiSetHostDevInfoParams
qaGobiApiDms.h, 1822
SLQSSwiSetOSInfo
qaGobiApiDms.h, 1844
SLQSSwiSetOSInfoParams
qaGobiApiDms.h, 1822
SLQSSwiSetUSBComp
qaGobiApiDms.h, 1844
SLQSTmdDeRegNotMitigationLvl
qaGobiApiTmd.h, 2063
SLQSTmdGetMitigationDevList
qaGobiApiTmd.h, 2064
SLQSTmdGetMitigationLvl
qaGobiApiTmd.h, 2064
SLQSTmdMitigationLvlRptCallback
qaGobiApiCbk.h, 1799
SLQSTmdRegNotMitigationLvl
qaGobiApiTmd.h, 2064
SLQSUIMAuthenticate
qaGobiApiUim.h, 2067
SLQSUIMChangePin
qaGobiApiUim.h, 2068
SLQSUIMDepersonalization
qaGobiApiUim.h, 2068
SLQSUIMEventRegister
qaGobiApiUim.h, 2069
SLQSUIMGetCardStatus
qaGobiApiUim.h, 2069
SLQSUIMGetConfiguration
qaGobiApiUim.h, 2070
SLQSUIMGetFileAttributes
qaGobiApiUim.h, 2070
SLQSUIMGetSlotsStatus
qaGobiApiUim.h, 2071
SLQSUIMGetState
qaGobiApiDms.h, 1845
SLQSUIMPowerDown
qaGobiApiUim.h, 2071
SLQSUIMPowerUp
qaGobiApiUim.h, 2072
SLQSUIMReadTransparent
qaGobiApiUim.h, 2072
SLQSUIMRefreshComplete
qaGobiApiUim.h, 2073
SLQSUIMRefreshGetLastEvent
qaGobiApiUim.h, 2073
SLQSUIMRefreshOK
qaGobiApiUim.h, 2074
SLQSUIMRefreshRegister
qaGobiApiUim.h, 2074
SLQSUIMReset
qaGobiApiUim.h, 2075
SLQSUIMSetPinProtection

qaGobiApiUim.h, 2075
SLQSUIMSetRefreshCallBack
qaGobiApiCbk.h, 1799
SLQSUIMSetStatusChangeCallBack
qaGobiApiCbk.h, 1800
SLQSUIMSwitchSlot
qaGobiApiUim.h, 2076
SLQSUIMUnblockPin
qaGobiApiUim.h, 2076
SLQSUIMVerifyPin
qaGobiApiUim.h, 2077
SLQSUpgradeFirmware9x07
qaGobiApiFms.h, 1871
SLQSUpgradeFirmware9x15
qaGobiApiFms.h, 1871
SLQSVoiceALSSelectLine
qaGobiApiVoice.h, 2083
SLQSVoiceALSSetLineSwitching
qaGobiApiVoice.h, 2083
SLQSVoiceAnswerCall
qaGobiApiVoice.h, 2084
SLQSVoiceBindSubscription
qaGobiApiVoice.h, 2084
SLQSVoiceBurstDTMF
qaGobiApiVoice.h, 2085
SLQSVoiceDialCall
qaGobiApiVoice.h, 2085
SLQSVoiceEndCall
qaGobiApiVoice.h, 2086
SLQSVoiceGetAllCallInfo
qaGobiApiVoice.h, 2086
SLQSVoiceGetCLIP
qaGobiApiVoice.h, 2089
SLQSVoiceGetCLIR
qaGobiApiVoice.h, 2090
SLQSVoiceGetCNAP
qaGobiApiVoice.h, 2090
SLQSVoiceGetCOLP
qaGobiApiVoice.h, 2091
SLQSVoiceGetCOLR
qaGobiApiVoice.h, 2091
SLQSVoiceGetCallBarring
qaGobiApiVoice.h, 2087
SLQSVoiceGetCallForwardingStatus
qaGobiApiVoice.h, 2087
SLQSVoiceGetCallInfo
qaGobiApiVoice.h, 2088
SLQSVoiceGetCallWaiting
qaGobiApiVoice.h, 2088
SLQSVoiceGetConfig
qaGobiApiVoice.h, 2092
SLQSVoiceIndicationRegister
qaGobiApiVoice.h, 2092
SLQSVoiceInfoRecCallback
qaGobiApiCbk.h, 1800
SLQSVoiceManageCalls
qaGobiApiVoice.h, 2093
SLQSVoiceOrigUSSDNoWait
qaGobiApiVoice.h, 2093
SLQSVoiceSendFlash
qaGobiApiVoice.h, 2094
SLQSVoiceSetAllCallStatusCallBack
qaGobiApiCbk.h, 1801
SLQSVoiceSetCallBarringPassword
qaGobiApiVoice.h, 2094
SLQSVoiceSetConfig
qaGobiApiVoice.h, 2095
SLQSVoiceSetDTMFEventCallBack
qaGobiApiCbk.h, 1801
SLQSVoiceSetOTASPStatusCallBack
qaGobiApiCbk.h, 1801
SLQSVoiceSetPreferredPrivacy
qaGobiApiVoice.h, 2095
SLQSVoiceSetPrivacyChangeCallBack
qaGobiApiCbk.h, 1802
SLQSVoiceSetSUPSCallBack
qaGobiApiCbk.h, 1802
SLQSVoiceSetSUPSNotificationCallback
qaGobiApiCbk.h, 1803
SLQSVoiceSetSUPSService
qaGobiApiVoice.h, 2096
SLQSVoiceStartContDTMF
qaGobiApiVoice.h, 2097
SLQSVoiceStopContDTMF
qaGobiApiVoice.h, 2097
SLQSWCDMADecodeLongTextMsg
qaGobiApiSms.h, 1990
SLQSWCDMADecodeMTTextMsg
qaGobiApiSms.h, 1990
SLQSWCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1990
SLQSWdsGoActive
qaGobiApiWds.h, 2140
SLQSWdsGoDormant
qaGobiApiWds.h, 2140
SLQSWdsSetEventReport
qaGobiApiWds.h, 2140
SLQSWdsSwiPDPRuntimeSettings
qaGobiApiWds.h, 2141
SLQSWmsAsyncRawSendCallBack
qaGobiApiCbk.h, 1803
SLQSWmsMemoryFullCallBack
qaGobiApiCbk.h, 1804
SLQSWmsMessageWaitingCallBack
qaGobiApiCbk.h, 1804
SMS_CONFIG_LEN
sms.h, 2167
SMS_MAX_SMS_ROUTES
sms.h, 2167
SMS_NUM_OF_SET
sms.h, 2167
SMSAsyncRawSend
qaGobiApiCbk.h, 1733
SMSAsyncRawSend_s, 971
alphaIDLen, 972
causeCode, 972

errorClass, 972
messageID, 972
msgDelFailureCause, 972
msgDelFailureType, 972
pAlphaID, 972
RPCause, 972
sendStatus, 972
TPCause, 972
userData, 972
SMSC_TYPE_LEN
sms.h, 2167
SMSCAddress, 972
data, 973
length, 973
SMSCAddressInfo
qaGobiApiCbk.h, 1734
sMSCAddressInfo, 973
data, 973
length, 973
sMSCAddressTlv, 973
SMSCInfo, 974
TlvPresent, 974
SMSCInfo
sMSCAddressTlv, 974
SMSCTlv
unpack_sms_SetNewSMSCallback_ind_t, 1279
SMSEtwsMessage, 974
data, 974
length, 974
notificationType, 974
SMSEtwsMessageInfo
qaGobiApiCbk.h, 1734
sMSEtwsMessageInfo, 975
data, 975
length, 975
notificationType, 975
sMSEtwsMessageTlv, 975
EtwsMessageInfo, 976
TlvPresent, 976
SMSEtwsPlmn, 976
mobileCountryCode, 976
mobileNetworkCode, 976
SMSEtwsPlmnInfo
qaGobiApiCbk.h, 1734
sMSEtwsPlmnInfo, 976
mobileCountryCode, 977
mobileNetworkCode, 977
SMSEventInfo
qaGobiApiCbk.h, 1734
SMSEventInfo_s, 977
pEtwsMessageInfo, 978
pEtwsPlmnInfo, 978
pMTMessageInfo, 978
pMessageModeInfo, 978
pSMSCAddressInfo, 978
pSMSOnIMSInfo, 978
pTransferRouteMTMessageInfo, 978
smsEventType, 978
SMSEventType
qaGobiApiCbk.h, 1766
SMSMTMessage, 981
messageIndex, 981
storageType, 981
SMSMTMessageInfo
qaGobiApiCbk.h, 1735
sMSMTMessageInfo, 981
messageIndex, 982
storageType, 982
SMSMemoryInfo, 979
messageMode, 980
storageType, 980
SMSMessageMode, 980
messageMode, 980
SMSMessageModeInfo
qaGobiApiCbk.h, 1735
sMSMessageModeInfo, 980
messageMode, 980
SMSOnIMS, 982
smsOnIMS, 982
SMSOnIMSInfo
qaGobiApiCbk.h, 1736
sMSOnIMSInfo, 982
smsOnIMS, 983
sMSOnIMSTlv, 983
IMSInfo, 983
TlvPresent, 983
SMSSupport
pack_dms_SetCustFeature_t, 634
unpack_dms_GetCustFeature_t, 1107
SMSTransferRouteMTMessage, 985
ackIndicator, 986
data, 986
format, 986
length, 986
transactionID, 986
SMSTransferRouteMTMessageInfo
qaGobiApiCbk.h, 1736
sMSTransferRouteMTMessageInfo, 986
ackIndicator, 986
data, 986
format, 986
length, 986
transactionID, 987
SNR
nas_NetworkStatEVDO, 507
NetworkStatEVDO, 616
sNonIntraSearch
LTEInfoIntrafreq, 420
nas_LTEInfoIntrafreq, 488
SO
nas_NetworkStat1x, 505
NetworkStat1x, 614
SOMask
CurrNetworkInfo, 208
currNetworkInfo, 209
wds_currNetworkInfo, 1510

SPC_SIZE
dms.h, 1561
sPhyCaAggPcellInfo
nasGetLTECphyCa, 571
QmiCbkNasLTECphyCaInfo, 853
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1197
sPhyCaAggScellDlBw
nasGetLTECphyCa, 571
QmiCbkNasLTECphyCaInfo, 853
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1197
sPhyCaAggScellIndType
nasGetLTECphyCa, 571
QmiCbkNasLTECphyCaInfo, 853
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1198
sPhyCaAggScellIndex
nasGetLTECphyCa, 571
QmiCbkNasLTECphyCaInfo, 853
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1198
sPhyCaAggScellInfo
nasGetLTECphyCa, 571
QmiCbkNasLTECphyCaInfo, 853
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1198
sQosFlowStat, 987
bearerId, 987
tx_bytes, 987
tx_bytes_drp, 987
tx_pkt, 987
tx_pkt_drp, 987
sQosStat, 987
apnId, 988
numQosFlow, 988
qosFlow, 988
total_rx_bytes, 988
total_rx_pkt, 989
total_tx_bytes, 989
total_tx_bytes_drp, 989
total_tx_pkt, 989
total_tx_pkt_drp, 989
SSInfo
unpack_nas_SetServingSystemCallback_ind_t, 1199
sSLQSSignalStrengthsInfo
nas_SLQSSignalStrengthsTlv, 545
SSTlv
unpack_nas_SetEventReportInd_t, 1197
SUPSInfo, 993
isModByCC, 993
svcType, 993
SUPSInformation
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1346
voiceSUPSInfo, 1494
SUPSType
pack_voice_SLQSVoiceManageCalls_t, 753
voiceManageCallsReq, 1477
SV, 993
id, 994
mask, 994
system, 994
SVInfo, 994
len, 995
pSV, 995
SWI Audio Service(SWIAUDIO), 58
SWI Open Mobile Alliance Service (SWIOMA), 51
SWI_API
SwiDataTypes.h, 2186
SWI_STRUCT_CarrierImage, 995
m_FwBuildId, 996
m_FwImageId, 996
m_PriBuildId, 996
m_PriImageId, 997
m_nCarrierId, 996
m_nFolderId, 996
m_nStorage, 996
SWIDMS, 62
SWIWWANCMAPI.h, 2201
samplesPerBatch
accelAcceptReady_s, 115
accelTempAcceptReady_s, 116
gyroAcceptReady_s, 313
gyroTempAcceptReady_s, 313
loc_accelAcceptReady, 382
loc_accelTempAcceptReady, 382
loc_gyroAcceptReady, 388
loc_gyroTempAcceptReady, 389
sar.h, 2162
pack_sar_SLQSGetRfSarState, 2162
pack_sar_SLQSSetRfSarState, 2162
unpack_sar_SLQSGetRfSarState, 2163
unpack_sar_SLQSSetRfSarState, 2163
satelliteInfo, 896
azimuth, 898
elevation, 898
gnssSvId, 898
healthStatus, 898
snr, 898
svInfoMask, 898
svListLen, 898
svStatus, 898
system, 898
validMask, 898
SaveSMS
qaGobiApiSms.h, 1975
sbas_almanac_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
sbas_ephemeris_sv_msk
GPSStateInfo, 305
unpack_pds_SLQSGetGPSStateInfo_t, 1253
sbas_health_sv_msk
GPSStateInfo, 305

unpack_pds_SLQSGetGPSStateInfo_t, 1253
sbas_visible_sv_msk
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
SccRxInfo, 899
numInstances, 899
rsrq, 899
sigInfo, 899
snr, 899
TlvPresent, 899
scell_idx
nas_PhyCaAggScellIndex, 512
NASPhyCaAggScellIndex, 591
PhyCaAggScellIndex, 796
scell_state
nas_PhyCaAggScellIndType, 513
nas_PhyCaAggScellInfo, 516
NASPhyCaAggScellIndType, 592
NASPhyCaAggScellInfo, 593
PhyCaAggScellIndType, 797
PhyCaAggScellInfo, 800
screeningInd
connectNumInfo, 194
voice_connectNumInfo, 1424
sduErrorRatio
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
seDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
seDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
sePCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
sePCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1387
secActivate
fileAttributes, 259
uim_fileAttributes, 1056
secActivateMask
fileAttributes, 259
uim_fileAttributes, 1056
secChA
CDMAChannel, 164
nas_CDMAChannel, 446
secChB
CDMAChannel, 165
nas_CDMAChannel, 446
secDeactivate
fileAttributes, 259
uim_fileAttributes, 1056
secDeactivateMask
fileAttributes, 259
uim_fileAttributes, 1056
secIncrease
fileAttributes, 259
uim_fileAttributes, 1056
secIncreaseMask
fileAttributes, 259
uim_fileAttributes, 1056
SecProt
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 823
secRead
fileAttributes, 259
uim_fileAttributes, 1056
secReadMask
fileAttributes, 259
uim_fileAttributes, 1056
secWrite
fileAttributes, 259
uim_fileAttributes, 1056
secWriteMask
fileAttributes, 259
uim_fileAttributes, 1056
secdns
unpack_wds_GetDefaultProfile_t, 1357
secdnsv6
unpack_wds_GetDefaultProfile_t, 1357
second
nas_timeInfo, 553
nas_UniversalTime, 559
timeInfo, 1033
UniversalTime, 1101
secondaryDNS
pack_wds_SetDefaultProfile_t, 767
SecondaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1378
SecondaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1378
secondaryHA
unpack_wds_GetMobileIPProfile_t, 1360
SectorIDLen
nas_NetworkStatEVDO, 507
NetworkStatEVDO, 616
selNetwork
nas_servSystem, 541
servSystem, 908
selected
BroadcastConfig, 144
CDMABroadcastConfig, 164
sms_BroadcastConfig, 958
sms_CDMABroadcastConfig, 959
selectedNetwork
NASServingSystemInfo, 601
ServingSystemInfo, 907
selection
pack_swioma_SLQSOMADMSendSelection_t, 728
SendSMS
qaGobiApiSms.h, 1976

sendStatus
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
sensorData, 899
flags, 901
sensorDataLen, 901
timeOfFirstSample, 901
timeOffset, 901
xAxis, 901
yAxis, 901
zAxis, 901
sensorData_t, 901
flags, 902
sensorDataLen, 902
timeOfFirstSample, 902
timeOffset, 902
xAxis, 902
yAxis, 902
zAxis, 902
sensorDataLen
sensorData, 901
sensorData_t, 902
sensorDataUsage
qaGobiApiCbk.h, 1732
sensorDataUsage_s, 903
aidingIndicatorMask, 903
usageMask, 903
serialNumbersInfo, 903
esnSize, 904
imeiSize, 904
imeiSvnSize, 904
meidSize, 904
pESNString, 904
pIMEIString, 904
pImeiSvnString, 904
pMEIDString, 904
qaGobiApiDms.h, 1820
servSystem, 907
csAttachState, 908
numRadioInterfaces, 908
psAttachState, 908
radioInterface, 908
regState, 908
selNetwork, 908
ServerAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1378
serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 654
serverStatus
QmiCbkLocGetServerInd, 841
QmiCbkLocSetServerInd, 853
unpack_loc_GetServer_Ind_t, 1168
unpack_loc_SetServer_Ind_t, 1182
serverType
LocGetServerReq, 404
LocSetServerReq, 411
pack_loc_SLQSLOCGetServer_t, 654
pack_loc_SLQSLOCSetServer_t, 663
QmiCbkLocGetServerInd, 841
unpack_loc_GetServer_Ind_t, 1168
serviceCategory
CDMABroadcastConfig, 164
sms_CDMABroadcastConfig, 959
serviceClassInformation
qaGobiApiVoice.h, 2081
serviceDomain
nas_RejectReasonTlv, 525
nasNetworkReject, 587
serviceProviderName, 905
displayCondition, 905
spn, 905
spnLength, 905
servingCellId
LTEInfoIntrafreq, 420
nas_LTEInfoIntrafreq, 488
ServingSystem
qaQmiServingSystemParam, 833
unpack_nas_SLQSGetServingSystem_t, 1209
ServingSystemInfo, 905
csAttachState, 906
hdrPersonality, 906
psAttachState, 906
radioInterfaceList, 906
radioInterfaceNo, 907
registrationState, 907
selectedNetwork, 907
sessionControl
pack_pds_StartPDSTrackingSessionExt_t, 711
sessionEndReason
_packetSrvStatus, 72
slqsSessionStateInfo, 950
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1381
SessionId
LOCStartReq, 412
pack_loc_Start_t, 665
pack_loc_Stop_t, 665
sessionId
LOCStopReq, 413
QmiCbkLocPositionReportInd, 850
ssdatasession_params, 992
unpack_loc_PositionRpt_Ind_t, 1178
sessionInfo, 908
omaDmConfig, 909
omaDmFota, 909
omaDmNotifications, 909
pack_uim_ChangePin_t, 731
pack_uim_ReadTransparent_t, 732
pack_uim_SetPinProtection_t, 733
pack_uim_SLQSUIMAuthenticate_t, 734
pack_uim_SLQSUIMGetFileAttributes_t, 736
pack_uim_SLQSUIMRefreshComplete_t, 737
pack_uim_SLQSUIMRefreshGetLastEvent_t, 738
pack_uim_SLQSUIMRefreshOK_t, 738
pack_uim_SLQSUIMRefreshRegister_t, 739

pack_uim_UnblockPin_t, 740
pack_uim_VerifyPin_t, 742
sessionInfoTlv, 909
sessionInfoTlvExt, 910
UIMAuthenticateReq, 1069
UIMChangePinReq, 1070
UIMGetFileAttributesReq, 1074
UIMReadTransparentReq, 1078
UIMRefreshCompleteReq, 1080
UIMRefreshGetLastEventReq, 1082
UIMRefreshOKReq, 1083
UIMRefreshRegisterReq, 1083
UIMSetPinProtectionReq, 1085
UIMUnblockPinReq, 1089
UIMVerifyPinReq, 1090
SessionInfoConfig
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1293
sessionInfoExt, 909
omaDmConfig, 909
omaDmFota, 909
SessionInfoFota
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1293
SessionInfoNotification
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1293
sessionInfoTlv, 909
sessionInfo, 909
sessionType, 909
TlvPresent, 909
sessionInfoTlvExt, 910
sessionInfo, 910
sessionType, 910
TlvPresent, 910
sessionInformation
qaGobiApiCbk.h, 1732
sessionInformationExt
qaGobiApiCbk.h, 1733
sessionOperation
pack_pds_StartPDSTrackingSessionExt_t, 711
sessionServerOption
pack_pds_StartPDSTrackingSessionExt_t, 711
SessionState
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
SessionStatus
unpack_pds_SetEventReport_Ind_t, 1248
sessionStatus
omaDmNotificationsTlv, 628
QmiCbkLocPositionReportInd, 850
unpack_loc_PositionRpt_Ind_t, 1179
unpack_omaDmNotificationsTlv_t, 1242
SessionType
_SwiAvmsEventInd, 102
pack_swioma_SLQSOMADMGetSessionInfo_t, 728
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
sessionType
omaDmFotaTlv, 625
pack_pds_StartPDSTrackingSessionExt_t, 711
pack_swioma_SLQSOMADMCancelSession_t, 727
pack_swioma_SLQSOMADMStartSession_t, 730
sessionInfoTlv, 909
sessionInfoTlvExt, 910
uim_refreshevent, 1061
uim_sessionInformation, 1063
uim_UIMSessionInformation, 1067
UIMRefreshEvent, 1081
UIMSessionInformation, 1084
unpack_omaDmFotaTlv_t, 1241
set_fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocSetAutoStartReq, 1003
set_fix_type
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocSetAutoStartReq, 1003
set_function
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocSetAutoStartReq, 1003
set_max_dist
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocSetAutoStartReq, 1003
set_max_time
pack_swiloc_SwiLocSetAutoStart_t, 727
SwiLocSetAutoStartReq, 1003
SetACCOLC
qaGobiApiNas.h, 1919
SetActivationStatusCallback
qaGobiApiCbk.h, 1767
SetActiveMobileIPProfile
qaGobiApiWds.h, 2118
SetAudioPathConfigReq, 910
pCodecSTGain, 911
pDTMFTXGain, 911
pECMode, 912
pNSEnable, 912
pRXAGCList, 912
pRXAVCAGCSwitch, 912
pRXAVCList, 912
pRXPCMIIRFltr, 912
pTXAGCList, 912
pTXAVCSwitch, 912
pTXGain, 912
pTXPCMIIRFltr, 912
Profile, 912
SetAudioProfileReq, 912
EarMute, 913
Generator, 913
MicMute, 913
Profile, 913
Volume, 913
SetAudioVolTLBConfigReq, 913

Generator, 914
Item, 914
Profile, 914
VolValue, 914
Volume, 914
SetAudioVolTLBConfigResp, 914
ResCode, 914
SetAutoconnect
qaGobiApiWds.h, 2118
SetCATEventCallback
qaGobiApiCbk.h, 1767
SetCDMANetworkParameters
qaGobiApiNas.h, 1920
SetCfgParamChangeCallback
qaGobiApiCbk.h, 1768
setCustomSettingV2, 915
cust_id, 915
cust_value, 915
value_length, 915
SetDataCapabilitiesCallback
qaGobiApiCbk.h, 1769
SetDefaultProfile
qaGobiApiWds.h, 2118
SetDefaultProfileLTE
qaGobiApiWds.h, 2120
SetDefaultProfileLTEV2
qaGobiApiWds.h, 2121
SetDefaultProfileNum
qaGobiApiWds.h, 2123
SetDeviceStateChangeCbk
qaGobiApiCbk.h, 1769
setDyingGaspCfg, 915
pDestSMSContent, 916
pDestSMSNum, 916
SetFwDldCompletionCbk
qaGobiApiCbk.h, 1769
SetGPSCallback
qaGobiApiCbk.h, 1770
SetIMSSMSConfigReq, 916
pPhoneCtxtURI, 916
pPhoneCtxtURILen, 916
pSMSFormat, 916
pSMSOverIPNwInd, 916
SetIMSSMSConfigResp, 916
pSettingResp, 917
SetIMSUserConfigReq, 917
pIMSDomain, 917
pIMSDomainLen, 917
SetIMSUserConfigResp, 917
pSettingResp, 918
SetIMSVoIPConfigReq, 918
pAmrMode, 920
pAmrOctetAligned, 920
pAmrWBMode, 920
pAmrWBOctetAligned, 920
pAmrWbEnable, 920
pMinSessionExpiryTimer, 920
pRTPRTCPInactTimer, 920
pRingBackTimer, 920
pRingingTimer, 920
pScrAmrEnable, 920
pScrAmrWbEnable, 920
pSessionExpiryTimer, 920
SetIMSVoIPConfigResp, 920
pSettingResp, 921
SetImagePath
qaGobiApiFms.h, 1862
SetImagesPreference
qaGobiApiFms.h, 1863
setIndicationRegReq
qaGobiApiSms.h, 1973
SetLURejectCallback
qaGobiApiCbk.h, 1775
SetLocBestAvailPosCallback
qaGobiApiCbk.h, 1770
SetLocCradleMountCallback
qaGobiApiCbk.h, 1770
SetLocDeleteAssistDataCallback
qaGobiApiCbk.h, 1771
SetLocEngineStateCallback
qaGobiApiCbk.h, 1771
SetLocEventMaskNMEACallback
qaGobiApiCbk.h, 1771
SetLocEventPositionCallback
qaGobiApiCbk.h, 1772
SetLocEventTimeSyncCallback
qaGobiApiCbk.h, 1772
SetLocGetServerCallback
qaGobiApiCbk.h, 1772
SetLocGnssSvInfoCallback
qaGobiApiCbk.h, 1772
SetLocInjectSensorDataCallback
qaGobiApiCbk.h, 1773
SetLocInjectTimeCallback
qaGobiApiCbk.h, 1773
SetLocOpModeCallback
qaGobiApiCbk.h, 1773
SetLocSensorStreamingCallback
qaGobiApiCbk.h, 1774
SetLocSetExtPowerConfigCallback
qaGobiApiCbk.h, 1774
SetLocSetServerCallback
qaGobiApiCbk.h, 1774
SetM2MAVMuteReq, 924
EarMute, 924
MicMute, 924
pCwtMute, 924
Profile, 925
SetM2MAudioAVCFGReq, 921
Device, 921
PIFACEId, 921
pPCMParams, 921
Profile, 921
SetM2MAudioLPBKReq, 922
Enable, 922
SetM2MAudioProfileReq, 922

pCwtMute, 923
pEarMute, 923
pGenerator, 923
pMicMute, 923
pVolume, 923
Profile, 923
SetM2MAudioVolumeReq, 923
Generator, 924
Level, 924
Profile, 924
SetM2MSpkrGainReq, 925
Profile, 925
Value, 925
SetMobileIP
qaGobiApiWds.h, 2123
SetMobileIPParameters
qaGobiApiWds.h, 2124
SetMobileIPProfile
qaGobiApiWds.h, 2125
SetMobileIPStatusCallback
qaGobiApiCbk.h, 1775
SetMuxID
qaGobiApiWds.h, 2126
SetNMEACallback
qaGobiApiCbk.h, 1777
SetNasLTECphyCaIndCallback
qaGobiApiCbk.h, 1775
SetNetChangeCbk
qaGobiApiCbk.h, 1776
SetNetworkPreference
qaGobiApiNas.h, 1921
SetNewSMSCallback
qaGobiApiCbk.h, 1776
SetOMADMStateCallback
qaGobiApiCbk.h, 1777
SetPDSDefaults
qaGobiApiPds.h, 1950
SetPDSState
qaGobiApiPds.h, 1951
SetPDSStateCallback
qaGobiApiCbk.h, 1778
setPINProtection, 925
pinID, 926
pinLength, 926
pinOperation, 926
pinValue, 926
SetPortAutomaticTracking
qaGobiApiPds.h, 1951
SetPower
qaGobiApiDms.h, 1832
SetPowerCallback
qaGobiApiCbk.h, 1778
SetQmapIPTable
qaGobiApiDcs.h, 1809
SetQmapIPv6Table
qaGobiApiDcs.h, 1809
SetQosMuxID
qaGobiApiQos.h, 1959
SetRFInfoCallback
qaGobiApiCbk.h, 1778
SetRMTransferStatisticsCallback
qaGobiApiCbk.h, 1779
SetRankIndicatorCallback
qaGobiApiCbk.h, 1778
SetRegMgrConfigReq, 926
pCSCFPortName, 927
pCSCFPortNameLen, 927
pIMSTestMode, 927
pPriCSCFPort, 927
SetRegMgrConfigResp, 927
pSettingResp, 927
SetRoamingIndicatorCallback
qaGobiApiCbk.h, 1779
SetSDKImagePath
qaGobiApiDcs.h, 1809
SetSIPConfigReq, 932
pSIPLocalPort, 933
pSigCompEnabled, 933
pSubscribeTimer, 933
pTimerSIPReg, 933
pTimerT1, 933
pTimerT2, 933
pTimerTf, 933
SetSIPConfigResp, 933
pSettingResp, 934
SetSLQSOMADMAlertCallback
qaGobiApiCbk.h, 1780
SetSLQSOMADMAlertCallbackExt
qaGobiApiCbk.h, 1781
SetSMSCAddress
qaGobiApiSms.h, 1976
SetSMSWake
qaGobiApiRms.h, 1966
SetServiceAutomaticTracking
qaGobiApiPds.h, 1952
SetSignalStrengthCallback
qaGobiApiCbk.h, 1779
setSignalStrengthInfo, 927
pCDMAECIODelta, 931
pCDMAECIOThresh, 931
pCDMARSSIDelta, 931
pCDMARSSIThresh, 931
pGSMRSSIDelta, 931
pGSMRSSIThresh, 931
pHDRECIODelta, 931
pHDRECIOThresh, 931
pHDRIODelta, 931
pHDRIOThresh, 931
pHDRRSSIDelta, 931
pHDRRSSIThresh, 931
pHDRSINRDelta, 931
pHDRSINRThresh, 931
pLTERSRPDelta, 931
pLTERSRPThresh, 931
pLTERSRQDelta, 931
pLTERSRQThresh, 931

pLTERSSIDelta, 932
pLTERSSIThresh, 932
pLTESNRDelta, 932
pLTESNRThresh, 932
pLTESigRptConfig, 932
pTDSCDMAECIODelta, 932
pTDSCDMAECIOThresh, 932
pTDSCDMARSCPDelta, 932
pTDSCDMARSCPThresh, 932
pTDSCDMARSSIDelta, 932
pTDSCDMARSSIThresh, 932
pTDSCDMASINRDelta, 932
pTDSCDMASINRThresh, 932
pWCDMAECIODelta, 932
pWCDMAECIOThresh, 932
pWCDMARSSIDelta, 932
pWCDMARSSIThresh, 932
SetUSSDNoWaitIndicationCallback
qaGobiApiCbk.h, 1782
SetUSSDNotificationCallback
qaGobiApiCbk.h, 1781
SetUSSDReleaseCallback
qaGobiApiCbk.h, 1782
SetUimSlotStatusChangeCallback
qaGobiApiCbk.h, 1781
SetXTRAAutomaticDownload
qaGobiApiPds.h, 1952
SetXTRANetwork
qaGobiApiPds.h, 1953
SetupEventList
CatEventListTlv, 162
Severity
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
severity
omaDmFotaTlv, 625
unpack_omaDmFotaTlv_t, 1241
Short Message Service (SMS), 44
shortName
nas_PLMNNetworkNameData, 519
nasPLMNNameResp, 596
PLMNNetworkNameData, 804
unpack_nas_SLQSGetPLMNName_t, 1204
shortNameCI
nasPLMNNameResp, 596
unpack_nas_SLQSGetPLMNName_t, 1204
shortNameEn
nasPLMNNameResp, 596
unpack_nas_SLQSGetPLMNName_t, 1205
shortNameLen
nas_PLMNNetworkNameData, 519
nasPLMNNameResp, 596
PLMNNetworkNameData, 804
unpack_nas_SLQSGetPLMNName_t, 1205
shortNameSB
nasPLMNNameResp, 596
unpack_nas_SLQSGetPLMNName_t, 1205
shortNameSpareBits
nas_PLMNNetworkNameData, 519
PLMNNetworkNameData, 805
sid
CDMAInfo, 166
nas_CDMAInfo, 447
nas_sidNid, 541
sidNid, 935
unpack_nas_GetHomeNetwork_t, 1189
SidNid
homeSIDNID, 323
nas_homeSIDNID, 480
sidNid, 934
nid, 935
sid, 935
SigInd
LibPackUMTSReqQoSSigInd, 380
UMTSReqQoSSigInd, 1099
sigInfo, 935
nas_SccRxInfo, 539
pECIOThresh, 936
pHDRSINRThresh, 936
pIOThresh, 936
pLTESNRThresh, 936
pLTESigRptCfg, 936
pRSRPThresh, 936
pRSRQThresh, 936
pRSSIThresh, 936
pTDSCDMASINRCONFThresh, 936
SccRxInfo, 899
signal
signalInfo, 937
voice_signalInfo, 1442
signalInfo, 936
alertPitch, 937
signal, 937
signalType, 937
signalStrength
nas_SignalStrengthTlv, 542
SignalStrengthDataType, 937
thresholds, 937
thresholdsSize, 937
signalStrengthReqMask
slqsSignalStrengthInfo, 952
unpack_nas_SLQSGetSignalStrength_t, 1211
signalType
signalInfo, 937
voice_signalInfo, 1442
SimCapability
unpack_dms_GetDeviceCap_t, 1108
simCapability
unpack_dms_GetDeviceCapabilities_t, 1110
sinr
HDRSSInfo, 319
hdrSSInfo, 320
nas_SLQSSignalStrengthsInformation, 545
slqsSignalStrengthInfo, 953
SLQSSignalStrengthsInformation, 956
TDSCDMASigInfoExt, 1025

tdscdmaSigInfoExt, 1026
unpack_nas_SLQSGetSignalStrength_t, 1211
sinrDelta
nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
sinrThresholdList
nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
sinrThresholdListLen
nas_SLQSSignalStrengthsIndReq, 544
SLQSSignalStrengthsIndReq, 954
sku_str
slqsfwinfo_s, 944
unpack_dms_GetFirmwareInfo_t, 1114
slot
pack_uim_SLQSUIMPowerDown_t, 736
pack_uim_SLQSUIMPowerUp_t, 737
UIMPowerDownReq, 1077
UIMPowerUpReq, 1077
slot_t, 937
bICCID, 938
bICCIDLength, 938
bLogicalSlot, 938
uPhyCardStatus, 938
uPhySlotStatus, 938
slotInf, 938
AppStatus, 939
cardState, 939
errorState, 939
numApp, 939
upinRetries, 939
upinState, 940
upukRetries, 940
SlotInfo
cardStatus, 158
uim_cardStatus, 1050
slotInfo, 940
AppStatus, 941
cardState, 941
errorState, 941
numApp, 941
upinRetries, 941
upinState, 941
upukRetries, 941
slots_t, 941
uimSlotStatus, 942
slotsstatusChange
UIMSlotStatusChangeInfo, 1087
unpack_uim_SetUimSlotStatusChangeCallback_-
ind_t, 1302
slqs3GPPConfigItem
qaGobiApiWds.h, 2103
SlqsNas3GppNetworkInfo, 944
Description, 945
Forbidden, 945
InUse, 945
MCC, 945
MNC, 945
Preferred, 945
Roaming, 946
SlqsNas3GppNetworkRAT
qaGobiApiNas.h, 1897
SlqsNasPcsDigit, 946
includes_pcs_digit, 946
MCC, 946
MNC, 946
slqsNetworkScanInfo
qaGobiApiNas.h, 1897
SlqsProfile3GPP
unpackWdsProfileParam, 1388
wds_profileInfo, 1524
WdsProfileParam, 1542
SlqsProfile3GPP2
unpackWdsProfileParam, 1388
wds_profileInfo, 1524
WdsProfileParam, 1543
slqsSessionStateInfo, 950
pQmiInterfaceInfo, 950
reconfiguration_required, 950
sessionEndReason, 950
state, 950
slqsSignalStrengthInfo, 950
ecioList, 952
ecioListLen, 952
errorRateList, 952
errorRateListLen, 952
Io, 952
ltersrp, 952
ltesnr, 952
rsrqInfo, 952
rxSignalStrengthList, 952
rxSignalStrengthListLen, 952
signalStrengthReqMask, 952
sinr, 953
slqsWdsEventInfo, 956
pDataBearer, 957
pDormancyStatus, 957
pPacketsCountRX, 957
pPacketsCountTX, 957
pQmiInterfaceInfo, 957
pTotalBytesRX, 958
pTotalBytesTX, 958
slqsautoconnect, 942
acroamsetting, 942
acsetting, 942
action, 942
slqsfwinfo_s, 943
appversion_str, 944
bootversion_str, 944
carrier_str, 944
cur_carr_name, 944
cur_carr_rev, 944
modelid_str, 944
packageid_str, 944
priversion_str, 944
sku_str, 944

slqssendasyncsmsparams_s, 946
messageFormat, 948
messageSize, 948
pFollowOnDC, 948
pForceOnDC, 948
pLinktimer, 948
pMessage, 948
pRetryMessage, 948
pRetryMessageId, 948
pServiceOption, 948
pSmsOnIms, 948
pUserData, 948
slqssendsmsparams_s, 948
messageFailureCode, 949
messageFormat, 949
messageID, 949
messageSize, 949
pLinktimer, 949
pMessage, 949
pSmsOnIms, 949
sms.h
LITEQMI_QMI_CBK_PARAM_NOCHANGE, 2167
LITEQMI_QMI_CBK_PARAM_RESET, 2167
LITEQMI_QMI_CBK_PARAM_SET, 2167
sms.h, 2163
eqmiCbkSetStatus, 2167
MAX_MSE_TWS_MSG, 2167
MAX_SMS_LIST_SIZE, 2167
pack_sms_GetSMSCAddress, 2167
pack_sms_SLQSDeleteSMS, 2169
pack_sms_SLQSGetIndicationRegister, 2169
pack_sms_SLQSGetMessageWaiting, 2170
pack_sms_SLQSGetSMS, 2170
pack_sms_SLQSGetSMSList, 2171
pack_sms_SLQSGetSmsBroadcastConfig, 2170
pack_sms_SLQSGetTransLayerInfo, 2171
pack_sms_SLQSGetTransNWRegInfo, 2172
pack_sms_SLQSModifySMSStatus, 2172
pack_sms_SLQSSendAsyncSMS, 2172
pack_sms_SLQSSetIndicationRegister, 2173
pack_sms_SLQSSetSmsBroadcastActivation, 2173
pack_sms_SLQSSetSmsBroadcastConfig, 2173
pack_sms_SLQSSetSmsStorage, 2174
pack_sms_SLQSSmsGetMaxStorageSize, 2174
pack_sms_SLQSSmsGetMessageProtocol, 2174
pack_sms_SLQSSmsSetRoutes, 2175
pack_sms_SLQSSwiGetSMSStorage, 2175
pack_sms_SaveSMS, 2168
pack_sms_SendSMS, 2168
pack_sms_SetNewSMSCallback, 2168
pack_sms_SetSMSCAddress, 2169
SMS_CONFIG_LEN, 2167
SMS_MAX_SMS_ROUTES, 2167
SMS_NUM_OF_SET, 2167
SMSC_TYPE_LEN, 2167
unpack_sms_GetSMSCAddress, 2176
unpack_sms_SLQSDeleteSMS, 2178
unpack_sms_SLQSGetIndicationRegister, 2178
unpack_sms_SLQSGetMessageWaiting, 2178
unpack_sms_SLQSGetSMS, 2179
unpack_sms_SLQSGetSMSList, 2179
unpack_sms_SLQSGetSmsBroadcastConfig, 2179
unpack_sms_SLQSGetTransLayerInfo, 2180
unpack_sms_SLQSGetTransNWRegInfo, 2180
unpack_sms_SLQSModifySMSStatus, 2180
unpack_sms_SLQSNWRegInfoCallback_ind, 2181
unpack_sms_SLQSSendAsyncSMS, 2181
unpack_sms_SLQSSetIndicationRegister, 2181
unpack_sms_SLQSSetSmsBroadcastActivation, 2182
unpack_sms_SLQSSetSmsBroadcastConfig, 2182
unpack_sms_SLQSSetSmsStorage, 2182
unpack_sms_SLQSSmsGetMaxStorageSize, 2183
unpack_sms_SLQSSmsGetMessageProtocol, 2183
unpack_sms_SLQSSmsSetRoutes, 2183
unpack_sms_SLQSSwiGetSMSStorage, 2184
unpack_sms_SLQSTransLayerInfoCallback_ind, 2184
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind, 2184
unpack_sms_SLQSWmsMemoryFullCallBack_ind, 2185
unpack_sms_SLQSWmsMessageWaitingCall-Back_ind, 2185
unpack_sms_SaveSMS, 2176
unpack_sms_SendSMS, 2176
unpack_sms_SetNewSMSCallback, 2177
unpack_sms_SetNewSMSCallback_ind, 2177
unpack_sms_SetSMSCAddress, 2177
sms_BroadcastConfig, 958
fromServiceId, 958
selected, 958
toServiceId, 958
sms_CDMABroadcastConfig, 958
language, 959
selected, 959
serviceCategory, 959
sms_getIndicationReg, 959
pRegCallStatInfoEvt, 960
pRegTransLayerInfoEvt, 960
pRegTransNWRegInfoEvt, 960
sms_getMsgWaitingInfo, 960
msgWaitInfo, 960
numInstances, 960
sms_getTransLayerInfo, 960
pRegInd, 961
pTransLayerInfo, 961
sms_getTransNWRegInfo, 961
pRegStatus, 962
sms_maxStorageSizeReq, 962
pMessageMode, 962
storageType, 962
sms_maxStorageSizeResp, 962

freeSlots, 963
maxStorageSize, 963
sms_messageWaitingInfoContent, 963
activeInd, 963
msgCount, 963
msgType, 963
sms_msgProtocolResp, 964
msgProtocol, 964
sms_qaQmi3GPP2BroadcastCfgInfo, 964
activated_ind, 965
CDMABroadcastConfig, 965
num_instances, 965
sms_qaQmi3GPPBroadcastCfgInfo, 965
activated_ind, 965
broadcastConfig, 965
num_instances, 965
sms_routeEntry, 965
messageClass, 966
messageType, 966
receiptAction, 966
routeStorage, 966
sms_sendAsyncsmsParams, 966
messageFormat, 968
messageSize, 968
pFollowOnDC, 968
pForceOnDC, 968
pLinktimer, 968
pMessage, 968
pRetryMessage, 968
pRetryMessageId, 968
pServiceOption, 968
pSmsOnIms, 968
pUserData, 968
sms_setIndicationReg, 969
pRegCallStatInfoEvt, 969
pRegTransLayerInfoEvt, 969
pRegTransNWRegInfoEvt, 969
sms_setRoutesReq, 969
numOfRoutes, 970
pTransferStatusReport, 970
routeList, 970
sms_transLayerInfo, 970
TransCap, 970
TransType, 971
smsEventType
SMSEventInfo_s, 978
smsMaxStorageSizeReq, 978
pMessageMode, 978
storageType, 979
smsMaxStorageSizeResp, 979
freeSlots, 979
maxStorageSize, 979
smsMsgprotocolResp, 980
msgProtocol, 981
smsOnIMS
SMSOnIMS, 982
sMSOnIMSInfo, 983
smsRouteEntry, 983
messageClass, 984
messageType, 984
receiptAction, 984
routeStorage, 984
smsSetRoutesReq, 984
numOfRoutes, 985
pTransferStatusReport, 985
routeList, 985
smsStorage
pack_sms_SLQSSetSmsStorage_t, 723
snr
loc_satelliteInfo, 395
LTESSInfo, 430
lteSSInfo, 431
nas_SccRxInfo, 539
satelliteInfo, 898
SccRxInfo, 899
snrlevel
lteSnrinformation, 428
nas_lteSnrinformation, 495
soMask
DataBearerTech, 217
dataBearerTechnology, 219
qmiWDSDataBearerTechnology, 860
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
Source
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
source
_getResetInfoNotification, 68
altitudeSrcInfo, 122
altSrcInfo_t, 123
dmsSwiGetResetInfo, 244
unpack_dms_GetNetworkTime_t, 1119
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1134
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1135
sourceIPMask
LibPackTFTIDParams, 377
TFTIDParams, 1031
SourceLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1295
spc
pack_dms_ResetToFactoryDefaults_t, 631
pack_nas_SetACCOLC_t, 666
pack_wds_SetMobileIPProfile_t, 771
Specific Absorption Rate (SAR), 50
spn
nas_serviceProviderName, 539
nasPLMNNameResp, 596
serviceProviderName, 905
unpack_nas_SLQSGetPLMNName_t, 1205
spnEncoding
nasPLMNNameResp, 596
unpack_nas_SLQSGetPLMNName_t, 1205
spnLength

nas_serviceProviderName, 539
nasPLMNNameResp, 597
serviceProviderName, 905
unpack_nas_SLQSGetPLMNName_t, 1205
srcPortRangeEnd
LibPackTFTIDParams, 377
TFTIDParams, 1031
srcPortRangeStart
LibPackTFTIDParams, 377
TFTIDParams, 1031
srvCapability
detailSvcInfo, 226
nas_detailSvcInfo, 462
nas_sysInfoCommon, 548
sysInfoCommon, 1018
srvCapabilityValid
nas_sysInfoCommon, 548
sysInfoCommon, 1018
srvDomain
nas_sysInfoCommon, 548
sysInfoCommon, 1018
SrvDomainPref
NASServDomainPrefTlv, 600
srvDomainValid
nas_sysInfoCommon, 548
sysInfoCommon, 1018
srvOption
arrSvcOption, 136
voice_arrSvcOption, 1407
srvStatus
detailSvcInfo, 227
GSMSrvStatusInfo, 309
nas_detailSvcInfo, 462
nas_GSMSrvStatusInfo, 471
nas_SrvStatusInfo, 546
SrvStatusInfo, 989
SrvStatusInfo, 989
isPrefDataPath, 989
srvStatus, 989
srxlev
cellParams, 181
gsmCellInfo, 307
nas_cellParams, 453
nas_gsmCellInfo, 470
nas_umtsLTENbrCell, 558
nas_wcdmaCellInfo, 560
umtsLTENbrCell, 1093
wcdmaCellInfo, 1497
ssdatasession_params, 989
action, 991
failureReason, 991
failureReasonv4, 992
failureReasonv6, 992
instanceId, 992
ipfamily, 992
pAuthentication, 992
pPassword, 992
pProfileId3GPP, 992
pProfileId3GPP2, 992
pTechnology, 992
pUsername, 992
rcv4, 992
rcv6, 992
sessionId, 992
v4sessionId, 992
v6sessionId, 992
verbFailReason, 992
verbFailReasonType, 992
stage
uim_refreshevent, 1061
UIMRefreshEvent, 1081
StartPDSTrackingSessionExt
qaGobiApiPds.h, 1957
State
nas_NetworkStat1x, 505
nas_NetworkStatEVDO, 507
NetworkStat1x, 614
NetworkStatEVDO, 616
state
LocSetCradleMountReq, 410
omaDmConfigTlv, 621
omaDmConfigTlvExt, 623
omaDmFotaTlv, 625
omaDmFotaTlvExt, 627
pack_loc_SLQSLOCSetCradleMountConfig_t, 662
QosFlowInfoState, 870
QosMap, 871
slqsSessionStateInfo, 950
unpack_dms_GetActivationState_t, 1102
unpack_dms_SLQSUIMGetState_t, 1149
unpack_omaDmConfigTlv_t, 1239
unpack_omaDmFotaTlv_t, 1241
unpack_qos_QosFlowInfoState_t, 1259
statmask
pack_wds_GetPacketStatus_t, 764
StatsMask
TransferStatInd, 1038
wds_transferStatInd, 1525
statsMask
TrStatInd, 1039
wds_TrStatInd, 1525
StatsPeriod
TransferStatInd, 1038
wds_transferStatInd, 1525
statsPeriod
TrStatInd, 1039
wds_TrStatInd, 1525
Status
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1296
status
delAssistDataStatus, 224
lteEARFCN, 414
ltePCI, 423
nas_lteEARFCN, 482
nas_ltePCI, 491

nas_wcdmaUARFCN, 566
pack_sms_SetNewSMSCallback_t, 717
QmiCbkLocBestAvailPosInd, 839
QmiCbkLocInjectPositionInd, 842
QmiCbkLocInjectUTCTimeInd, 845
QmiCbkLocSetExtPowerConfigInd, 852
unpack_loc_BestAvailPos_Ind_t, 1163
unpack_loc_DeleteAssistData_Ind_t, 1164
unpack_loc_InjectPositionCallback_Ind_t, 1169
unpack_loc_InjectUTCTimeCallback_Ind_t, 1172
unpack_loc_SetExtPowerConfig_Ind_t, 1180
unpack_loc_SetOperationMode_Ind_t, 1181
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1263
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1265
wcdmaUARFCN, 1507
statusChange
UIMStatusChangeInfo, 1087
StopPDSTrackingSession
qaGobiApiPds.h, 1958
storageIndex
FMSImageIdElement, 263
ImageIdElement, 326
storageType
pack_sms_SaveSMS_t, 716
pack_sms_SLQSDeleteSMS_t, 718
pack_sms_SLQSGetSMS_t, 719
pack_sms_SLQSGetSMSList_t, 720
pack_sms_SLQSModifySMSStatus_t, 721
sms_maxStorageSizeReq, 962
smsMaxStorageSizeReq, 979
SMSMemoryInfo, 980
SMSMTMessage, 981
sMSMTMessageInfo, 982
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1288
String
unpack_dms_GetDeviceHardwareRev_t, 1110
unpack_dms_GetDeviceMfr_t, 1111
unpack_dms_GetFSN_t, 1117
unpack_dms_UIMGetICCID_t, 1152
stringSize
unpack_dms_GetDeviceHardwareRev_t, 1110
unpack_dms_GetDeviceMfr_t, 1111
unpack_dms_UIMGetICCID_t, 1152
subAddr
calledPartySubAdd, 148
voice_calledPartySubAdd, 1410
subAddrLen
calledPartySubAdd, 148
voice_calledPartySubAdd, 1410
subAddrType
calledPartySubAdd, 148
voice_calledPartySubAdd, 1410
subnetMask
IPv4Addr, 345
unpack_qos_IPv4Addr_t, 1255
SubnetMaskV4
unpack_wds_SLQSGetRuntimeSettings_t, 1378
subsType
pack_voice_SLQSVoiceBindSubscription_t, 745
voiceBindSubscriptionInfo, 1447
SupUSBComps
unpack_dms_GetUSBComp_t, 1125
SuppOA
CUGInfo, 201
voice_CUGInfo, 1425
SuppPrefCUG
CUGInfo, 201
voice_CUGInfo, 1425
supportedMsgLen
SupportedMsgList, 993
SupportedMsgList, 992
supportedMsgLen, 993
supportedMsgs, 993
supportedMsgs
SupportedMsgList, 993
svInfoMask
loc_satelliteInfo, 395
satelliteInfo, 898
svListLen
loc_satelliteInfo, 395
satelliteInfo, 898
svStatus
loc_satelliteInfo, 395
satelliteInfo, 898
svUsedforFix
qaGobiApiCbk.h, 1736
svUsedforFix_s, 995
gnssSvUsedList, 995
gnssSvUsedList_len, 995
svc
pack_qmi_t, 712
SvcClass
callFWExtInfo, 152
callFWInfo, 153
voice_callFWExtInfo, 1414
voice_callFWInfo, 1415
SvcStatus
callFWExtInfo, 152
callFWInfo, 153
voice_callFWExtInfo, 1414
voice_callFWInfo, 1415
svcType
ccSUPSType, 163
SUPSInfo, 993
voice_ccSUPSType, 1419
voice_SUPSInfo, 1443
sw1
cardResult, 157
uim_cardResult, 1049
sw2
cardResult, 157
uim_cardResult, 1049
swVerString

pack_dms_SLQSSwiSetHostDevInfo_t, 641
unpack_dms_SLQSSwiGetHostDevInfo_t, 1147
SwiAvmsEventInd
qaGobiApiCbk.h, 1737
SwiDataTypes.h, 2185
BOOL, 2186
BYTE, 2186
CHAR, 2186
FLOAT, 2186
INT32, 2186
INT8, 2186
LPCSTR, 2186
SHORT, 2186
SWI_API, 2186
ULONG, 2187
ULONGLONG, 2187
UNUSEDPARAM, 2186
USHORT, 2187
WORD, 2187
SwiDmsGetMTUResp, 999
pEhrpdMTUSize, 999
pHrpdMTUSize, 999
pMTUSize3gpp, 999
pUsbMTUSize, 999
SwiDmsSLQSGetUsbNetNum, 999
usbNetNum, 1000
SwiLocGetAutoStart
qaGobiApiLoc.h, 1890
SwiLocGetAutoStartResp, 1000
fix_rate, 1001
fix_rate_reported, 1001
fix_type, 1001
fix_type_reported, 1001
function, 1001
function_reported, 1001
max_dist, 1001
max_dist_reported, 1001
max_time, 1001
max_time_reported, 1001
SwiLocSetAutoStart
qaGobiApiLoc.h, 1890
SwiLocSetAutoStartReq, 1001
fix_rate, 1003
fix_type, 1003
function, 1003
max_dist, 1003
max_time, 1003
set_fix_rate, 1003
set_fix_type, 1003
set_function, 1003
set_max_dist, 1003
set_max_time, 1003
swiModemStatusResp, 1003
commonInfo, 1004
pLTEInfo, 1004
SwiOTAMsg
qaGobiApiCbk.h, 1737
SwiOTAMsg_s, 1004
data, 1004
data_len, 1004
pLteNasRelInfo, 1004
pTime, 1004
type, 1005
swiPDPRuntimeSettingsReq, 1005
contextId, 1005
contextType, 1005
swiPDPRuntimeSettingsResp, 1005
pAPNName, 1007
pBearerId, 1007
pContextId, 1007
pIPv4Address, 1007
pIPv4GWAddress, 1007
pIPv6Address, 1007
pIPv6GWAddress, 1007
pPrDNSIPv4Address, 1007
pPrDNSIPv6Address, 1007
pPrPCSCFIPv4Address, 1007
pPrPCSCFIPv6Address, 1007
pSeDNSIPv4Address, 1007
pSeDNSIPv6Address, 1007
pSePCSCFIPv4Address, 1007
pSePCSCFIPv6Address, 1007
swiQosFilter, 1007
index, 1009
pEspSpi, 1009
pIPv4DstAddr, 1009
pIPv4SrcAddr, 1009
pIPv6DstAddr, 1009
pIPv6Label, 1009
pIPv6SrcAddr, 1009
pIPv6TrafCls, 1009
pId, 1009
pNxtHdrProto, 1009
pPrecedence, 1010
pTCPDstPort, 1010
pTCPSrcPort, 1010
pTos, 1010
pTranDstPort, 1010
pTranSrcPort, 1010
pUDPDstPort, 1010
pUDPSrcPort, 1010
version, 1010
swiQosFlow, 1010
index, 1012
p3GPP2Pri, 1012
p3GPPImCn, 1012
p3GPPResResidualBER, 1013
p3GPPSigInd, 1013
p3GPPTraHdlPri, 1013
pDataRate, 1013
pJitter, 1013
pLatency, 1013
pLteQci, 1013
pMaxAllowedPktSz, 1013
pMinPolicedPktSz, 1013
pPktErrRate, 1013

pProfileId3GPP2, 1013
pTokenBucket, 1013
pTrafficClass, 1013
swiQosGranted, 1013
pRxFlow, 1013
pTxFlow, 1013
swiQosIds, 1013
pIds, 1014
sz, 1014
swiQosModifyReq, 1014
id, 1014
pRxFilter, 1014
pRxFlow, 1014
pTxFilter, 1014
pTxFlow, 1015
swiQosReq, 1015
index, 1015
pRxFilter, 1015
pRxFlow, 1015
pTxFilter, 1015
pTxFlow, 1015
swiRMTrasnferStaticsReq, 1015
bResetStatistics, 1016
ulMask, 1016
swidms.h, 2187
pack_swidms_SLQSSwiDmsGetMTU, 2187
pack_swidms_SLQSSwiDmsGetUsbNetNum, 2188
pack_swidms_SLQSSwiDmsSetMTU, 2188
pack_swidms_SLQSSwiDmsSetUsbNetNum, 2188
unpack_swidms_SLQSSwiDmsGetMTU, 2189
unpack_swidms_SLQSSwiDmsGetUsbNetNum, 2189
unpack_swidms_SLQSSwiDmsSetMTU, 2189
unpack_swidms_SLQSSwiDmsSetUsbNetNum, 2190
swidms_ehrpdMTUSizeTlv, 997
ehrpdMTUSize, 997
TlvPresent, 997
swidms_hrpdMTUSizeTlv, 997
hrpdMTUSize, 997
TlvPresent, 998
swidms_mtuSize3gppTlv, 998
MTUSize3gpp, 998
TlvPresent, 998
swidms_usbMTUSizeTlv, 998
TlvPresent, 998
UsbMTUSize, 999
swiloc.h, 2190
pack_swiloc_SwiLocGetAutoStart, 2191
pack_swiloc_SwiLocSetAutoStart, 2191
unpack_swiloc_SwiLocGetAutoStart, 2191
unpack_swiloc_SwiLocSetAutoStart, 2192
swioma.h, 2192
pack_swioma_SLQSOMADMAlertCallback, 2193
pack_swioma_SLQSOMADMCancelSession, 2193
pack_swioma_SLQSOMADMGetSessionInfo, 2194
pack_swioma_SLQSOMADMGetSettings, 2195
pack_swioma_SLQSOMADMSendSelection, 2195
pack_swioma_SLQSOMADMSetSettings, 2196
pack_swioma_SLQSOMADMStartSession, 2197
unpack_swioma_SLQSOMADMAlertCallback, 2197
unpack_swioma_SLQSOMADMAlertCallback_ind, 2198
unpack_swioma_SLQSOMADMCancelSession, 2198
unpack_swioma_SLQSOMADMGetSessionInfo, 2199
unpack_swioma_SLQSOMADMGetSettings, 2199
unpack_swioma_SLQSOMADMSendSelection, 2199
unpack_swioma_SLQSOMADMSetSettings, 2200
unpack_swioma_SLQSOMADMStartSession, 2200
switchOption
pack_voice_SLQSVoiceALSSetLineSwitching_t, 744
voiceALSSetLineSwitchInfo, 1446
sysInfoCDMA
CDMASysInfo, 179
nas_CDMASysInfo, 452
sysInfoCommon, 1016
isSysForbidden, 1018
isSysForbiddenValid, 1018
roamStatus, 1018
roamStatusValid, 1018
srvCapability, 1018
srvCapabilityValid, 1018
srvDomain, 1018
srvDomainValid, 1018
sysInfoGSM
GSMSysInfo, 312
nas_GSMSysInfo, 474
sysInfoHDR
HDRSysInfo, 322
nas_HDRSysInfo, 479
sysInfoLTE
LTESysInfo, 433
nas_LTESysInfo, 499
sysInfoWCDMA
nas_WCDMASysInfo, 565
WCDMASysInfo, 1506
sysSelectPrefInfo
qaGobiApiNas.h, 1898
sysSelectPrefParams
qaGobiApiNas.h, 1902
sysSelectPrefParamsExt, 1019
pCiotAcqOrderPref, 1022
pCiotLteOpMode, 1022
pCiotLteOpModePref, 1022
pLteM1BandPref, 1022
pLteNb1BandPref, 1022

pNr5gBandPref, 1022
pSysSelectPrefParams, 1022
system
loc_satelliteInfo, 395
loc_SV, 397
satelliteInfo, 898
SV, 994
systemDiscontinuities
pack_pds_PDSInjectTimeReference_t, 700
SystemID
qaQmiServingSystemParam, 833
unpack_nas_SLQSGetServingSystem_t, 1209
systemID
CDMASysInfo, 179
nas_CDMASysInfo, 452
systemMode
CommInfo, 191
nas_CommInfo, 456
systemTime
pack_pds_PDSInjectTimeReference_t, 701
sz
swiQosIds, 1014
szAPN
_QmiSwiAvmsSetSettingsAPNInfo, 82
szAlertMsg
_QmiSwiAvmsEventReportConfigResp, 76
szDescription
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
szName
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
szPWD
_QmiSwiAvmsSetSettingsAPNInfo, 82
szUname
_QmiSwiAvmsSetSettingsAPNInfo, 82
szVersion
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75

t3396_apn
nasTimers, 610
unpack_nas_SLQSNasTimerCallback_ind_t, 1230
t3396_plmn_id
nasTimers, 610
unpack_nas_SLQSNasTimerCallback_ind_t, 1230
t3396_val
nasTimers, 610
unpack_nas_SLQSNasTimerCallback_ind_t, 1230
t_Sv, 1023
entries, 1023
len, 1023
t_gpsTime, 1022
gpsTimeOfWeekMs, 1022
gpsWeek, 1022
t_sensor, 1022
aidingIndicatorMask, 1023
usageMask, 1023
TCPDstPort
unpack_qos_swiQosFilter_t, 1268
TCPSrcPort
unpack_qos_swiQosFilter_t, 1268
TDSCDMAECIOThresh, 1023
TDSCDMAECIOThreshListLen
nas_TDSCDMAECIOThresh, 549
TDSCDMAECIOThresh, 1023
TDSCDMARSCPThresh, 1024
TDSCDMARSCPThreshListLen
nas_TDSCDMARSCPThresh, 549
TDSCDMARSCPThresh, 1024
TDSCDMARSSIThresh, 1024
TDSCDMARSSIThreshListLen
nas_TDSCDMARSSIThresh, 550
TDSCDMARSSIThresh, 1024
TDSCDMASINRCONFThresh, 1026
TDSCDMASINRThresh, 1027
TDSCDMASINRThreshListLen
nas_TDSCDMASINRThresh, 551
TDSCDMASINRThresh, 1027
TDSCDMASigInfoExt, 1025
ecio, 1025
rscp, 1025
rssi, 1025
sinr, 1025
tFNASwiLTECphyCaInfo
qaGobiApiCbk.h, 1738
tFNASwiOTAMsg
qaGobiApiCbk.h, 1738
tFNActivationStatus
qaGobiApiCbk.h, 1737
tFNAllCallStatus
qaGobiApiCbk.h, 1738
tFNAsyncRawSend
qaGobiApiCbk.h, 1738
tFNBandPreference
qaGobiApiCbk.h, 1739
tFNBestAvailPos
qaGobiApiCbk.h, 1740
tFNCATEvent
qaGobiApiCbk.h, 1741
tFNCbkUimSlotStatusChangeInd
qaGobiApiCbk.h, 1741
tFNDHCPv4ClientLeaseStatus
qaGobiApiCbk.h, 1743
tFNDTMFEvent
qaGobiApiCbk.h, 1743
tFNDUNCallInfo
qaGobiApiCbk.h, 1743
tFNDataCapabilities
qaGobiApiCbk.h, 1741
tFNDataSysStatus
qaGobiApiCbk.h, 1742
tFNDelAssistData
qaGobiApiCbk.h, 1742
tFNDeviceStateChange
qaGobiApiCbk.h, 1742
tFNEventNmea

qaGobiApiCbk.h, 1743
tFNEventPosition
qaGobiApiCbk.h, 1743
tFNFwDldCompletion
qaGobiApiCbk.h, 1744
tFNGetServer
qaGobiApiCbk.h, 1744
tFNGnssSvInfo
qaGobiApiCbk.h, 1744
tFNHDRPersonaity
qaGobiApiCbk.h, 1744
tFNImsRegMgrConfig
qaGobiApiCbk.h, 1745
tFNImsSIPConfig
qaGobiApiCbk.h, 1746
tFNImsSMSConfig
qaGobiApiCbk.h, 1746
tFNImsUserConfig
qaGobiApiCbk.h, 1746
tFNImsVoIPConfig
qaGobiApiCbk.h, 1746
tFNImsaPdpStatus
qaGobiApiCbk.h, 1745
tFNImsaRatStatus
qaGobiApiCbk.h, 1745
tFNImsaRegStatus
qaGobiApiCbk.h, 1745
tFNImsaSvcStatus
qaGobiApiCbk.h, 1745
tFNInfoRec
qaGobiApiCbk.h, 1746
tFNInjectPosition
qaGobiApiCbk.h, 1747
tFNInjectSensorData
qaGobiApiCbk.h, 1747
tFNInjectTimeStatus
qaGobiApiCbk.h, 1747
tFNInjectUTCTime
qaGobiApiCbk.h, 1747
tFNLUReject
qaGobiApiCbk.h, 1748
tFNMemoryFull
qaGobiApiCbk.h, 1750
tFNMessageWaiting
qaGobiApiCbk.h, 1750
tFNMitiLvlRpt
qaGobiApiCbk.h, 1750
tFNMobileIPStatus
qaGobiApiCbk.h, 1750
tFNModemTempInfo
qaGobiApiCbk.h, 1750
tFNNasTimer
qaGobiApiCbk.h, 1751
tFNNet
qaGobiApiCbk.h, 1751
tFNNetworkReject
qaGobiApiCbk.h, 1751
tFNNetworkTime
qaGobiApiCbk.h, 1751
tFNNewGPS
qaGobiApiCbk.h, 1752
tFNNewNMEA
qaGobiApiCbk.h, 1752
tFNNewRMTransferStatistics
qaGobiApiCbk.h, 1752
tFNNewSMS
qaGobiApiCbk.h, 1753
tFNOMADMState
qaGobiApiCbk.h, 1753
tFNOTASPStatus
qaGobiApiCbk.h, 1755
tFNOpMode
qaGobiApiCbk.h, 1754
tFNPDSState
qaGobiApiCbk.h, 1755
tFNPSMCfgParamChange
qaGobiApiCbk.h, 1756
tFNPacketSrvState
qaGobiApiCbk.h, 1755
tFNPower
qaGobiApiCbk.h, 1756
tFNPrivacyChange
qaGobiApiCbk.h, 1756
tFNQosNWStatus
qaGobiApiCbk.h, 1756
tFNQosPriEvent
qaGobiApiCbk.h, 1757
tFNQosStatus
qaGobiApiCbk.h, 1757
tFNRFInfo
qaGobiApiCbk.h, 1758
tFNRankIndicator
qaGobiApiCbk.h, 1758
tFNResetInfo
qaGobiApiCbk.h, 1758
tFNRoamingIndicator
qaGobiApiCbk.h, 1759
tFNSDKTerminated
qaGobiApiCbk.h, 1759
tFNSLQSOMADMAlert
qaGobiApiCbk.h, 1761
tFNSLQSQOSEvent
qaGobiApiCbk.h, 1761
tFNSLQSSessionState
qaGobiApiCbk.h, 1762
tFNSLQSSignalStrengths
qaGobiApiCbk.h, 1762
tFNSLQSWDSEvent
qaGobiApiCbk.h, 1762
tFNSMSEvents
qaGobiApiCbk.h, 1762
tFNSUPSInfo
qaGobiApiCbk.h, 1762
tFNSUPSNotification
qaGobiApiCbk.h, 1763
tFNSensorStreaming

qaGobiApiCbk.h, 1759
tFNServingSystem
qaGobiApiCbk.h, 1759
tFNSetCradleMount
qaGobiApiCbk.h, 1760
tFNSetEngineState
qaGobiApiCbk.h, 1760
tFNSetEventTimeSync
qaGobiApiCbk.h, 1760
tFNSetExtPowerConfig
qaGobiApiCbk.h, 1760
tFNSetServer
qaGobiApiCbk.h, 1760
tFNSigInfo
qaGobiApiCbk.h, 1761
tFNSignalStrength
qaGobiApiCbk.h, 1761
tFNSwiAvmEventReport
qaGobiApiCbk.h, 1763
tFNSysInfo
qaGobiApiCbk.h, 1763
tFNSysSelectionPref
qaGobiApiCbk.h, 1763
tFNUIMRefresh
qaGobiApiCbk.h, 1764
tFNUIMStatusChangeInfo
qaGobiApiCbk.h, 1764
tFNUSSDNoWaitIndication
qaGobiApiCbk.h, 1765
tFNUSSDNotification
qaGobiApiCbk.h, 1764
tFNUSSDRelease
qaGobiApiCbk.h, 1765
tFNtransLayerInfo
qaGobiApiCbk.h, 1764
tFNtransNWRegInfo
qaGobiApiCbk.h, 1764
TFTIDParams, 1029
destPortRangeEnd, 1031
destPortRangeStart, 1031
eValid, 1031
filterId, 1031
flowLabel, 1031
IPSECSPI, 1031
ipVersion, 1031
nextHeader, 1031
pSourceIP, 1031
sourceIPMask, 1031
srcPortRangeEnd, 1031
srcPortRangeStart, 1031
tosMask, 1031
THIRD_INSTANCE
qaGobiApiCbk.h, 1726
TIME_DATE_BUF
qaGobiApiSms.h, 1971
TIME_STAMP_BUF
qaGobiApiSms.h, 1971
TPCause
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
TRMessageTlv
unpack_sms_SetNewSMSCallback_ind_t, 1279
TX_PWR
nas_NetworkStat1x, 505
NetworkStat1x, 614
TXAGCList, 1040
pTXAIG, 1041
pTXComprSlope, 1041
pTXComprThres, 1041
pTXExpSlope, 1041
pTXExpThres, 1041
pTXStaticGain, 1041
TXBOTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 1368
TXChan
LTEInfo, 418
nas_LTEInfo, 486
tXDroppedCount
unpack_wds_GetPacketStatus_t, 1364
TXOKBytesCount
DUNCallInfoInd, 248
tXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1364
tXOkBytesCount
unpack_wds_GetPacketStatus_t, 1364
TXPCMIIRFltr, 1042
pFlag, 1043
pStage0Val, 1043
pStage1Val, 1043
pStage2Val, 1043
pStage3Val, 1043
pStage4Val, 1043
pStageCnt, 1043
tXPacketErrors
unpack_wds_GetPacketStatus_t, 1364
tXPacketOverflows
unpack_wds_GetPacketStatus_t, 1364
tXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1364
Tables, 63
tac
LTEInfoIntrafreq, 420
LTESysInfo, 433
nas_LTEInfoIntrafreq, 488
nas_LTESysInfo, 499
tacValid
LTESysInfo, 434
nas_LTESysInfo, 499
TdsBandCapability
unpack_dms_SLQSGetBandCapability_t, 1139
tdscdmaSigInfoExt, 1025
ecio, 1026
rscp, 1026
rssi, 1026

sinr, 1026
tech
NWProfile, 620
techName
_packetSrvStatus, 72
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1381
techType
DataBearerTech, 217
Technology
DeviceConfigDetail, 228
fwinfo_s, 266
nas_DeviceConfigDetail, 463
unpack_wds_SLQSGetRuntimeSettings_t, 1378
TechnologyPref
pack_nas_SetNetworkPreference_t, 668
tempData_t, 1027
temperature, 1028
temperatureDataLen, 1028
timeOfFirstSample, 1028
timeOffset, 1028
timeSource, 1028
TempStat
dms_TemperatureTlv, 237
TempTlv
unpack_dms_SwiEventReportCallBack_ind_t, 1150
Temperature
dms_TemperatureTlv, 237
temperature
CommInfo, 191
nas_CommInfo, 456
tempData_t, 1028
tempratureData, 1029
temperatureDataLen
tempData_t, 1028
tempratureData, 1029
tempratureData, 1028
temperature, 1029
temperatureDataLen, 1029
timeOfFirstSample, 1029
timeOffset, 1029
timeSource, 1029
textMsgLength
cdmaMsgEncodingParams, 174
Thermal Mitigation Device(TMD), 60
threshGsmHigh
lteGsmCellInfo, 415
nas_lteGsmCellInfo, 484
threshGsmLow
lteGsmCellInfo, 416
nas_lteGsmCellInfo, 484
threshServingLow
LTEInfoIntrafreq, 420
nas_LTEInfoIntrafreq, 488
threshXHigh
infoInterFreq, 344
nas_infoInterFreq, 481
threshXLow
infoInterFreq, 344
nas_infoInterFreq, 481
threshXhigh
lteWcdmaCellInfo, 434
nas_lteWcdmaCellInfo, 500
threshXlow
lteWcdmaCellInfo, 435
nas_lteWcdmaCellInfo, 500
thresholds
SignalStrengthDataType, 937
thresholdsSize
SignalStrengthDataType, 937
Time
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1296
wcdmaLongMsgDecodingParams, 1500
wcdmaMsgDecodingParams, 1501
time
NASTimeInfoTlv, 609
Time_uncert_ms
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
timeBase
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 706
timeInfo, 1031
day, 1033
dayLtSavingAdj, 1033
dayOfWeek, 1033
hour, 1033
minute, 1033
month, 1033
radioInterface, 1033
second, 1033
timeZone, 1033
TlvPresent, 1033
year, 1033
TimeLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1296
timeMsec
pack_loc_SLQSLOCInjectUTCTime_t, 661
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 706
timeOfFirstSample
sensorData, 901
sensorData_t, 902
tempData_t, 1028
tempratureData, 1029
timeOffset
sensorData, 901
sensorData_t, 902
tempData_t, 1028
tempratureData, 1029
timeSource
tempData_t, 1028
tempratureData, 1029

TimeStmp_gps_week
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
TimeStmp_tow_ms
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
timeSyncRefCounter
QmiCbkLocEventTimeSyncInd, 840
unpack_loc_EventTimeSyncCallback_Ind_t, 1166
timeTlv
NASQmiCbkNasSwiOTAMessageInd, 598
timeUncMsec
pack_loc_SLQSLOCInjectUTCTime_t, 661
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 707
timeZone
nas_timeInfo, 553
timeInfo, 1033
timeout
pack_pds_SetPDSDefaults_t, 703
pack_qmi_t, 712
Timers
_QmiSwiAvmsSetSettingsConnectionRetryTimers, 82
_SLQSAVMSSettingsConnectionRetryTimers, 88
timestamp
unpack_dms_GetNetworkTime_t, 1119
timestampAge
pack_loc_SLQSLOCInjectPosition_t, 659
timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 659
timingAdvance
GERANInfo, 268
nas_GERANInfo, 468
TlvPresent
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
_QmiSwiAvmsEventReportConfigResp, 76
_QmiSwiAvmsEventReportConnectionRequest-Resp, 77
_QmiSwiAvmsEventReportDataSessionStatus-Resp, 77
_QmiSwiAvmsEventReportHTTPStatusResp, 78
_QmiSwiAvmsEventReportNotificationResp, 79
_QmiSwiAvmsEventReportPackageIDResp, 80
_QmiSwiAvmsEventReportRegStatusResp, 80
_QmiSwiAvmsEventReportSessionTypeResp, 81
_QmiSwiAvmsEventReportWAMSParamChange-Resp, 81
CatCommonEventTlv, 160
CsgId, 199
DataULongLongTlv, 223
DataULongTlv, 223
dms_ActivationStatusTlv, 231
dms_OperatingModeTlv, 231
dms_PSMActiveTimerIndTlv, 232
dms_PSMActiveTimerTlv, 232
dms_PSMDurationDueToOOSTlv, 233
dms_PSMDurationThresholdTlv, 233
dms_PSMEarlyWakeupTimeTlv, 234
dms_PSMEnableStateIndTlv, 234
dms_PSMEnableStateTlv, 235
dms_PSMPeriodicUpdateTimerIndTlv, 235
dms_PSMPeriodicUpdateTimerTlv, 236
dms_PSMRandomizationWindowTlv, 236
dms_TemperatureTlv, 237
dms_VoltageTlv, 238
ehrpdMTUSizeTlv, 251
eTWSPLMNInfoTlv, 255
hrpdMTUSizeTlv, 324
IPv4Info, 346
IPv6Info, 349
loc_IPv4Info, 390
loc_IPv6Info, 391
loc_urlAddr, 398
LTEOperationMode, 423
messageModeTlv, 435
mtuSize3gppTlv, 440
nas_LTEOperationMode, 490
nas_PhyCaAggPcellInfo, 511
nas_PhyCaAggScellDlBw, 512
nas_PhyCaAggScellIndex, 512
nas_PhyCaAggScellIndType, 513
nas_PhyCaAggScellInfo, 516
nas_RankIndicatorTlv, 525
nas_RejectReasonTlv, 525
nas_RfBandInfoExtFormat, 527
nas_RfBandwidthInfo, 528
nas_RfDedicatedBandInfo, 530
nas_RFInfoTlv, 531
nas_SccRxInfo, 539
nas_SignalStrengthTlv, 542
nas_SLQSSignalStrengthsTlv, 545
nas_timeInfo, 553
NASBandPreferenceTlv, 566
NASEmergencyModeTlv, 568
NASGWAcqOrderPrefTlv, 578
NASLTEBandPreferenceTlv, 583
NASLteNasReleaseInfoTlv, 584
NASModePreferenceTlv, 584
NASNetSelPreferenceTlv, 584
NASOTAMessageTlv, 590
NASPhyCaAggPcellInfo, 591
NASPhyCaAggScellDlBw, 591
NASPhyCaAggScellIndex, 592
NASPhyCaAggScellIndType, 592
NASPhyCaAggScellInfo, 593
NASPRLPreferenceTlv, 597
NASRoamPreferenceTlv, 599
NASServDomainPrefTlv, 600
NASTimeInfoTlv, 609
newMTMessageTlv, 617
PhyCaAggPcellInfo, 795
PhyCaAggScellDlBw, 796
PhyCaAggScellIndex, 796
PhyCaAggScellIndType, 797

PhyCaAggScellInfo, 800
PlmnID, 802
PSMActiveTimerTlv, 824
PSMDurationDueToOOSTlv, 825
PSMDurationThresholdTlv, 825
PSMEarlyWakeupTimeTlv, 826
PSMEnableStateTlv, 826
PSMPeriodicUpdateTimerTlv, 827
PSMRandomizationWindowTlv, 827
RfBandInfoExtFormat, 880
RfBandwidthInfo, 882
RfDedicatedBandInfo, 883
RoamingInfo, 885
SccRxInfo, 899
sessionInfoTlv, 909
sessionInfoTlvExt, 910
sMSCAddressTlv, 974
sMSEtwsMessageTlv, 976
sMSOnIMSTlv, 983
swidms_ehrpdMTUSizeTlv, 997
swidms_hrpdMTUSizeTlv, 998
swidms_mtuSize3gppTlv, 998
swidms_usbMTUSizeTlv, 998
timeInfo, 1033
transferRouteMessageTlv, 1038
unpack_pds_SetPdsState_Ind_t, 1249
unpack_uim_SLQSUIMRefreshCallback_Ind_t, 1306
urlAddr, 1389
usbMTUSizeTlv, 1393
wds_ChannelRateTlv, 1508
wds_ConnStatusTlv, 1508
wds_DataBearTechTlv, 1511
wds_DataULongLongTlv, 1511
wds_DataULongTlv, 1512
wds_DHCPLeaseOptTlv, 1512
wds_DHCPLeaseStateTlv, 1513
wds_DHCPProfileIdTlv, 1514
wds_DormStatTlv, 1518
wds_IPv4AdTlv, 1519
wds_LastMdmCallEndRsnTlv, 1521
wds_RXBytesOKTlv, 1524
wds_TXBytesOKTlv, 1526

tlvPresent
voice_USSDNotificationNetworkInfo, 1444

Tlvresult
pack_dms_GetCustFeaturesV2_t, 631
pack_dms_SetCustFeaturesV2_t, 635
pack_dms_SetPower_t, 637
pack_dms_SetUSBComp_t, 638
pack_dms_UIMGetICCID_t, 644
pack_fms_GetImagesPreference_t, 647
pack_fms_GetStoredImages_t, 648
pack_fms_SetImagesPreference_t, 649
pack_loc_Delete_Assist_Data_t, 650
pack_loc_EventRegister_t, 652
pack_loc_SetExtPowerState_t, 652
pack_loc_SetOperationMode_t, 653
pack_loc_SLQSLOCGetBestAvailPos_t, 653
pack_loc_Start_t, 665
pack_loc_Stop_t, 665
pack_nas_SetNetworkPreference_t, 668
pack_uim_ChangePin_t, 731
pack_uim_ReadTransparent_t, 732
pack_uim_SetPinProtection_t, 733
pack_uim_UnblockPin_t, 740
pack_uim_VerifyPin_t, 742
unpack_dms_GetBandCapability_t, 1104
unpack_dms_GetCrashAction_t, 1104
unpack_dms_GetCustFeature_t, 1107
unpack_dms_GetCustFeaturesV2_t, 1107
unpack_dms_GetDeviceCap_t, 1108
unpack_dms_GetDeviceHardwareRev_t, 1110
unpack_dms_GetDeviceMfr_t, 1111
unpack_dms_GetDeviceSerialNumbers_t, 1112
unpack_dms_GetFirmwareInfo_t, 1114
unpack_dms_GetFirmwareRevision_t, 1115
unpack_dms_GetFirmwareRevisions_t, 1116
unpack_dms_GetFSN_t, 1117
unpack_dms_GetIMSI_t, 1117
unpack_dms_GetManufacturer_t, 1118
unpack_dms_GetModelID_t, 1118
unpack_dms_GetNetworkTime_t, 1119
unpack_dms_GetOfflineReason_t, 1120
unpack_dms_GetPower_t, 1121
unpack_dms_GetPRLVersion_t, 1122
unpack_dms_GetUSBComp_t, 1125
unpack_dms_GetVoiceNumber_t, 1126
unpack_dms_PSMCfgChange_ind_t, 1127
unpack_dms_ResetToFactoryDefaults_t, 1127
unpack_dms_SetActivationStatusCallback_t, 1128
unpack_dms_SetCustFeature_t, 1128
unpack_dms_SetCustFeaturesV2_t, 1129
unpack_dms_SetEventReport_ind_t, 1129
unpack_dms_SetEventReport_t, 1130
unpack_dms_SetFirmwarePreference_t, 1130
unpack_dms_SetIndicationRegister_t, 1131
unpack_dms_SetPower_t, 1131
unpack_dms_SetUSBComp_t, 1131
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1134
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1135
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1136
unpack_dms_SLQSGetERIFile_t, 1139
unpack_dms_SLQSSetPowerSaveModeConfig_t, 1141
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 1141
unpack_dms_SLQSSwiGetCrashInfo_t, 1142
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1142
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1143
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1146
unpack_dms_SLQSSwiGetHostDevInfo_t, 1147

unpack_dms_SLQSSwiGetOSInfo_t, 1147
unpack_dms_SLQSSwiGetSerialNoExt_t, 1148
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1148
unpack_dms_SLQSSwiSetHostDevInfo_t, 1149
unpack_dms_SLQSSwiSetOSInfo_t, 1149
unpack_dms_SLQSUIMGetState_t, 1149
unpack_dms_SwiSetEventReport_t, 1150
unpack_dms_UIMGetControlKeyStatus_t, 1151
unpack_dms_UIMGetICCID_t, 1152
unpack_dms_UIMGetPINStatus_t, 1153
unpack_dms_UIMSetControlKeyProtection_t, 1154
unpack_dms_UIMSetPINProtection_t, 1155
unpack_dms_UIMUnblockControlKey_t, 1155
unpack_fms_GetImagesPreference_t, 1156
unpack_fms_GetStoredImages_t, 1156
unpack_fms_SetImagesPreference_t, 1157
unpack_loc_BestAvailPos_Ind_t, 1163
unpack_loc_Delete_Assist_Data_t, 1164
unpack_loc_DeleteAssistData_Ind_t, 1164
unpack_loc_EngineState_Ind_t, 1165
unpack_loc_EventNMEA_Ind_t, 1166
unpack_loc_EventRegister_t, 1166
unpack_loc_GetServer_Ind_t, 1168
unpack_loc_GnssSvInfo_Ind_t, 1168
unpack_loc_PositionRpt_Ind_t, 1179
unpack_loc_SetExtPowerConfig_Ind_t, 1180
unpack_loc_SetExtPowerState_t, 1180
unpack_loc_SetOperationMode_Ind_t, 1181
unpack_loc_SetOperationMode_t, 1182
unpack_loc_SetServer_Ind_t, 1182
unpack_loc_SLQSLOCGetBestAvailPos_t, 1183
unpack_loc_Start_t, 1183
unpack_loc_Stop_t, 1184
unpack_nas_GetNetworkPreference_t, 1190
unpack_nas_SetNetworkPreference_t, 1198
unpack_nas_SetServingSystemCallback_ind_t, 1199
unpack_nas_SlqsGetLTECphyCAInfo_t, 1201
unpack_nas_SLQSNasGetRFInfo_t, 1224
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1229
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1230
unpack_swidms_SLQSSwiDmsSetMTU_t, 1291
unpack_uim_ChangePin_t, 1299
unpack_uim_GetCardStatus_t, 1299
unpack_uim_ReadTransparent_t, 1300
unpack_uim_SetPinProtection_t, 1301
unpack_uim_UnblockPin_t, 1307
unpack_uim_VerifyPin_t, 1308
unpack_wds_SLQSCreateProfile_t, 1366
unpack_wds_SLQSGetProfileSettings_t, 1375
unpack_wds_SLQSSetIPFamilyPreference_t, 1379
TmdDeRegNotMitigationLvlReq, 1033
mitigationDevID, 1033
mitigationDevIDLen, 1033
TmdGetMitigationDevListResp, 1033
pMitigationDevList, 1034
pMitigationDevListLen, 1034
TmdGetMitigationLvlReq, 1034
mitigationDevID, 1034
mitigationDevIDLen, 1034
TmdGetMitigationLvlResp, 1035
pCurrentmitigationLvl, 1035
pReqMitigationLvl, 1035
TmdMitigationLvlIndReq, 1035
mitigationDevID, 1036
mitigationDevIDLen, 1036
TmdRegNotMitigationLvlReq, 1036
mitigationDevID, 1036
mitigationDevIDLen, 1036
toServiceId
BroadcastConfig, 144
sms_BroadcastConfig, 958
toggleMode
lineCtrlInfo, 381
voice_lineCtrlInfo, 1433
TokenBucket
unpack_qos_swiQosFlow_t, 1272
tokenBucket, 1036
bucketSz, 1037
peakRate, 1037
tokenRate, 1037
tokenRate
tokenBucket, 1037
unpack_qos_tokenBucket_t, 1273
Tos, 1037
mask, 1037
val, 1037
tosMask
LibPackTFTIDParams, 377
TFTIDParams, 1031
total_rx_bytes
sQosStat, 988
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
total_rx_pkt
sQosStat, 989
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
total_tx_bytes
sQosStat, 989
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
total_tx_bytes_drp
sQosStat, 989
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
total_tx_pkt
sQosStat, 989
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
total_tx_pkt_drp
sQosStat, 989
unpack_qos_SLQSQosSwiReadDataStats_t, 1262
TrStatInd, 1039
statsMask, 1039
statsPeriod, 1039
TrackAreaCode
unpack_nas_SLQSGetServingSystem_t, 1209

trackAreaCode
qaQmiServingSystemParam, 833
TrackingStatus
unpack_pds_SetPdsState_Ind_t, 1249
TrafficClass
unpack_qos_swiQosFlow_t, 1272
trafficClass
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
trafficPriority
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
TranDstPort
unpack_qos_swiQosFilter_t, 1268
TranSrcPort
unpack_qos_swiQosFilter_t, 1268
TransCap
_transLayerinfo, 113
sms_transLayerInfo, 970
transLayerInfo
qaGobiApiSms.h, 1974
transLayerNotification
qaGobiApiCbk.h, 1765
transNWRegInfoNotification
qaGobiApiCbk.h, 1765
TransType
_transLayerinfo, 113
sms_transLayerInfo, 971
transactionID
SMSTransferRouteMTMessage, 986
sMSTransferRouteMTMessageInfo, 987
transferDelay
LibPackUMTSQoS, 379
UMTSMinQoS, 1096
UMTSQoS, 1098
wds_UMTSMinQoS, 1528
TransferRouteMTMessageInfo
transferRouteMessageTlv, 1038
transferRouteMessageTlv, 1037
TlvPresent, 1038
TransferRouteMTMessageInfo, 1038
TransferStatInd, 1038
StatsMask, 1038
StatsPeriod, 1038
transferStats
pack_wds_SLQSSetWdsEventCallback_t, 780
TransferStatsDataType, 1038
interval, 1038
trueIMSI, 1039
imsiT1112, 1040
imsiTS1, 1040
imsiTS2, 1040
imsiTaddrNum, 1040
mccT, 1040
trueSrvStatus
GSMSrvStatusInfo, 309
nas_GSMSrvStatusInfo, 471
tx_bytes
NetStats, 611
sQosFlowStat, 987
unpack_QosFlowStat_t, 1274
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
tx_bytes_drp
sQosFlowStat, 987
unpack_QosFlowStat_t, 1274
tx_errors
NetStats, 611
tx_overflows
NetStats, 611
tx_packets
NetStats, 611
tx_pkt
sQosFlowStat, 987
unpack_QosFlowStat_t, 1274
tx_pkt_drp
sQosFlowStat, 987
unpack_QosFlowStat_t, 1274
tx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
TxByteOKCnt
wds_TXBytesOKTlv, 1526
TxDropConutTlv
QmiCbkWdsStatisticsIndState, 855
unpack_RMTransferStatistics_ind_t, 1276
txInfo, 1041
isInTraffic, 1042
txPower, 1042
txOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1375
TxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 855
unpack_RMTransferStatistics_ind_t, 1276
TxOkConutTlv
QmiCbkWdsStatisticsIndState, 855
unpack_RMTransferStatistics_ind_t, 1276
txPower
nas_txInfo, 554
txInfo, 1042
TxQFilter
unpack_qos_QosFlowInfo_t, 1259
TxQFlowGranted
unpack_qos_QosFlowInfo_t, 1259
type
_getResetInfoNotification, 68
dmsSwiGetResetInfo, 244
pack_wds_GetDefaultProfileNum_t, 761
pack_wds_SetDefaultProfileNum_t, 767
SwiOTAMsg_s, 1005
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1134

unpack_dms_SLQSDmsSwiGetResetInfo_t, 1135
unpack_qmi_t, 1254
typeSize
unpack_sms_GetSMSCAddress_t, 1277

u16PRLVersion
unpack_dms_GetPRLVersion_t, 1122
u8PRLPreference
unpack_dms_GetPRLVersion_t, 1122
UATISIZE
qaGobiApiNas.h, 1897
UDPDstPort
unpack_qos_swiQosFilter_t, 1268
UDPSrcPort
unpack_qos_swiQosFilter_t, 1268
UIMAuthenticateReq, 1068
authData, 1069
pIndicationToken, 1069
sessionInfo, 1069
UIMAuthenticateResp, 1069
pAuthenticateResult, 1070
pCardResult, 1070
pIndicationToken, 1070
UIMChangePIN
qaGobiApiDms.h, 1845
UIMChangePinReq, 1070
changePIN, 1070
pIndicationToken, 1070
pKeyReferenceID, 1070
sessionInfo, 1070
UIMDepersonalizationReq, 1071
depersonilisationInfo, 1071
UIMDepersonalizationResp, 1071
pRemainingRetries, 1071
UIMEventRegisterReqResp, 1071
eventMask, 1072
UIMGetCardStatusResp, 1072
pCardStatus, 1072
pHotSwapStatus, 1072
UIMGetConfigurationReq, 1072
pConfigurationMask, 1073
UIMGetConfigurationResp, 1073
pAutoSelection, 1074
pHaltSubscription, 1074
pPersonalizationStatus, 1074
UIMGetControlKeyStatus
qaGobiApiDms.h, 1846
UIMGetFileAttributesReq, 1074
fileIndex, 1074
pIndicationToken, 1074
sessionInfo, 1074
UIMGetFileAttributesResp, 1074
pCardResult, 1075
pFileAttributes, 1075
pIndicationToken, 1075
UIMGetICCID
qaGobiApiDms.h, 1847
UIMGetPINStatus
qaGobiApiDms.h, 1848
UIMGetSlotsStatusResp, 1075
pNumberOfPhySlot, 1075
pUimSlotsStatus, 1075
UIMPinResp, 1076
pEncryptedPIN1, 1076
pIndicationToken, 1076
pRemainingRetries, 1076
UIMPowerDownReq, 1076
slot, 1077
UIMPowerUpReq, 1077
pIgnoreHotSwapSwitch, 1077
slot, 1077
UIMReadTransparentReq, 1077
fileIndex, 1078
pEncryptData, 1078
pIndicationToken, 1078
readTransparent, 1078
sessionInfo, 1078
UIMReadTransparentResp, 1078
pCardResult, 1079
pEncryptedData, 1079
pIndicationToken, 1079
pReadResult, 1079
UIMRefreshCompleteReq, 1079
refreshComplete, 1080
sessionInfo, 1080
UIMRefreshEvent, 1080
aid, 1081
aidLength, 1081
arrfileInfo, 1081
mode, 1081
numOfFiles, 1081
sessionType, 1081
stage, 1081
UIMRefreshGetLastEventReq, 1081
sessionInfo, 1082
UIMRefreshGetLastEventResp, 1082
pRefreshEvent, 1082
UIMRefreshOKReq, 1082
OKtoRefresh, 1083
sessionInfo, 1083
UIMRefreshRegisterReq, 1083
regRefresh, 1083
sessionInfo, 1083
UIMSessionInformation, 1083
aid, 1084
aidLength, 1084
sessionType, 1084
UIMSetControlKeyProtection
qaGobiApiDms.h, 1849
UIMSetPINProtection
qaGobiApiDms.h, 1849
UIMSetPinProtectionReq, 1084
pIndicationToken, 1085
pKeyReferenceID, 1085
pinProtection, 1085
sessionInfo, 1085
UIMSlotStatus, 1085

bICCID, 1086
bICCIDLength, 1086
bLogicalSlot, 1086
uPhyCardStatus, 1086
uPhySlotStatus, 1086
UIMSlotStatusChangeInfo, 1087
bNumberOfPhySlots, 1087
slotsstatusChange, 1087
UIMSlotsStatus, 1085
uimSlotStatus, 1085
UIMStatusChangeInfo, 1087
statusChange, 1087
UIMSwitchSlotReq, 1087
bLogicalSlot, 1088
ulPhysicalSlot, 1088
UIMUnblockControlKey
qaGobiApiDms.h, 1850
UIMUnblockPIN
qaGobiApiDms.h, 1851
UIMUnblockPinReq, 1088
pIndicationToken, 1089
pKeyReferenceID, 1089
sessionInfo, 1089
unblockPIN, 1089
UIMVerifyPIN
qaGobiApiDms.h, 1852
UIMVerifyPinReq, 1089
pEncryptedPIN1, 1090
pIndicationToken, 1090
pKeyReferenceID, 1090
sessionInfo, 1090
verifyPIN, 1090
ULONG
SwiDataTypes.h, 2187
ULONGLONG
SwiDataTypes.h, 2187
UMTSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1378
UMTSInfo, 1090
cellID, 1091
ecio, 1091
geranInst, 1091
GeranInstInfo, 1091
lac, 1091
plmn, 1091
psc, 1092
rscp, 1092
UMTSInstInfo, 1092
uarfcn, 1092
umtsInst, 1092
UMTSInstInfo
nas_UMTSInfo, 556
UMTSInfo, 1092
UMTSLTENbrCell
nas_WCDMAInfoLTENeighborCell, 561
WCDMAInfoLTENeighborCell, 1498
UMTSMinQoS, 1093
deliveryErrSDU, 1095
grntDownlinkBitrate, 1095
grntUplinkBitrate, 1096
maxDownlinkBitrate, 1096
maxSDUSize, 1096
maxUplinkBitrate, 1096
qosDeliveryOrder, 1096
resBerRatio, 1096
sduErrorRatio, 1096
trafficClass, 1096
trafficPriority, 1096
transferDelay, 1096
UMTSQoS, 1096
deliveryErrSDU, 1098
grntDownlinkBitrate, 1098
grntUplinkBitrate, 1098
maxDownlinkBitrate, 1098
maxSDUSize, 1098
maxUplinkBitrate, 1098
qosDeliveryOrder, 1098
resBerRatio, 1098
sduErrorRatio, 1098
trafficClass, 1098
trafficPriority, 1098
transferDelay, 1098
UMTSReqQoS
LibPackUMTSReqQoSSigInd, 380
UMTSReqQoSSigInd, 1099
UMTSReqQoSSigInd, 1098
SigInd, 1099
UMTSReqQoS, 1099
UMTSinstInfo, 1092
umtsEcio, 1092
umtsPsc, 1092
umtsRscp, 1092
umtsUarfcn, 1092
UNIQUE_ID_LEN
dms.h, 1561
qaGobiApiFms.h, 1857
UNUSEDPARAM
common.h, 1550
SwiDataTypes.h, 2186
uPhyCardStatus
slot_t, 938
UIMSlotStatus, 1086
uPhySlotStatus
slot_t, 938
UIMSlotStatus, 1086
URLAddrInfo, 1389
urlAddr, 1389
uResult
sGetDeviceSeriesResult, 934
USBComp
pack_dms_SetUSBComp_t, 638
unpack_dms_GetUSBComp_t, 1125
USBCompConfig, 1389
pUSBComp, 1390
USBCompParams, 1390
pNumSupUSBComps, 1392

pSupUSBComps, 1392
pUSBComp, 1392
USHORT
SwiDataTypes.h, 2187
USSD_DCS_8BIT
qaGobiApiCbk.h, 1726
USSD_DCS_ASCII
qaGobiApiCbk.h, 1726
USSD_DCS_UCS2
qaGobiApiCbk.h, 1726
USSDNoWaitIndicationInfo, 1393
pAlphaIdentifier, 1393
pError, 1393
pFailureCause, 1393
pUSSDData, 1393
USSDNotificationNetworkInfo
unpack_voice_USSDNotificationCallback_ind_t, 1348
USSDRespFNetwork, 1393
pRespData, 1394
pTypeCode, 1394
USSInfo, 1394
ussDCS, 1394
ussData, 1394
ussLen, 1394
USSInformation
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 754
voiceOrigUSSDNoWaitInfo, 1478
USSResp, 1395
pAlphaIDInfo, 1395
pCCSuppsType, 1395
pCallId, 1395
pCcResultType, 1395
pUSSDInfo, 1395
pfailureCause, 1395
UUSData
UUSInfo, 1396
voice_UUSInfo, 1445
UUSDatalen
UUSInfo, 1396
voice_UUSInfo, 1445
UUSDcs
UUSInfo, 1396
voice_UUSInfo, 1445
UUSInfo, 1395
UUSData, 1396
UUSDatalen, 1396
UUSDcs, 1396
UUSType, 1396
UUSType
UUSInfo, 1396
voice_UUSInfo, 1445
uarfcn
lteWcdmaCellInfo, 435
nas_lteWcdmaCellInfo, 500
nas_UMTSInfo, 556
nas_wcdmaUARFCN, 566
UMTSInfo, 1092
wcdmaUARFCN, 1507
ueInIdle
LTEInfoInterfreq, 418
LTEInfoIntrafreq, 420
LTEInfoNeighboringGSM, 421
LTEInfoNeighboringWCDMA, 422
nas_LTEInfoInterfreq, 486
nas_LTEInfoIntrafreq, 488
nas_LTEInfoNeighboringGSM, 489
nas_LTEInfoNeighboringWCDMA, 490
uim.h, 2201
MAX_ICCID_LENGTH, 2204
MAX_NO_OF_SLOTS, 2204
MAX_SLOTS_STATUS, 2204
pack_uim_ChangePin, 2204
pack_uim_GetCardStatus, 2204
pack_uim_ReadTransparent, 2205
pack_uim_SLQSUIMAuthenticate, 2205
pack_uim_SLQSUIMDepersonalization, 2206
pack_uim_SLQSUIMEventRegister, 2206
pack_uim_SLQSUIMGetConfiguration, 2207
pack_uim_SLQSUIMGetFileAttributes, 2207
pack_uim_SLQSUIMGetSlotsStatus, 2207
pack_uim_SLQSUIMPowerDown, 2208
pack_uim_SLQSUIMPowerUp, 2208
pack_uim_SLQSUIMRefreshComplete, 2208
pack_uim_SLQSUIMRefreshGetLastEvent, 2209
pack_uim_SLQSUIMRefreshOK, 2209
pack_uim_SLQSUIMRefreshRegister, 2210
pack_uim_SLQSUIMReset, 2210
pack_uim_SLQSUIMSwitchSlot, 2210
pack_uim_SetPinProtection, 2205
pack_uim_UnblockPin, 2211
pack_uim_VerifyPin, 2211
unpack_uim_ChangePin, 2211
unpack_uim_GetCardStatus, 2212
unpack_uim_ReadTransparent, 2212
unpack_uim_SLQSUIMAuthenticate, 2213
unpack_uim_SLQSUIMDepersonalization, 2214
unpack_uim_SLQSUIMEventRegister, 2214
unpack_uim_SLQSUIMGetConfiguration, 2214
unpack_uim_SLQSUIMGetFileAttributes, 2215
unpack_uim_SLQSUIMGetSlotsStatus, 2215
unpack_uim_SLQSUIMPowerDown, 2215
unpack_uim_SLQSUIMPowerUp, 2216
unpack_uim_SLQSUIMRefreshCallback_Ind, 2216
unpack_uim_SLQSUIMRefreshComplete, 2216
unpack_uim_SLQSUIMRefreshGetLastEvent, 2217
unpack_uim_SLQSUIMRefreshOK, 2217
unpack_uim_SLQSUIMRefreshRegister, 2217
unpack_uim_SLQSUIMReset, 2218
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind, 2218
unpack_uim_SLQSUIMSwitchSlot, 2218
unpack_uim_SetPinProtection, 2212
unpack_uim_SetUimSlotStatusChangeCallback_ind, 2213

unpack_uim_UnblockPin, 2219
unpack_uim_VerifyPin, 2219
uim_UIMSessionInformation, 1066
aid, 1067
aidLength, 1067
sessionType, 1067
uim_appStatus, 1043
aidLength, 1046
aidVal, 1046
appState, 1046
appType, 1046
persoFeature, 1046
persoRetries, 1046
persoState, 1046
persoUnblockRetries, 1046
pin1Retries, 1046
pin1State, 1046
pin2Retries, 1046
pin2State, 1046
puk1Retries, 1046
puk2Retries, 1046
univPin, 1046
uim_authenticateResult, 1046
content, 1047
contentLen, 1047
uim_authenticationData, 1047
context, 1048
data, 1048
dataLen, 1048
uim_cardResult, 1048
sw1, 1049
sw2, 1049
uim_cardStatus, 1049
index1xPri, 1050
index1xSec, 1050
indexGwPri, 1050
indexGwSec, 1050
numSlot, 1050
SlotInfo, 1050
uim_changeUIMPIN, 1050
oldPINLen, 1051
oldPINVal, 1051
pinID, 1051
pinLen, 1051
pinVal, 1051
uim_depersonalizationInformation, 1051
ckLen, 1052
ckVal, 1052
feature, 1052
operation, 1052
uim_encryptedPIN1, 1052
pin1Len, 1053
pin1Val, 1053
uim_fileAttributes, 1053
fileID, 1056
fileSize, 1056
fileType, 1056
rawLen, 1056
rawValue, 1056
recordCount, 1056
recordSize, 1056
secActivate, 1056
secActivateMask, 1056
secDeactivate, 1056
secDeactivateMask, 1056
secIncrease, 1056
secIncreaseMask, 1056
secRead, 1056
secReadMask, 1056
secWrite, 1056
secWriteMask, 1056
uim_fileInfo, 1056
fileID, 1057
path, 1057
pathLen, 1057
uim_hotSwapStatus, 1057
hotSwap, 1057
hotSwapLength, 1057
uim_personalizationStatus, 1057
feature, 1058
numFeatures, 1058
unblockLeft, 1058
verifyLeft, 1058
uim_readResult, 1058
content, 1059
contentLen, 1059
uim_readTransparentInfo, 1059
length, 1059
offset, 1059
uim_refreshevent, 1059
aid, 1061
aidLength, 1061
arrfileInfo, 1061
mode, 1061
numOfFiles, 1061
sessionType, 1061
stage, 1061
uim_registerRefresh, 1061
arrfileInfo, 1062
numFiles, 1062
registerFlag, 1062
voteForInit, 1062
uim_remainingRetries, 1062
unblockLeft, 1062
verifyLeft, 1062
uim_sessionInformation, 1063
aid, 1063
aidLength, 1063
sessionType, 1063
uim_setPINProtection, 1063
pinID, 1064
pinLength, 1064
pinOperation, 1064
pinValue, 1064
uim_slotInfo, 1064
AppStatus, 1066

cardState, 1066
errorState, 1066
numApp, 1066
upinRetries, 1066
upinState, 1066
upukRetries, 1066
uim_unblockUIMPIN, 1067
newPINLen, 1067
newPINVal, 1067
pinID, 1067
pukLen, 1067
pukVal, 1068
uim_verifyUIMPIN, 1068
pinID, 1068
pinLen, 1068
pinVal, 1068
uimSlotStatus
slots_t, 942
UIMSlotsStatus, 1085
ulData
DataULongTlv, 223
wds_DataULongTlv, 1512
ulMask
rmTrasnferStaticsReq, 884
swiRMTrasnferStaticsReq, 1016
ulMax
_QmiSwiAvmsSetSettingsPeriodInfo, 82
ulMin
_QmiSwiAvmsSetSettingsPeriodInfo, 82
ulPhysicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 739
UIMSwitchSlotReq, 1088
ulPkgDownloadComplete
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
ulPkgDownloadSize
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 75
ullData
DataULongLongTlv, 223
wds_DataULongLongTlv, 1511
umtsEcio
nas_UMTSinstInfo, 557
UMTSinstInfo, 1092
umtsInst
nas_UMTSInfo, 556
UMTSInfo, 1092
umtsLTENbrCell, 1092
cellIsTDD, 1093
earfcn, 1093
pci, 1093
rsrp, 1093
rsrq, 1093
srxlev, 1093
umtsLTENbrCellLen
nas_WCDMAInfoLTENeighborCell, 561
WCDMAInfoLTENeighborCell, 1498
umtsPsc
nas_UMTSinstInfo, 557
UMTSinstInfo, 1092
umtsRscp
nas_UMTSinstInfo, 557
UMTSinstInfo, 1092
umtsUarfcn
nas_UMTSinstInfo, 557
UMTSinstInfo, 1092
UnPackGetProfileSettingOut, 1387
curProfile, 1388
pExtErrCode, 1388
unblockLeft
personalizationStatus, 794
remainingRetries, 875
uim_personalizationStatus, 1058
uim_remainingRetries, 1062
unblockPIN
UIMUnblockPinReq, 1089
unblockRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1151
unpack_dms_UIMSetPINProtection_t, 1155
unpack_dms_UIMUnblockControlKey_t, 1155
unblockUIMPIN, 1099
newPINLen, 1100
newPINVal, 1100
pinID, 1100
pukLen, 1100
pukVal, 1100
uniqueID
CurrImageInfo, 206
image_info_t, 324
univPin
appStats, 126
appStatus, 129
uim_appStatus, 1046
UniversalTime, 1100
day, 1101
dayOfWeek, 1101
hour, 1101
minute, 1101
month, 1101
second, 1101
year, 1101
universalTime
nasNetworkTime, 588
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1226
unpack_QosFlowStat_t, 1274
bearerId, 1274
tx_bytes, 1274
tx_bytes_drp, 1274
tx_pkt, 1274
tx_pkt_drp, 1274
unpack_RMTransferStatistics_ind_t, 1275
RxDropConutTlv, 1276
RxOkByteCountTlv, 1276
RxOkConutTlv, 1276
TxDropConutTlv, 1276

TxOkByteCountTlv, 1276
TxOkConutTlv, 1276
unpack_dms_ActivateAutomatic
dms.h, 1585
unpack_dms_GetActivationState
dms.h, 1586
unpack_dms_GetActivationState_t, 1101
state, 1102
unpack_dms_GetBandCapability
dms.h, 1586
unpack_dms_GetBandCapability_t, 1102
BandCapability, 1103
Tlvresult, 1104
unpack_dms_GetCrashAction
dms.h, 1587
unpack_dms_GetCrashAction_t, 1104
DevCrashState, 1104
Tlvresult, 1104
unpack_dms_GetCustFeature
dms.h, 1587
unpack_dms_GetCustFeature_t, 1104
DHCPRelayEnabled, 1106
DisableIMSI, 1106
GPSLPM, 1106
GPSSel, 1106
GpsEnable, 1106
IPFamSupport, 1106
IsVoiceEnabled, 1106
RMAutoConnect, 1107
SMSSupport, 1107
Tlvresult, 1107
unpack_dms_GetCustFeaturesV2
dms.h, 1587
unpack_dms_GetCustFeaturesV2_t, 1107
GetCustomFeatureV2, 1107
Tlvresult, 1107
unpack_dms_GetDeviceCap
dms.h, 1588
unpack_dms_GetDeviceCap_t, 1107
DataServiceCapability, 1108
MaxRXChannelRate, 1108
MaxTXChannelRate, 1108
RadioIfaces, 1108
RadioIfacesSize, 1108
SimCapability, 1108
Tlvresult, 1108
unpack_dms_GetDeviceCapabilities
dms.h, 1588
unpack_dms_GetDeviceCapabilities_t, 1108
dataServiceCaCapability, 1109
maxRxChannelRate, 1109
maxTxChannelRate, 1109
RadioIfaces, 1110
radioIfacesSize, 1110
simCapability, 1110
unpack_dms_GetDeviceHardwareRev
dms.h, 1588
unpack_dms_GetDeviceHardwareRev_t, 1110
String, 1110
stringSize, 1110
Tlvresult, 1110
unpack_dms_GetDeviceMfr
dms.h, 1589
unpack_dms_GetDeviceMfr_t, 1110
String, 1111
stringSize, 1111
Tlvresult, 1111
unpack_dms_GetDeviceSerialNumbers
dms.h, 1589
unpack_dms_GetDeviceSerialNumbers_t, 1111
ESNString, 1112
esnSize, 1112
IMEIString, 1112
imeiSize, 1112
imeiSvnSize, 1112
ImeiSvnString, 1112
MEIDString, 1112
meidSize, 1112
Tlvresult, 1112
unpack_dms_GetFSN
dms.h, 1590
unpack_dms_GetFSN_t, 1116
String, 1117
Tlvresult, 1117
unpack_dms_GetFirmwareInfo
dms.h, 1589
unpack_dms_GetFirmwareInfo_t, 1112
appversion_str, 1114
bootversion_str, 1114
carrier_str, 1114
cur_carr_name, 1114
cur_carr_rev, 1114
modelid_str, 1114
packageid_str, 1114
priversion_str, 1114
sku_str, 1114
Tlvresult, 1114
unpack_dms_GetFirmwareRevision
dms.h, 1590
unpack_dms_GetFirmwareRevision_t, 1114
AMSSString, 1115
amssSize, 1115
PRIString, 1115
Tlvresult, 1115
unpack_dms_GetFirmwareRevisions
dms.h, 1590
unpack_dms_GetFirmwareRevisions_t, 1115
AMSSString, 1116
amssSize, 1116
bootSize, 1116
BootString, 1116
PRIString, 1116
priSize, 1116
Tlvresult, 1116
unpack_dms_GetHardwareRevision
dms.h, 1591

unpack_dms_GetHardwareRevision_t, 1117
hwVer, 1117
unpack_dms_GetIMSI
dms.h, 1591
unpack_dms_GetIMSI_t, 1117
imsi, 1117
Tlvresult, 1117
unpack_dms_GetManufacturer
dms.h, 1591
unpack_dms_GetManufacturer_t, 1118
manufacturer, 1118
Tlvresult, 1118
unpack_dms_GetModelID
dms.h, 1592
unpack_dms_GetModelID_t, 1118
modelid, 1118
Tlvresult, 1118
unpack_dms_GetNetworkTime
dms.h, 1592
unpack_dms_GetNetworkTime_t, 1119
source, 1119
timestamp, 1119
Tlvresult, 1119
unpack_dms_GetOfflineReason
dms.h, 1592
unpack_dms_GetOfflineReason_t, 1119
pReasonMask, 1120
pbPlatform, 1120
Tlvresult, 1120
unpack_dms_GetPRLVersion
dms.h, 1593
unpack_dms_GetPRLVersion_t, 1122
Tlvresult, 1122
u16PRLVersion, 1122
u8PRLPreference, 1122
unpack_dms_GetPower
dms.h, 1593
unpack_dms_GetPower_t, 1120
HardwareControlledMode, 1121
OfflineReason, 1121
OperationMode, 1121
Tlvresult, 1121
unpack_dms_GetSerialNumbers
dms.h, 1593
unpack_dms_GetSerialNumbers_t, 1122
esn, 1123
imei_no, 1123
imeisv_svn, 1123
meid, 1123
unpack_dms_GetUSBComp
dms.h, 1594
unpack_dms_GetUSBComp_t, 1123
NumSupUSBComps, 1125
SupUSBComps, 1125
Tlvresult, 1125
USBComp, 1125
unpack_dms_GetVoiceNumber
dms.h, 1594
unpack_dms_GetVoiceNumber_t, 1125
MIN, 1126
minSize, 1126
Tlvresult, 1126
VoiceNumber, 1126
voiceNumberSize, 1126
unpack_dms_PSMCfgChange_ind
dms.h, 1595
unpack_dms_PSMCfgChange_ind_t, 1126
ActiveTimer, 1127
EnableState, 1127
PeriodicUpdateTimer, 1127
Tlvresult, 1127
unpack_dms_ResetToFactoryDefaults
dms.h, 1595
unpack_dms_ResetToFactoryDefaults_t, 1127
Tlvresult, 1127
unpack_dms_SLQSDmsSwiGetPCInfo
dms.h, 1599
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1131
has_LpmFlag, 1133
has_PersistentLpm, 1133
has_Wdisable, 1133
LpmFlag, 1133
opMode, 1133
PersistentLpm, 1133
PowerOffMode, 1133
Wdisable, 1133
unpack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1599
unpack_dms_SLQSDmsSwiGetResetInfo_Ind
dms.h, 1600
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1133
source, 1134
Tlvresult, 1134
type, 1134
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1134
source, 1135
Tlvresult, 1135
type, 1135
unpack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1600
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1135
Tlvresult, 1136
unpack_dms_SLQSGetBandCapability
dms.h, 1600
unpack_dms_SLQSGetBandCapability_t, 1136
bandCapability, 1139
LteBandCapability, 1139
TdsBandCapability, 1139
unpack_dms_SLQSGetERIFile
dms.h, 1601
unpack_dms_SLQSGetERIFile_t, 1139
eriFile, 1139
Tlvresult, 1139
unpack_dms_SLQSGetPowerSaveModeConfig
dms.h, 1601
unpack_dms_SLQSGetPowerSaveModeConfig_t, 1139

pActiveTimer, 1140
pDurationThreshold, 1140
pEarlyWakeupTime, 1140
pPeriodicUpdateTimer, 1140
pPsmEnableState, 1140
pRandomizationWindow, 1140
unpack_dms_SLQSSetPowerSaveModeConfig
dms.h, 1601
unpack_dms_SLQSSetPowerSaveModeConfig_t, 1141
Tlvresult, 1141
unpack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1602
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 1141
Tlvresult, 1141
unpack_dms_SLQSSwiGetCrashInfo
dms.h, 1602
unpack_dms_SLQSSwiGetCrashInfo_t, 1141
crashInfoParam, 1142
Tlvresult, 1142
unpack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1603
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1142
pGetDyingGaspCfg, 1142
Tlvresult, 1142
unpack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1603
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1142
pGetDyingGaspStatistics, 1143
Tlvresult, 1143
unpack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1603
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1143
carrier, 1144
fwvers, 1144
numEntries, 1144
pCurrImgInfo, 1144
pkgver, 1144
priver, 1144
unpack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1604
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1144
imgType, 1145
logString, 1145
refData, 1145
refString, 1146
ResCode, 1146
Tlvresult, 1146
unpack_dms_SLQSSwiGetHostDevInfo
dms.h, 1604
unpack_dms_SLQSSwiGetHostDevInfo_t, 1146
manString, 1146
modelString, 1146
plasmaIDString, 1147
swVerString, 1147
Tlvresult, 1147
unpack_dms_SLQSSwiGetOSInfo
dms.h, 1604
unpack_dms_SLQSSwiGetOSInfo_t, 1147
nameString, 1147
Tlvresult, 1147
versionString, 1147
unpack_dms_SLQSSwiGetSerialNoExt
dms.h, 1605
unpack_dms_SLQSSwiGetSerialNoExt_t, 1147
meidString, 1148
Tlvresult, 1148
unpack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1605
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1148
Tlvresult, 1148
unpack_dms_SLQSSwiSetHostDevInfo
dms.h, 1605
unpack_dms_SLQSSwiSetHostDevInfo_t, 1148
Tlvresult, 1149
unpack_dms_SLQSSwiSetOSInfo
dms.h, 1606
unpack_dms_SLQSSwiSetOSInfo_t, 1149
Tlvresult, 1149
unpack_dms_SLQSUIMGetState
dms.h, 1606
unpack_dms_SLQSUIMGetState_t, 1149
state, 1149
Tlvresult, 1149
unpack_dms_SetActivationStatusCallback
dms.h, 1595
unpack_dms_SetActivationStatusCallback_t, 1127
Tlvresult, 1128
unpack_dms_SetCrashAction
dms.h, 1596
unpack_dms_SetCrashAction_t, 1128
notused, 1128
unpack_dms_SetCustFeature
dms.h, 1596
unpack_dms_SetCustFeature_t, 1128
Tlvresult, 1128
unpack_dms_SetCustFeaturesV2
dms.h, 1596
unpack_dms_SetCustFeaturesV2_t, 1129
Tlvresult, 1129
unpack_dms_SetEventReport
dms.h, 1597
unpack_dms_SetEventReport_ind
dms.h, 1597
unpack_dms_SetEventReport_ind_t, 1129
ActivationStatusTlv, 1129
OperatingModeTlv, 1129
Tlvresult, 1129
unpack_dms_SetEventReport_t, 1130
Tlvresult, 1130
unpack_dms_SetFirmwarePreference
dms.h, 1597
unpack_dms_SetFirmwarePreference_t, 1130
Tlvresult, 1130
unpack_dms_SetIndicationRegister
dms.h, 1598

unpack_dms_SetIndicationRegister_t, 1130
Tlvresult, 1131
unpack_dms_SetPower
dms.h, 1598
unpack_dms_SetPower_t, 1131
Tlvresult, 1131
unpack_dms_SetUSBComp
dms.h, 1598
unpack_dms_SetUSBComp_t, 1131
Tlvresult, 1131
unpack_dms_SwiEventReportCallBack_ind
dms.h, 1606
unpack_dms_SwiEventReportCallBack_ind_t, 1149
TempTlv, 1150
VoltTlv, 1150
unpack_dms_SwiSetEventReport
dms.h, 1607
unpack_dms_SwiSetEventReport_t, 1150
Tlvresult, 1150
unpack_dms_UIMChangePIN
dms.h, 1607
unpack_dms_UIMGetControlKeyStatus
dms.h, 1607
unpack_dms_UIMGetControlKeyStatus_t, 1150
facilityState, 1151
Tlvresult, 1151
unblockRetriesLeft, 1151
verifyRetriesLeft, 1151
unpack_dms_UIMGetICCID
dms.h, 1608
unpack_dms_UIMGetICCID_t, 1151
String, 1152
stringSize, 1152
Tlvresult, 1152
unpack_dms_UIMGetPINStatus
dms.h, 1608
unpack_dms_UIMGetPINStatus_t, 1152
p1Status, 1153
p1UnblockRetriesLeft, 1153
p1VerifyRetriesLeft, 1153
p2Status, 1153
p2UnblockRetriesLeft, 1153
p2VerifyRetriesLeft, 1153
Tlvresult, 1153
unpack_dms_UIMSetControlKeyProtection
dms.h, 1608
unpack_dms_UIMSetControlKeyProtection_t, 1154
Tlvresult, 1154
verifyRetriesLeft, 1154
unpack_dms_UIMSetPINProtection
dms.h, 1609
unpack_dms_UIMSetPINProtection_t, 1154
Tlvresult, 1155
unblockRetriesLeft, 1155
verifyRetriesLeft, 1155
unpack_dms_UIMUnblockControlKey
dms.h, 1609
unpack_dms_UIMUnblockControlKey_t, 1155
Tlvresult, 1155
unblockRetriesLeft, 1155
unpack_dms_UIMUnblockPIN
dms.h, 1610
unpack_dms_UIMVerifyPIN
dms.h, 1610
unpack_dms_ValidateSPC
dms.h, 1610
unpack_fms_GetImagesPreference
fms.h, 1613
unpack_fms_GetImagesPreference_t, 1155
ImageListSize, 1156
pImageList, 1156
Tlvresult, 1156
unpack_fms_GetStoredImages
fms.h, 1613
unpack_fms_GetStoredImages_t, 1156
imageList, 1156
imagelistSize, 1156
Tlvresult, 1156
unpack_fms_SetImagesPreference
fms.h, 1613
unpack_fms_SetImagesPreference_t, 1157
ImageTypes, 1157
ImageTypesSize, 1157
Tlvresult, 1157
unpack_loc_BestAvailPos_Ind
loc.h, 1625
unpack_loc_BestAvailPos_Ind_t, 1157
pAltitudeWrtEllipsoid, 1162
pAltitudeWrtMeanSeaLevel, 1162
pGpsTime, 1162
pHeading, 1162
pHeadingUnc, 1162
pHorCirConf, 1162
pHorEllpConf, 1162
pHorReliability, 1162
pHorUncCircular, 1162
pHorUncEllipseOrientAzimuth, 1162
pHorUncEllipseSemiMajor, 1162
pHorUncEllipseSemiMinor, 1162
pLatitude, 1162
pLongitude, 1162
pMagneticDeviation, 1162
pPrecisionDilution, 1162
pSensorDataUsage, 1162
pSpeedHorizontal, 1162
pSpeedUnc, 1162
pSpeedVertical, 1162
pSpeedVerticalUnc, 1162
pSvUsedforFix, 1162
pTechnologyMask, 1162
pTimeSrc, 1162
pTimeUnc, 1162
pTimestampUtc, 1162
pVertConfidence, 1162
pVertReliability, 1162
pVertUnc, 1163

pXid, 1163
status, 1163
TlvResult, 1163
unpack_loc_CradleMountCallback_Ind
loc.h, 1625
unpack_loc_CradleMountCallback_Ind_t, 1163
cradleMountConfigStatus, 1163
unpack_loc_Delete_Assist_Data_t, 1163
TlvResult, 1164
unpack_loc_DeleteAssistData
loc.h, 1626
unpack_loc_DeleteAssistData_Ind
loc.h, 1626
unpack_loc_DeleteAssistData_Ind_t, 1164
status, 1164
TlvResult, 1164
unpack_loc_EngineState_Ind
loc.h, 1626
unpack_loc_EngineState_Ind_t, 1165
engineState, 1165
TlvResult, 1165
unpack_loc_EventNMEA_Ind
loc.h, 1627
unpack_loc_EventNMEA_Ind_t, 1165
NMEAData, 1166
TlvResult, 1166
unpack_loc_EventRegister
loc.h, 1627
unpack_loc_EventRegister_t, 1166
TlvResult, 1166
unpack_loc_EventTimeSyncCallback_Ind
loc.h, 1627
unpack_loc_EventTimeSyncCallback_Ind_t, 1166
timeSyncRefCounter, 1166
unpack_loc_GetServer_Ind
loc.h, 1628
unpack_loc_GetServer_Ind_t, 1166
pIPv4AddrInfo, 1168
pIPv6AddrInfo, 1168
pURL, 1168
serverStatus, 1168
serverType, 1168
TlvResult, 1168
unpack_loc_GnssSvInfo_Ind
loc.h, 1628
unpack_loc_GnssSvInfo_Ind_t, 1168
altitudeAssumed, 1168
pSatelliteInfo, 1168
TlvResult, 1168
unpack_loc_InjectPositionCallback_Ind
loc.h, 1628
unpack_loc_InjectPositionCallback_Ind_t, 1168
status, 1169
unpack_loc_InjectSensorDataCallback_Ind
loc.h, 1629
unpack_loc_InjectSensorDataCallback_Ind_t, 1169
injectSensorDataStatus, 1170
pAccelSamplesAccepted, 1170
pAccelTempSamplesAccepted, 1170
pGyroSamplesAccepted, 1170
pGyroTempSamplesAccepted, 1170
pOpaqueIdentifier, 1170
unpack_loc_InjectTimeSyncDataCallback_Ind
loc.h, 1629
unpack_loc_InjectTimeSyncDataCallback_Ind_t, 1171
injectTimeSyncStatus, 1171
unpack_loc_InjectUTCTimeCallback_Ind
loc.h, 1629
unpack_loc_InjectUTCTimeCallback_Ind_t, 1171
status, 1172
unpack_loc_PositionRpt_Ind
loc.h, 1630
unpack_loc_PositionRpt_Ind_t, 1172
pAltitudeAssumed, 1177
pAltitudeWrtEllipsoid, 1177
pAltitudeWrtMeanSeaLevel, 1177
pFixId, 1178
pGpsTime, 1178
pHeading, 1178
pHeadingUnc, 1178
pHorConfidence, 1178
pHorReliability, 1178
pHorUncCircular, 1178
pHorUncEllipseOrientAzimuth, 1178
pHorUncEllipseSemiMajor, 1178
pHorUncEllipseSemiMinor, 1178
pLatitude, 1178
pLeapSeconds, 1178
pLongitude, 1178
pMagneticDeviation, 1178
pPrecisionDilution, 1178
pSensorDataUsage, 1178
pSpeedHorizontal, 1178
pSpeedUnc, 1178
pSpeedVertical, 1178
pSvUsedforFix, 1178
pTechnologyMask, 1178
pTimeSrc, 1178
pTimeUnc, 1178
pTimestampUtc, 1178
pVertConfidence, 1178
pVertReliability, 1178
pVertUnc, 1178
sessionId, 1178
sessionStatus, 1179
TlvResult, 1179
unpack_loc_SLQSLOCGetBestAvailPos
loc.h, 1632
unpack_loc_SLQSLOCGetBestAvailPos_t, 1183
TlvResult, 1183
unpack_loc_SLQSLOCGetServer
loc.h, 1632
unpack_loc_SLQSLOCInjectPosition
loc.h, 1633
unpack_loc_SLQSLOCInjectSensorData
loc.h, 1633

unpack_loc_SLQSLOCInjectUTCTime
loc.h, 1633
unpack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1634
unpack_loc_SLQSLOCSetServer
loc.h, 1634
unpack_loc_SensorStreamingCallback_Ind
loc.h, 1630
unpack_loc_SensorStreamingCallback_Ind_t, 1179
pAccelAcceptReady, 1179
pAccelTempAcceptReady, 1179
pGyroAcceptReady, 1179
pGyroTempAcceptReady, 1179
unpack_loc_SetExtPowerConfig_Ind
loc.h, 1630
unpack_loc_SetExtPowerConfig_Ind_t, 1179
status, 1180
TlvResult, 1180
unpack_loc_SetExtPowerState
loc.h, 1631
unpack_loc_SetExtPowerState_t, 1180
TlvResult, 1180
unpack_loc_SetOperationMode
loc.h, 1631
unpack_loc_SetOperationMode_Ind
loc.h, 1631
unpack_loc_SetOperationMode_Ind_t, 1181
status, 1181
TlvResult, 1181
unpack_loc_SetOperationMode_t, 1181
TlvResult, 1182
unpack_loc_SetServer_Ind
loc.h, 1632
unpack_loc_SetServer_Ind_t, 1182
serverStatus, 1182
TlvResult, 1182
unpack_loc_Start
loc.h, 1634
unpack_loc_Start_t, 1183
TlvResult, 1183
unpack_loc_Stop
loc.h, 1635
unpack_loc_Stop_t, 1183
TlvResult, 1184
unpack_nas_GetACCOLC
nas.h, 1663
unpack_nas_GetANAAAAuthenticationStatus
nas.h, 1663
unpack_nas_GetCDMANetworkParameters
nas.h, 1664
unpack_nas_GetCDMANetworkParameters_t, 1184
Application, 1186
Broadcast, 1186
CustomSCP, 1186
ForceRev0, 1186
Protocol, 1186
RegForeignNID, 1186
RegForeignSID, 1186
RegHomeSID, 1186
Roaming, 1186
SCI, 1186
SCM, 1186
unpack_nas_GetHomeNetwork
nas.h, 1664
unpack_nas_GetHomeNetwork3GPP2
nas.h, 1664
unpack_nas_GetHomeNetwork3GPP2_t, 1186
nameSize, 1188
pMCC, 1188
pMNC, 1188
pNID, 1188
pName, 1188
pNw2DescDisp, 1188
pNw2DescEnc, 1188
pNw2DescLen, 1188
pNw2MCC, 1188
pNw2MNC, 1188
pNw2Name, 1188
pSID, 1188
unpack_nas_GetHomeNetwork_t, 1188
mcc, 1189
mnc, 1189
name, 1189
nid, 1189
sid, 1189
unpack_nas_GetNetworkPreference
nas.h, 1665
unpack_nas_GetNetworkPreference_t, 1189
ActiveTechPref, 1190
Duration, 1190
PersistentTechPref, 1190
TlvResult, 1190
unpack_nas_GetRFInfo
nas.h, 1665
unpack_nas_GetRFInfo_t, 1190
instancesSize, 1191
RFBandInfoElements, 1191
unpack_nas_GetServingNetwork
nas.h, 1665
unpack_nas_GetServingNetwork_t, 1191
CSDomain, 1193
DataCaps, 1193
DataCapsLen, 1193
MCC, 1193
MNC, 1193
Name, 1193
nameSize, 1193
PSDomain, 1193
RAN, 1193
RadioIfaces, 1193
RadioIfacesSize, 1193
RegistrationState, 1193
Roaming, 1193
unpack_nas_GetServingNetworkCapabilities
nas.h, 1666
unpack_nas_GetServingNetworkCapabilities_t, 1193

DataCaps, 1194
DataCapsLen, 1194
unpack_nas_GetSignalStrengths
nas.h, 1666
unpack_nas_GetSignalStrengths_t, 1194
len, 1194
radio, 1195
rssi, 1195
unpack_nas_InitiateDomainAttach
nas.h, 1666
unpack_nas_PerformNetworkScan
nas.h, 1667
unpack_nas_PerformNetworkScan_t, 1195
p3GppNetworkInfoInstances, 1196
p3GppNetworkInstanceSize, 1196
pPCSInstance, 1196
pPCSInstanceSize, 1196
pRATINstance, 1196
pRATInstanceSize, 1196
pScanResult, 1196
unpack_nas_SLQSConfigSigInfo
nas.h, 1670
unpack_nas_SLQSGetErrorRate
nas.h, 1671
unpack_nas_SLQSGetErrorRate_t, 1199
pCDMAFrameErrRate, 1200
pGSMBER, 1200
pHDRPackErrRate, 1200
pWCDMABER, 1200
unpack_nas_SLQSGetNetworkTime
nas.h, 1671
unpack_nas_SLQSGetNetworkTime_t, 1201
p3GPP2TimeInfo, 1201
p3GPPTimeInfo, 1201
unpack_nas_SLQSGetOperatorNameData
nas.h, 1672
unpack_nas_SLQSGetOperatorNameData_t, 1201
pNITZInformation, 1202
pOperatorNameString, 1202
pOperatorPLMNList, 1202
pPLMNNetworkName, 1202
pSrvcProviderName, 1202
unpack_nas_SLQSGetPLMNName
nas.h, 1672
unpack_nas_SLQSGetPLMNName_t, 1202
longName, 1204
longNameCI, 1204
longNameEn, 1204
longNameLen, 1204
longNameSB, 1204
shortName, 1204
shortNameCI, 1204
shortNameEn, 1205
shortNameLen, 1205
shortNameSB, 1205
spn, 1205
spnEncoding, 1205
spnLength, 1205
unpack_nas_SLQSGetServingSystem
nas.h, 1673
unpack_nas_SLQSGetServingSystem_t, 1205
BasestationID, 1208
BasestationLatitude, 1208
BasestationLongitude, 1208
CDMASystemInfoExt, 1208
CallBarStatus, 1208
CellID, 1208
ConcSvcInfo, 1208
CurrentPLMN, 1208
DTMInd, 1208
DataSrvCapabilities, 1208
DefaultRoamInd, 1208
DetailedSvcInfo, 1208
Gpp2TimeZone, 1208
GppNetworkDSTAdjustment, 1208
GppTimeZone, 1208
HdrPersonality, 1208
Lac, 1208
NetworkID, 1208
PRLInd, 1209
RoamIndicatorVal, 1209
RoamingIndicatorList, 1209
ServingSystem, 1209
SystemID, 1209
TrackAreaCode, 1209
unpack_nas_SLQSGetSignalStrength
nas.h, 1673
unpack_nas_SLQSGetSignalStrength_t, 1209
ecioList, 1211
ecioListLen, 1211
errorRateList, 1211
errorRateListLen, 1211
Io, 1211
ltersrp, 1211
ltesnr, 1211
rsrqInfo, 1211
rxSignalStrengthList, 1211
rxSignalStrengthListLen, 1211
signalStrengthReqMask, 1211
sinr, 1211
unpack_nas_SLQSGetSysInfo
nas.h, 1673
unpack_nas_SLQSGetSysInfo_t, 1211
pAddCDMASysInfo, 1214
pAddGSMSysInfo, 1214
pAddHDRSysInfo, 1214
pAddLTESysInfo, 1214
pCDMASrvStatusInfo, 1214
pCDMASysInfo, 1214
pGSMCallBarringSysInfo, 1214
pGSMCipherDomainSysInfo, 1214
pGSMSrvStatusInfo, 1214
pGSMSysInfo, 1214
pHDRSrvStatusInfo, 1214
pHDRSysInfo, 1214
pLTESrvStatusInfo, 1214

pLTESysInfo, 1214
pLTEVoiceSupportSysInfo, 1214
pWCDMASysInfo, 1215
unpack_nas_SLQSGetSysSelectionPref
nas.h, 1674
unpack_nas_SLQSGetSysSelectionPref_t, 1215
pBandPref, 1219
pEmerMode, 1219
pGWAcqOrderPref, 1219
pLTEBandPref, 1219
pModePref, 1219
pNetSelPref, 1219
pPRLPref, 1219
pRoamPref, 1219
pSrvDomainPref, 1219
unpack_nas_SLQSInitiateNetworkRegistration
nas.h, 1674
unpack_nas_SLQSNASGeteDRXParams
nas.h, 1675
unpack_nas_SLQSNASGeteDRXParams_t, 1222
pCycleLen, 1222
pEdrxEnable, 1222
unpack_nas_SLQSNASSeteDRXParams
nas.h, 1678
unpack_nas_SLQSNASSwiGetChannelLock
nas.h, 1679
unpack_nas_SLQSNASSwiGetChannelLock_t, 1228
unpack_nas_SLQSNASSwiSetChannelLock
nas.h, 1680
unpack_nas_SLQSNasConfigSigInfo2
nas.h, 1674
unpack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1675
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1219
pDirNum, 1220
unpack_nas_SLQSNasGetCellLocationInfo
nas.h, 1675
unpack_nas_SLQSNasGetCellLocationInfo_t, 1220
pCDMAInfo, 1221
pLTEInfoInterfreq, 1222
pLTEInfoIntrafreq, 1222
pUMTSInfo, 1222
unpack_nas_SLQSNasGetHDRColorCode
nas.h, 1676
unpack_nas_SLQSNasGetHDRColorCode_t, 1223
pColorCode, 1223
unpack_nas_SLQSNasGetRFInfo
nas.h, 1676
unpack_nas_SLQSNasGetRFInfo_t, 1223
pLTEOperationMode, 1224
pRfBandInfoExtFormat, 1224
pRfBandwidthInfo, 1224
pRfDedicatedBandInfo, 1224
rfbandInfoList, 1224
Tlvresult, 1224
unpack_nas_SLQSNasGetSigInfo
nas.h, 1676
unpack_nas_SLQSNasGetSigInfo_t, 1224
CDMASSInfo, 1225
GSMSSInfo, 1225
HDRSSInfo, 1225
LTESSInfo, 1225
unpack_nas_SLQSNasGetTxRxInfo
nas.h, 1677
unpack_nas_SLQSNasGetTxRxInfo_t, 1225
pRXChain0Info, 1225
pRXChain1Info, 1225
pTXInfo, 1225
unpack_nas_SLQSNasIndicationRegisterExt
nas.h, 1677
unpack_nas_SLQSNasNetworkTimeCallBack_ind
nas.h, 1678
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1225
pDayltSavAdj, 1226
pRadioInterface, 1226
pTimeZone, 1226
universalTime, 1226
unpack_nas_SLQSNasSigInfoCallback_ind
nas.h, 1678
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1226
pRscp, 1227
unpack_nas_SLQSNasSwiIndicationRegister
nas.h, 1679
unpack_nas_SLQSNasSwiModemStatus
nas.h, 1679
unpack_nas_SLQSNasSwiModemStatus_t, 1228
commonInfo, 1229
pLTEInfo, 1229
unpack_nas_SLQSNasSwiOTAMessageCallback_ind
nas.h, 1680
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1229
Info, 1229
Tlvresult, 1229
unpack_nas_SLQSNasSysInfoCallback_ind
nas.h, 1680
unpack_nas_SLQSNasTimerCallback_ind
nas.h, 1681
unpack_nas_SLQSNasTimerCallback_ind_t, 1229
t3396_apn, 1230
t3396_plmn_id, 1230
t3396_val, 1230
unpack_nas_SLQSSetBandPreference
nas.h, 1681
unpack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1681
unpack_nas_SLQSSetSysSelectionPref
nas.h, 1681
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
nas.h, 1682
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1230
Info, 1230
Tlvresult, 1230
unpack_nas_SLQSSetSysSelectionPrefExt

nas.h, 1682
unpack_nas_SLQSSwiGetHDRPersonality
nas.h, 1682
unpack_nas_SLQSSwiGetHDRPersonality_t, 1230
pCurrentPersonality, 1231
pPersonalityListLength, 1231
pProtocolSubtypeElement, 1231
unpack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1683
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1231
pAppSubType, 1232
pCurrentPrsnlty, 1232
pPersonalityListLength, 1232
unpack_nas_SLQSSwiGetHRPDStats
nas.h, 1683
unpack_nas_SLQSSwiGetHRPDStats_t, 1232
pPilotSetData, 1233
pUATI, 1233
unpack_nas_SLQSSwiGetLteCQI
nas.h, 1684
unpack_nas_SLQSSwiGetLteCQI_t, 1233
ValidityCW0, 1233
ValidityCW1, 1233
unpack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1684
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1234
pSccRxInfo, 1234
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind
nas.h, 1684
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t, 1234
unpack_nas_SLQSSwiNetworkDebug
nas.h, 1685
unpack_nas_SLQSSwiNetworkDebug_t, 1234
pDataStatusDetail, 1235
pDeviceConfigDetail, 1235
pNetworkStat1x, 1235
pNetworkStatEVDO, 1235
pObjectVer, 1235
unpack_nas_SLQSSwiPSDetach
nas.h, 1685
unpack_nas_SLQSSwiRandIndicatorCallback_Ind
nas.h, 1685
unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t, 1235
rankIndicatorTlv, 1236
unpack_nas_SLQSSysInfoCallback_ind_t, 1236
pGSMSysInfo, 1238
pHDRSysInfo, 1238
pLTESysInfo, 1238
pSysInfoNoChange, 1239
unpack_nas_SetACCOLC
nas.h, 1667
unpack_nas_SetCDMANetworkParameters
nas.h, 1667
unpack_nas_SetDataCapabilitiesCallback_ind
nas.h, 1668
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1196
dataCaps, 1196
dataCapsSize, 1196
unpack_nas_SetEventReportInd
nas.h, 1668
unpack_nas_SetEventReportInd_t, 1196
RFTlv, 1197
RRTlv, 1197
SLQSSSTlv, 1197
SSTlv, 1197
unpack_nas_SetLURejectCallback
nas.h, 1668
unpack_nas_SetNasLTECphyCaIndCallback_ind
nas.h, 1669
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1197
unpack_nas_SetNetworkPreference
nas.h, 1669
unpack_nas_SetNetworkPreference_t, 1198
Tlvresult, 1198
unpack_nas_SetRFInfoCallback
nas.h, 1669
unpack_nas_SetRoamingIndicatorCallback_ind
nas.h, 1670
unpack_nas_SetRoamingIndicatorCallback_ind_t, 1198
roaming, 1198
unpack_nas_SetServingSystemCallback_ind
nas.h, 1670
unpack_nas_SetServingSystemCallback_ind_t, 1199
SSInfo, 1199
Tlvresult, 1199
unpack_nas_SlqsGetLTECphyCAInfo
nas.h, 1671
unpack_nas_SlqsGetLTECphyCAInfo_t, 1200
LTECphyCaInfo, 1201
Tlvresult, 1201
unpack_omaDmConfigTlv_t, 1239
alertmsg, 1239
alertmsglength, 1239
state, 1239
userInputReq, 1240
userInputTimeout, 1240
unpack_omaDmFotaTlv_t, 1240
description, 1241
descriptionlength, 1241
fwdloadsize, 1241
fwloadComplete, 1241
namelength, 1241
package_name, 1241
sessionType, 1241
severity, 1241
state, 1241
updateCompleteStatus, 1241
userInputReq, 1241
userInputTimeout, 1242
version, 1242
versionlength, 1242
unpack_omaDmNotificationsTlv_t, 1242
notification, 1242

sessionStatus, 1242
unpack_pds_ForceXTRADownload
pds.h, 1699
unpack_pds_GetPDSDefaults
pds.h, 1700
unpack_pds_GetPDSDefaults_t, 1243
pAccuracy, 1243
pInterval, 1243
pOperation, 1243
pTimeout, 1243
unpack_pds_GetPDSState
pds.h, 1700
unpack_pds_GetPDSState_t, 1243
pEnabledStatus, 1244
pTrackingStatus, 1244
unpack_pds_GetPortAutomaticTracking
pds.h, 1700
unpack_pds_GetPortAutomaticTracking_t, 1244
pbAuto, 1244
unpack_pds_GetServiceAutomaticTracking
pds.h, 1701
unpack_pds_GetServiceAutomaticTracking_t, 1245
pbAuto, 1245
unpack_pds_GetXTRAAutomaticDownload
pds.h, 1701
unpack_pds_GetXTRAAutomaticDownload_t, 1245
pInterval, 1245
pbEnabled, 1245
unpack_pds_GetXTRANetwork
pds.h, 1701
unpack_pds_GetXTRANetwork_t, 1246
pPreference, 1246
unpack_pds_GetXTRAValidity
pds.h, 1702
unpack_pds_GetXTRAValidity_t, 1246
pDuration, 1247
pGPSWeek, 1247
pGPSWeekOffset, 1247
unpack_pds_PDSInjectTimeReference
pds.h, 1702
unpack_pds_ResetPDSData
pds.h, 1702
unpack_pds_SLQSGetAGPSConfig
pds.h, 1706
unpack_pds_SLQSGetAGPSConfig_t, 1249
pServerAddress, 1250
pServerPort, 1250
pServerURL, 1250
unpack_pds_SLQSGetGPSStateInfo
pds.h, 1706
unpack_pds_SLQSGetGPSStateInfo_t, 1250
Altitude, 1253
EngineState, 1253
HorizontalUncertainty, 1253
Iono_valid, 1253
Latitude, 1253
Longitude, 1253
Time_uncert_ms, 1254
ValidMask, 1254
VerticalUncertainty, 1254
unpack_pds_SLQSPDSDeterminePosition
pds.h, 1706
unpack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1707
unpack_pds_SLQSPDSInjectPositionData
pds.h, 1707
unpack_pds_SLQSSetAGPSConfig
pds.h, 1707
unpack_pds_SLQSSetPositionMethodState
pds.h, 1708
unpack_pds_SetEventReport_Ind
pds.h, 1703
unpack_pds_SetEventReport_Ind_t, 1247
dLatitude, 1248
dLongitude, 1248
has_PositionDataNMEA, 1248
has_SessionStatus, 1248
has_dLatitude, 1248
has_dLongitude, 1248
has_posSrc, 1248
posSrc, 1248
PositionDataNMEA, 1248
SessionStatus, 1248
unpack_pds_SetEventReportCallback
pds.h, 1703
unpack_pds_SetPDSDefaults
pds.h, 1703
unpack_pds_SetPDSState
pds.h, 1704
unpack_pds_SetPdsState_Ind
pds.h, 1704
unpack_pds_SetPdsState_Ind_t, 1249
EnabledStatus, 1249
TlvPresent, 1249
TrackingStatus, 1249
unpack_pds_SetPortAutomaticTracking
pds.h, 1704
unpack_pds_SetServiceAutomaticTracking
pds.h, 1705
unpack_pds_SetXTRAAutomaticDownload
pds.h, 1705
unpack_pds_SetXTRANetwork
pds.h, 1705
unpack_pds_StartPDSTrackingSessionExt
pds.h, 1708
unpack_pds_StopPDSTrackingSession
pds.h, 1708
unpack_qmi_t, 1254
msgid, 1254
type, 1254
xid, 1254
unpack_qos_BindDataPort
qos.h, 2155
unpack_qos_IPv4Addr_t, 1255
addr, 1255
subnetMask, 1255

unpack_qos_IPv6Addr_t, 1255
addr, 1256
prefixLen, 1256
unpack_qos_IPv6TrafCls_t, 1256
mask, 1256
val, 1256
unpack_qos_Port_t, 1257
port, 1257
range, 1257
unpack_qos_QosFlowInfo_t, 1257
BearerID, 1258
is_RxQFlowGranted_Available, 1258
is_TxQFlowGranted_Available, 1258
NumRxFilters, 1258
NumTxFilters, 1258
QFlowState, 1258
RxQFilter, 1258
RxQFlowGranted, 1259
TxQFilter, 1259
TxQFlowGranted, 1259
unpack_qos_QosFlowInfoState_t, 1259
id, 1259
isNewFlow, 1259
state, 1259
unpack_qos_SLQSQosGetNetworkStatus
qos.h, 2155
unpack_qos_SLQSQosGetNetworkStatus_t, 1259
NWQoSStatus, 1260
unpack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 2156
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1260
ambr_dl, 1261
ambr_dl_ext, 1261
ambr_dl_ext2, 1261
ambr_ul, 1261
ambr_ul_ext, 1261
ambr_ul_ext2, 1261
apnId, 1261
unpack_qos_SLQSQosSwiReadDataStats
qos.h, 2156
unpack_qos_SLQSQosSwiReadDataStats_t, 1261
apnId, 1262
numQosFlow, 1262
qosFlow, 1262
total_rx_bytes, 1262
total_rx_pkt, 1262
total_tx_bytes, 1262
total_tx_pkt, 1262
unpack_qos_SLQSSetQosEventCallback
qos.h, 2157
unpack_qos_SLQSSetQosEventCallback_ind
qos.h, 2157
unpack_qos_SLQSSetQosEventCallback_ind_t, 1262
NumFlows, 1263
QosFlowInfo, 1263
unpack_qos_SLQSSetQosNWStatusCallback_ind
qos.h, 2158
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1263
status, 1263
unpack_qos_SLQSSetQosPriEventCallback_ind
qos.h, 2158
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 1263
event, 1264
unpack_qos_SLQSSetQosStatusCallback_ind
qos.h, 2159
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1264
event, 1265
id, 1265
reason, 1265
status, 1265
unpack_qos_Tos_t, 1273
mask, 1274
val, 1274
unpack_qos_dataRate_t, 1254
dataRateMax, 1255
guaranteedRate, 1255
unpack_qos_pktErrRate_t, 1256
exponent, 1257
multiplier, 1257
unpack_qos_swiQosFilter_t, 1265
EspSpi, 1267
IPv4DstAddr, 1267
IPv4SrcAddr, 1267
IPv4Tos, 1267
IPv6DstAddr, 1267
IPv6Label, 1268
IPv6SrcAddr, 1268
IPv6TrafCls, 1268
Id, 1267
index, 1267
is_EspSpi_Available, 1268
is_IPv4DstAddr_Available, 1268
is_IPv4SrcAddr_Available, 1268
is_IPv4Tos_Available, 1268
is_IPv6DstAddr_Available, 1268
is_IPv6Label_Available, 1268
is_IPv6SrcAddr_Available, 1268
is_IPv6TrafCls_Available, 1268
is_Id_Available, 1268
is_NxtHdrProto_Available, 1268
is_Precedence_Available, 1268
is_TCPDstPort_Available, 1268
is_TCPSrcPort_Available, 1268
is_TranDstPort_Available, 1268
is_TranSrcPort_Available, 1268
is_UDPDstPort_Available, 1268
is_UDPSrcPort_Available, 1268
NxtHdrProto, 1268
Precedence, 1268
TCPDstPort, 1268
TCPSrcPort, 1268
TranDstPort, 1268
TranSrcPort, 1268

UDPDstPort, 1268
UDPSrcPort, 1268
version, 1269
unpack_qos_swiQosFlow_t, 1269
DataRate, 1271
index, 1271
is_DataRate_Available, 1271
is_Jitter_Available, 1272
is_Latency_Available, 1272
is_LteQci_Available, 1272
is_MaxAllowedPktSz_Available, 1272
is_MinPolicedPktSz_Available, 1272
is_PktErrRate_Available, 1272
is_ProfileId3GPP2_Available, 1272
is_TokenBucket_Available, 1272
is_TrafficClass_Available, 1272
is_val_3GPP2Pri_Available, 1272
is_val_3GPPImCn_Available, 1272
is_val_3GPPSigInd_Available, 1272
Jitter, 1272
Latency, 1272
LteQci, 1272
MaxAllowedPktSz, 1272
MinPolicedPktSz, 1272
PktErrRate, 1272
ProfileId3GPP2, 1272
TokenBucket, 1272
TrafficClass, 1272
val_3GPP2Pri, 1272
val_3GPPImCn, 1272
val_3GPPResResidualBER, 1272
val_3GPPSigInd, 1272
val_3GPPTraHdlPri, 1272
unpack_qos_tokenBucket_t, 1273
bucketSz, 1273
peakRate, 1273
tokenRate, 1273
unpack_result_code_only
common.h, 1552
unpack_rms_GetSMSWake
rms.h, 2161
unpack_rms_GetSMSWake_t, 1274
enabled, 1275
wake_mask, 1275
unpack_rms_SetSMSWake
rms.h, 2161
unpack_sar_SLQSGetRfSarState
sar.h, 2163
unpack_sar_SLQSGetRfSarState_t, 1276
pSarRFState, 1277
unpack_sar_SLQSSetRfSarState
sar.h, 2163
unpack_sms_GetSMSCAddress
sms.h, 2176
unpack_sms_GetSMSCAddress_t, 1277
addressSize, 1277
pSMSCAddress, 1277
pSMSCType, 1277
typeSize, 1277
unpack_sms_SLQSDeleteSMS
sms.h, 2178
unpack_sms_SLQSDeleteSMS_t, 1280
unpack_sms_SLQSGetIndicationRegister
sms.h, 2178
unpack_sms_SLQSGetIndicationRegister_t, 1280
pGetIndicationRegInfo, 1280
unpack_sms_SLQSGetMessageWaiting
sms.h, 2178
unpack_sms_SLQSGetMessageWaiting_t, 1280
pGetMsgWaitingInfoResp, 1281
unpack_sms_SLQSGetSMS
sms.h, 2179
unpack_sms_SLQSGetSMS_t, 1281
message, 1281
messageFormat, 1281
messageSize, 1281
messageTag, 1281
unpack_sms_SLQSGetSMSList
sms.h, 2179
unpack_sms_SLQSGetSMSList_t, 1282
messageList, 1283
messageListSize, 1283
unpack_sms_SLQSGetSmsBroadcastConfig
sms.h, 2179
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1282
pBroadcastConfig, 1282
pCDMABroadcastConfig, 1282
unpack_sms_SLQSGetTransLayerInfo
sms.h, 2180
unpack_sms_SLQSGetTransLayerInfo_t, 1283
pGetTransLayerInfo, 1283
unpack_sms_SLQSGetTransNWRegInfo
sms.h, 2180
unpack_sms_SLQSGetTransNWRegInfo_t, 1283
unpack_sms_SLQSModifySMSStatus
sms.h, 2180
unpack_sms_SLQSModifySMSStatus_t, 1283
unpack_sms_SLQSNWRegInfoCallback_ind
sms.h, 2181
unpack_sms_SLQSNWRegInfoCallback_ind_t, 1284
NWRegStat, 1284
unpack_sms_SLQSSendAsyncSMS
sms.h, 2181
unpack_sms_SLQSSetIndicationRegister
sms.h, 2181
unpack_sms_SLQSSetSmsBroadcastActivation
sms.h, 2182
unpack_sms_SLQSSetSmsBroadcastConfig
sms.h, 2182
unpack_sms_SLQSSetSmsStorage
sms.h, 2182
unpack_sms_SLQSSmsGetMaxStorageSize
sms.h, 2183
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1284
pMaxStorageSizeResp, 1285
unpack_sms_SLQSSmsGetMessageProtocol

sms.h, 2183
unpack_sms_SLQSSmsGetMessageProtocol_t, 1285
pMessageProtocol, 1285
unpack_sms_SLQSSmsSetRoutes
sms.h, 2183
unpack_sms_SLQSSwiGetSMSStorage
sms.h, 2184
unpack_sms_SLQSSwiGetSMSStorage_t, 1285
pSmsStorage, 1286
unpack_sms_SLQSTransLayerInfoCallback_ind
sms.h, 2184
unpack_sms_SLQSTransLayerInfoCallback_ind_t, 1286
pTransLayerInfo, 1286
regInd, 1286
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind
sms.h, 2184
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 1286
causeCode, 1288
errorClass, 1288
messageID, 1288
RPCause, 1288
sendStatus, 1288
TPCause, 1288
userData, 1288
unpack_sms_SLQSWmsMemoryFullCallBack_ind
sms.h, 2185
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1288
messageMode, 1288
storageType, 1288
unpack_sms_SLQSWmsMessageWaitingCallBack_ind
sms.h, 2185
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t, 1289
msgWaitInfo, 1289
numInstances, 1289
unpack_sms_SaveSMS
sms.h, 2176
unpack_sms_SaveSMS_t, 1277
pMessageIndex, 1278
unpack_sms_SendSMS
sms.h, 2176
unpack_sms_SendSMS_t, 1278
messageFailureCode, 1278
messageID, 1278
unpack_sms_SetNewSMSCallback
sms.h, 2177
unpack_sms_SetNewSMSCallback_ind
sms.h, 2177
unpack_sms_SetNewSMSCallback_ind_t, 1278
ETWSTlv, 1279
IMSTlv, 1279
MMTlv, 1279
NewMMTlv, 1279
SMSCTlv, 1279
TRMessageTlv, 1279
unpack_sms_SetNewSMSCallback_t, 1279
unpack_sms_SetSMSCAddress
sms.h, 2177
unpack_swidms_SLQSSwiDmsGetMTU
swidms.h, 2189
unpack_swidms_SLQSSwiDmsGetMTU_t, 1289
pMTUSize3gpp, 1290
unpack_swidms_SLQSSwiDmsGetUsbNetNum
swidms.h, 2189
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 1290
usbNetNum, 1290
unpack_swidms_SLQSSwiDmsSetMTU
swidms.h, 2189
unpack_swidms_SLQSSwiDmsSetMTU_t, 1290
Tlvresult, 1291
unpack_swidms_SLQSSwiDmsSetUsbNetNum
swidms.h, 2190
unpack_swiloc_SwiLocGetAutoStart
swiloc.h, 2191
unpack_swiloc_SwiLocGetAutoStart_t, 1291
fix_rate, 1292
fix_rate_reported, 1292
fix_type, 1292
fix_type_reported, 1292
function, 1292
function_reported, 1292
max_dist, 1292
max_dist_reported, 1292
max_time, 1292
max_time_reported, 1292
unpack_swiloc_SwiLocSetAutoStart
swiloc.h, 2192
unpack_swioma_SLQSOMADMAlertCallback
swioma.h, 2197
unpack_swioma_SLQSOMADMAlertCallback_ind
swioma.h, 2198
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1292
eventType, 1293
unpack_swioma_SLQSOMADMCancelSession
swioma.h, 2198
unpack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 2199
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1293
Date, 1295
DateLength, 1295
PkgDescLength, 1295
PkgDescription, 1295
PkgName, 1295
PkgNameLength, 1295
RetryCount, 1295
SessionState, 1295
SessionType, 1295
Severity, 1295
Source, 1295
SourceLength, 1295
Status, 1296
Time, 1296
TimeLength, 1296

UpdateCompleteStatus, 1296
unpack_swioma_SLQSOMADMGetSettings
swioma.h, 2199
unpack_swioma_SLQSOMADMGetSettings_t, 1296
Autosdm, 1297
FOTAdownload, 1297
FwAutoCheck, 1297
unpack_swioma_SLQSOMADMSendSelection
swioma.h, 2199
unpack_swioma_SLQSOMADMSetSettings
swioma.h, 2200
unpack_swioma_SLQSOMADMStartSession
swioma.h, 2200
unpack_swioma_SLQSOMADMStartSession_t, 1297
FwAvailability, 1298
unpack_uim_ChangePin
uim.h, 2211
unpack_uim_ChangePin_t, 1298
pEncryptedPIN1, 1298
pIndicationToken, 1298
pRemainingRetries, 1299
Tlvresult, 1299
unpack_uim_GetCardStatus
uim.h, 2212
unpack_uim_GetCardStatus_t, 1299
pCardStatus, 1299
pHotSwapStatus, 1299
Tlvresult, 1299
unpack_uim_ReadTransparent
uim.h, 2212
unpack_uim_ReadTransparent_t, 1299
pCardResult, 1300
pEncryptedData, 1300
pIndicationToken, 1300
pReadResult, 1300
Tlvresult, 1300
unpack_uim_SLQSUIMAuthenticate
uim.h, 2213
unpack_uim_SLQSUIMAuthenticate_t, 1302
pAuthenticateResult, 1302
pCardResult, 1302
pIndicationToken, 1302
unpack_uim_SLQSUIMDepersonalization
uim.h, 2214
unpack_uim_SLQSUIMDepersonalization_t, 1302
pRemainingRetries, 1303
unpack_uim_SLQSUIMEventRegister
uim.h, 2214
unpack_uim_SLQSUIMEventRegister_t, 1303
eventMask, 1303
unpack_uim_SLQSUIMGetConfiguration
uim.h, 2214
unpack_uim_SLQSUIMGetConfiguration_t, 1303
pAutoSelection, 1304
pHaltSubscription, 1304
pPersonalizationStatus, 1304
unpack_uim_SLQSUIMGetFileAttributes
uim.h, 2215
unpack_uim_SLQSUIMGetFileAttributes_t, 1304
pCardResult, 1305
pFileAttributes, 1305
pIndicationToken, 1305
unpack_uim_SLQSUIMGetSlotsStatus
uim.h, 2215
unpack_uim_SLQSUIMGetSlotsStatus_t, 1305
pNumberOfPhySlot, 1305
pUimSlotsStatus, 1305
unpack_uim_SLQSUIMPowerDown
uim.h, 2215
unpack_uim_SLQSUIMPowerUp
uim.h, 2216
unpack_uim_SLQSUIMRefreshCallback_Ind
uim.h, 2216
unpack_uim_SLQSUIMRefreshCallback_Ind_t, 1305
refreshEvent, 1306
TlvPresent, 1306
unpack_uim_SLQSUIMRefreshComplete
uim.h, 2216
unpack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 2217
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 1306
pRefreshEvent, 1306
unpack_uim_SLQSUIMRefreshOK
uim.h, 2217
unpack_uim_SLQSUIMRefreshRegister
uim.h, 2217
unpack_uim_SLQSUIMReset
uim.h, 2218
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind
uim.h, 2218
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind-
_t, 1306
unpack_uim_SLQSUIMSwitchSlot
uim.h, 2218
unpack_uim_SetPinProtection
uim.h, 2212
unpack_uim_SetPinProtection_t, 1300
pEncryptedPIN1, 1301
pIndicationToken, 1301
pRemainingRetries, 1301
Tlvresult, 1301
unpack_uim_SetUimSlotStatusChangeCallback_ind
uim.h, 2213
unpack_uim_SetUimSlotStatusChangeCallback_ind_t,
1301
bNumberOfPhySlots, 1301
slotsstatusChange, 1302
unpack_uim_UnblockPin
uim.h, 2219
unpack_uim_UnblockPin_t, 1307
pEncryptedPIN1, 1307
pIndicationToken, 1307
pRemainingRetries, 1307
Tlvresult, 1307
unpack_uim_VerifyPin
uim.h, 2219

unpack_uim_VerifyPin_t, 1307
pEncryptedPIN1, 1308
pIndicationToken, 1308
pRemainingRetries, 1308
TlvResult, 1308
unpack_valid_nas_GetCDMANetworkParameters
nas.h, 1686
unpack_valid_nas_SLQSGetServingSystem
nas.h, 1686
unpack_valid_nas_SLQSGetSignalStrength
nas.h, 1687
unpack_valid_nas_SLQSNasGetSigInfo
nas.h, 1688
unpack_voice_AnswerUSSD
voice.h, 2237
unpack_voice_CancelUSSD
voice.h, 2237
unpack_voice_DTMFEventCallback_ind
voice.h, 2238
unpack_voice_DTMFEventCallback_ind_t, 1311
DTMFInformation, 1312
pOffLength, 1312
pOnLength, 1312
unpack_voice_OTASPStatusCallback_ind
voice.h, 2238
unpack_voice_OTASPStatusCallback_ind_t, 1312
callID, 1313
OTASPStatus, 1313
unpack_voice_OriginateUSSD
voice.h, 2238
unpack_voice_SLQSOriginateUSSD
voice.h, 2239
unpack_voice_SLQSOriginateUSSD_t, 1313
pAlphaIDInfo, 1316
pCCSuppsType, 1316
pCallId, 1316
pCcResultType, 1316
pUSSDInfo, 1316
pfailureCause, 1316
unpack_voice_SLQSVoiceALSSelectLine
voice.h, 2239
unpack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 2239
unpack_voice_SLQSVoiceAnswerCall
voice.h, 2240
unpack_voice_SLQSVoiceAnswerCall_t, 1316
pCallId, 1317
unpack_voice_SLQSVoiceBindSubscription
voice.h, 2240
unpack_voice_SLQSVoiceBurstDTMF
voice.h, 2240
unpack_voice_SLQSVoiceBurstDTMF_t, 1317
pCallID, 1317
unpack_voice_SLQSVoiceDialCall
voice.h, 2241
unpack_voice_SLQSVoiceDialCall_t, 1317
pAlphaIDInfo, 1318
pCCResultType, 1318
pCCSUPSType, 1318
pCallID, 1318
unpack_voice_SLQSVoiceEndCall
voice.h, 2241
unpack_voice_SLQSVoiceEndCall_t, 1318
pCallId, 1319
unpack_voice_SLQSVoiceGetAllCallInfo
voice.h, 2241
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1319
pArrAlertingPattern, 1321
pArrAlertingType, 1321
pArrAlphaID, 1321
pArrCallEndReason, 1321
pArrCallInfo, 1321
pArrCalledPartyNum, 1321
pArrConnectPartyNum, 1321
pArrDiagInfo, 1321
pArrRedirPartyNum, 1321
pArrRemotePartyName, 1321
pArrRemotePartyNum, 1321
pArrSvcOption, 1321
pArrUUSInfo, 1321
pVoicePrivacy, 1321
unpack_voice_SLQSVoiceGetCLIP
voice.h, 2243
unpack_voice_SLQSVoiceGetCLIP_t, 1329
pAlphaIDInfo, 1330
pCCResType, 1330
pCallID, 1330
pFailCause, 1330
unpack_voice_SLQSVoiceGetCLIR
voice.h, 2244
unpack_voice_SLQSVoiceGetCLIR_t, 1330
pAlphaIDInfo, 1331
pCCResType, 1331
pCallID, 1331
pFailCause, 1332
unpack_voice_SLQSVoiceGetCNAP
voice.h, 2244
unpack_voice_SLQSVoiceGetCNAP_t, 1332
pAlphaIDInfo, 1333
pCCResType, 1333
pCallID, 1333
pFailCause, 1333
unpack_voice_SLQSVoiceGetCOLP
voice.h, 2244
unpack_voice_SLQSVoiceGetCOLP_t, 1333
pAlphaIDInfo, 1334
pCCResType, 1334
pCallID, 1334
pFailCause, 1334
unpack_voice_SLQSVoiceGetCOLR
voice.h, 2245
unpack_voice_SLQSVoiceGetCOLR_t, 1334
pAlphaIDInfo, 1336
pCCResType, 1336
pCallID, 1336
pFailCause, 1336

unpack_voice_SLQSVoiceGetCallBarring
voice.h, 2242
unpack_voice_SLQSVoiceGetCallBarring_t, 1321
pAlphaIDInfo, 1323
pCCResType, 1323
pCallID, 1323
pFailCause, 1323
pSvcClass, 1323
unpack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 2242
unpack_voice_SLQSVoiceGetCallForwardingStatus_t, 1323
pAlphaIDInfo, 1324
pCCResType, 1324
pCallID, 1324
pFailCause, 1324
unpack_voice_SLQSVoiceGetCallInfo
voice.h, 2243
unpack_voice_SLQSVoiceGetCallInfo_t, 1324
pAlertType, 1327
pAlertingPattern, 1327
pAlphaIDInfo, 1327
pCallInfo, 1327
pConnectNumInfo, 1327
pDiagInfo, 1327
pOTASPStatus, 1327
pRemotePartyName, 1327
pRemotePartyNum, 1327
pSrvOpt, 1327
pUUSInfo, 1327
pVoicePrivacy, 1327
unpack_voice_SLQSVoiceGetCallWaiting
voice.h, 2243
unpack_voice_SLQSVoiceGetCallWaiting_t, 1328
pAlphaIDInfo, 1329
pCCResType, 1329
pCallID, 1329
pFailCause, 1329
pSvcClass, 1329
unpack_voice_SLQSVoiceGetConfig
voice.h, 2245
unpack_voice_SLQSVoiceGetConfig_t, 1336
pAirTimerCnt, 1338
pAutoAnswerStat, 1338
pCurAMRConfig, 1338
pCurPrefVoiceSO, 1338
pCurVoiceDomainPref, 1338
pCurVoicePrivacyPref, 1338
pCurrTTYMode, 1338
pRoamTimerCnt, 1338
unpack_voice_SLQSVoiceIndicationRegister
voice.h, 2245
unpack_voice_SLQSVoiceManageCalls
voice.h, 2246
unpack_voice_SLQSVoiceManageCalls_t, 1338
pFailCause, 1338
unpack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 2246
unpack_voice_SLQSVoiceSUPSCallback_ind
voice.h, 2249
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 1344
pDataSrc, 1346
pFailCause, 1346
pReason, 1346
pSvcClass, 1346
unpack_voice_SLQSVoiceSendFlash
voice.h, 2246
unpack_voice_SLQSVoiceSendFlash_t, 1338
pCallID, 1339
unpack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 2247
unpack_voice_SLQSVoiceSetCallBarringPassword_t, 1339
pAlphaIDInfo, 1340
pCCResType, 1340
pCallID, 1340
pFailCause, 1340
unpack_voice_SLQSVoiceSetConfig
voice.h, 2247
unpack_voice_SLQSVoiceSetConfig_t, 1340
pAirTimerStatus, 1341
pAutoAnsStatus, 1341
pPrefVoiceSOStatus, 1341
pRoamTimerStatus, 1342
pTTYConfigStatus, 1342
pVoiceDomainPrefStatus, 1342
unpack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 2247
unpack_voice_SLQSVoiceSetSUPSService
voice.h, 2248
unpack_voice_SLQSVoiceSetSUPSService_t, 1342
pCallID, 1343
pFailCause, 1343
unpack_voice_SLQSVoiceStartContDTMF
voice.h, 2248
unpack_voice_SLQSVoiceStartContDTMF_t, 1343
pCallID, 1343
unpack_voice_SLQSVoiceStopContDTMF
voice.h, 2248
unpack_voice_SLQSVoiceStopContDTMF_t, 1343
callID, 1344
unpack_voice_SUPSNotificationCallback_ind
voice.h, 2249
unpack_voice_SUPSNotificationCallback_ind_t, 1346
callID, 1348
notifType, 1348
pCUGIndex, 1348
pECTNum, 1348
unpack_voice_USSDNotificationCallback_ind
voice.h, 2249
unpack_voice_USSDNotificationCallback_ind_t, 1348
notification_Type, 1348
unpack_voice_VoiceInfoRecCallback_ind
voice.h, 2250
unpack_voice_VoiceInfoRecCallback_ind_t, 1348
callID, 1350

pCLIRCause, 1350
pCallWaitInd, 1350
pCalledPartyInfo, 1350
pCallerIDInfo, 1350
pCallerNameInfo, 1350
pCallingPartyInfo, 1350
pConnectNumInfo, 1350
pDispInfo, 1350
pExtDispInfo, 1350
pExtDispRecInfo, 1350
pLineCtrlInfo, 1350
pNSSAudioCtrl, 1350
pNSSRelease, 1351
pRedirNumInfo, 1351
pSignalInfo, 1351
unpack_voice_allCallStatusCallback_ind
voice.h, 2237
unpack_voice_allCallStatusCallback_ind_t, 1308
arrCallInfomation, 1311
pArrAlertingPattern, 1311
pArrAlertingType, 1311
pArrAlphaID, 1311
pArrCallEndReason, 1311
pArrCalledPartyNum, 1311
pArrConnectPartyNum, 1311
pArrDiagInfo, 1311
pArrRedirPartyNum, 1311
pArrRemotePartyName, 1311
pArrRemotePartyNum, 1311
pArrSvcOption, 1311
unpack_voice_voicePrivacyChangeCallback_ind
voice.h, 2250
unpack_voice_voicePrivacyChangeCallback_ind_t, 1351
callID, 1351
voicePrivacy, 1351
unpack_wds_DHCPv4ClientLease_ind
wds.h, 2277
unpack_wds_DHCPv4ClientLease_ind_t, 1351
DHCPv4LeaseOptTlv, 1352
DHCPv4LeaseStateTlv, 1352
IPv4AddrTlv, 1352
ProfileIdTlv, 1352
unpack_wds_DHCPv4ClientLeaseChange
wds.h, 2277
unpack_wds_GetAutoconnect
wds.h, 2278
unpack_wds_GetAutoconnect_t, 1352
psetting, 1352
unpack_wds_GetByteTotals
wds.h, 2278
unpack_wds_GetByteTotals_t, 1352
pRXTotalBytes, 1353
pTXTotalBytes, 1353
unpack_wds_GetConnectionRate
wds.h, 2278
unpack_wds_GetConnectionRate_t, 1353
currentChannelRXRate, 1353
currentChannelTXRate, 1353
maxChannelRXRate, 1353
maxChannelTXRate, 1353
unpack_wds_GetDataBearerTechnology
wds.h, 2279
unpack_wds_GetDataBearerTechnology_t, 1354
pDataBearer, 1354
unpack_wds_GetDefaultProfile
wds.h, 2279
unpack_wds_GetDefaultProfile_t, 1354
apnname, 1356
apnsize, 1356
auth, 1356
ipaddr, 1356
ipaddrv6, 1356
name, 1356
namesize, 1356
pdptype, 1357
pridns, 1357
pridnsv6, 1357
secdns, 1357
secdnsv6, 1357
username, 1357
usersize, 1357
unpack_wds_GetDefaultProfileNum
wds.h, 2279
unpack_wds_GetDefaultProfileNum_t, 1357
index, 1357
unpack_wds_GetDormancyState
wds.h, 2280
unpack_wds_GetDormancyState_t, 1357
dormancyState, 1358
unpack_wds_GetLastMobileIPError
wds.h, 2280
unpack_wds_GetLastMobileIPError_t, 1358
error, 1358
unpack_wds_GetMobileIP
wds.h, 2280
unpack_wds_GetMobileIP_t, 1358
mipMode, 1358
unpack_wds_GetMobileIPProfile
wds.h, 2281
unpack_wds_GetMobileIPProfile_t, 1359
AAASPI, 1360
AAAState, 1360
address, 1360
enabled, 1360
HASPI, 1360
HAState, 1360
NAI, 1360
naiSize, 1360
primaryHA, 1360
revTunneling, 1360
secondaryHA, 1360
unpack_wds_GetPacketStatistics
wds.h, 2281
unpack_wds_GetPacketStatistics_t, 1360
pRXDroppedCount, 1362

pRXOKBytesLastCall, 1362
pRXOkBytesCount, 1362
pRXPacketErrors, 1362
pRXPacketOverflows, 1362
pRXPacketSuccesses, 1362
pTXDroppedCount, 1362
pTXOKBytesLastCall, 1362
pTXOkBytesCount, 1362
pTXPacketErrors, 1362
pTXPacketOverflows, 1362
pTXPacketSuccesses, 1362
unpack_wds_GetPacketStatus
wds.h, 2281
unpack_wds_GetPacketStatus_t, 1362
rXDroppedCount, 1364
rXOKBytesLastCall, 1364
rXOkBytesCount, 1364
rXPacketErrors, 1364
rXPacketOverflows, 1364
rXPacketSuccesses, 1364
tXDroppedCount, 1364
tXOKBytesLastCall, 1364
tXOkBytesCount, 1364
tXPacketErrors, 1364
tXPacketOverflows, 1364
tXPacketSuccesses, 1364
unpack_wds_GetSessionDuration
wds.h, 2282
unpack_wds_GetSessionDuration_t, 1364
callDuration, 1365
unpack_wds_GetSessionState
wds.h, 2282
unpack_wds_GetSessionState_t, 1365
connectionStatus, 1365
unpack_wds_RMSetTransferStatistics
wds.h, 2282
unpack_wds_RMSetTransferStatistics_t, 1365
unpack_wds_RMTransferStatistics_ind
wds.h, 2283
unpack_wds_SLQSCreateProfile
wds.h, 2285
unpack_wds_SLQSCreateProfile_t, 1366
pCreateProfileOut, 1366
pProfileID, 1366
Tlvresult, 1366
unpack_wds_SLQSDUNCallInfoCallBack_ind
wds.h, 2286
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 1367
unpack_wds_SLQSDeleteProfile
wds.h, 2286
unpack_wds_SLQSDeleteProfile_t, 1366
extendedErrorCode, 1367
unpack_wds_SLQSGet3GPPConfigItem
wds.h, 2286
unpack_wds_SLQSGet3GPPConfigItem_t, 1368
_3gppRelease, 1369
defaultPDNEnabled, 1369
LTEAttachProfile, 1369
profileList, 1370
unpack_wds_SLQSGetCurrDataSystemStat
wds.h, 2287
unpack_wds_SLQSGetCurrDataSystemStat_t, 1370
currNetworkInfo, 1370
networkInfoLen, 1370
prefNetwork, 1370
unpack_wds_SLQSGetCurrentChannelRate
wds.h, 2287
unpack_wds_SLQSGetCurrentChannelRate_t, 1370
max_channel_rx_rate, 1371
max_channel_tx_rate, 1371
unpack_wds_SLQSGetDUNCallInfo
wds.h, 2288
unpack_wds_SLQSGetDUNCallInfo_t, 1372
callEndReason, 1374
channelRate, 1374
connectionStatus, 1374
dataBearerTech, 1374
dormancyStatus, 1375
lastCallDataBearerTech, 1375
mdmCallDurationActive, 1375
rxOKBytesCount, 1375
txOKBytesCount, 1375
unpack_wds_SLQSGetDataBearerTechnology
wds.h, 2287
unpack_wds_SLQSGetDataBearerTechnology_t, 1371
curDataBearerTechnology, 1372
dataBearerMask, 1372
lastCallDataBearerTechnology, 1372
unpack_wds_SLQSGetProfileSettings
wds.h, 2288
unpack_wds_SLQSGetProfileSettings_t, 1375
pProfileSettings, 1375
ProfileType, 1375
Tlvresult, 1375
unpack_wds_SLQSGetRuntimeSettings
wds.h, 2288
unpack_wds_SLQSGetRuntimeSettings_t, 1375
APNName, 1377
Authentication, 1377
DomainList, 1377
GPRSGrantedQoS, 1377
GWAddressV4, 1378
IMCNflag, 1378
IPFamilyPreference, 1378
IPV6AddrInfo, 1378
IPV6GWAddrInfo, 1378
IPv4, 1378
Mtu, 1378
PDPType, 1378
PrimaryDNSV4, 1378
PrimaryDNSV6, 1378
ProfileID, 1378
ProfileName, 1378
SecondaryDNSV4, 1378
SecondaryDNSV6, 1378
ServerAddrList, 1378

SubnetMaskV4, 1378
Technology, 1378
UMTSGrantedQoS, 1378
Username, 1378
unpack_wds_SLQSModifyProfile
wds.h, 2289
unpack_wds_SLQSModifyProfile_t, 1378
pExtErrorCode, 1379
unpack_wds_SLQSResetPacketStatics
wds.h, 2289
unpack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 2291
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1383
pHwConfig, 1383
unpack_wds_SLQSSGetLoopback
wds.h, 2291
unpack_wds_SLQSSGetLoopback_t, 1383
ByteLoopbackMode, 1384
ByteLoopbackMultiplier, 1384
unpack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 2292
unpack_wds_SLQSSSetLoopback
wds.h, 2292
unpack_wds_SLQSSet3GPPConfigItem
wds.h, 2289
unpack_wds_SLQSSetIPFamilyPreference
wds.h, 2289
unpack_wds_SLQSSetIPFamilyPreference_t, 1379
Tlvresult, 1379
unpack_wds_SLQSSetPacketSrvStatusCallback
wds.h, 2290
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1379
bearerID, 1381
conn_status, 1381
ipFamily, 1381
reconfigReqd, 1381
sessionEndReason, 1381
techName, 1381
verboseSessnEndReason, 1381
verboseSessnEndReasonType, 1381
unpack_wds_SLQSSetWdsEventCallback
wds.h, 2290
unpack_wds_SLQSSetWdsEventCallback_ind
wds.h, 2290
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1381
currDBTechAvail, 1382
currNWInfo, 1382
dBTechAvail, 1382
dBTechnology, 1382
dBtechExtAvail, 1382
dBtechnologyExt, 1382
dataSysStatAvail, 1382
dormancyStatAvail, 1382
dormancyStatus, 1382
mipStatus, 1382
mipstatAvail, 1382
netInfoLen, 1383
prefNetwork, 1383
ratMask, 1383
rx_bytes, 1383
rx_pkts, 1383
soMask, 1383
tx_bytes, 1383
tx_pkts, 1383
xferStatAvail, 1383
unpack_wds_SLQSStartDataSession
wds.h, 2292
unpack_wds_SLQSStartDataSession_t, 1384
pFailureReason, 1385
pVerboseFailReasonType, 1385
pVerboseFailureReason, 1385
psid, 1385
unpack_wds_SLQSStopDataSession
wds.h, 2293
unpack_wds_SLQSWdsGoActive
wds.h, 2293
unpack_wds_SLQSWdsGoDormant
wds.h, 2293
unpack_wds_SLQSWdsSetEventReport
wds.h, 2293
unpack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 2294
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1385
apnName, 1387
bearerId, 1387
contextId, 1387
ipv4Address, 1387
ipv4GWAddress, 1387
ipv6Address, 1387
ipv6GWAddress, 1387
unpack_wds_SetAutoconnect
wds.h, 2283
unpack_wds_SetDefaultProfile
wds.h, 2283
unpack_wds_SetDefaultProfileNum
wds.h, 2284
unpack_wds_SetMobileIP
wds.h, 2284
unpack_wds_SetMobileIPParameters
wds.h, 2284
unpack_wds_SetMobileIPProfile
wds.h, 2285
unpack_wds_SetMobileIPProfile_t, 1365
unpack_wds_SetMuxID
wds.h, 2285
UnpackQmiProfileInfo
wds.h, 2257
unpackWdsProfileParam, 1388
SlqsProfile3GPP, 1388
SlqsProfile3GPP2, 1388
upLink
NSSAudioCtrl, 620
voice_NSSAudioCtrl, 1434
UpdateCompleteStatus

unpack_swioma_SLQSOMADMGetSessionInfo_t, 1296
updateCompleteStatus
omaDmFotaTlv, 625
unpack_omaDmFotaTlv_t, 1241
upgrade_9x07_9x50_fw
qaGobiApiFms.h, 1872
upgrade_mc77xx_fw
qaGobiApiFms.h, 1872
UpgradeFirmware2k
qaGobiApiFms.h, 1873
upinRetries
slotInf, 939
slotInfo, 941
uim_slotInfo, 1066
upinState
slotInf, 940
slotInfo, 941
uim_slotInfo, 1066
UpkQmiCbkCatEventReportInd
qaCbkCatEventReportInd.h, 1710
UpkQmiCbkSwiOmaDmEventReportInd
qaCbkSwiOmaDmEventReportInd.h, 1711
UpkQmiCbkSwiOmaDmEventReportIndExt
qaCbkSwiOmaDmEventReportInd.h, 1711
UpkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 2142
UpkQmiNasPerformNetworkScan
qaNasPerformNetworkScan.h, 2143
upukRetries
slotInf, 940
slotInfo, 941
uim_slotInfo, 1066
urlAddr, 1388
address, 1389
loc_URLAddrInfo, 399
TlvPresent, 1389
URLAddrInfo, 1389
usageMask
loc_sensorDataUsage, 396
sensorDataUsage_s, 903
t_sensor, 1023
UsbMTUSize
swidms_usbMTUSizeTlv, 999
usbMTUSizeTlv, 1393
usbMTUSizeTlv, 1392
TlvPresent, 1393
UsbMTUSize, 1393
usbNetNum
SwiDmsSLQSGetUsbNetNum, 1000
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 1290
User Identity Module Service (UIM), 54
userData
SMSAsyncRawSend_s, 972
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 1288
userInputReq
omaDmConfigTlv, 621
omaDmConfigTlvExt, 623
omaDmFotaTlv, 625
unpack_omaDmConfigTlv_t, 1240
unpack_omaDmFotaTlv_t, 1241
userInputTimeout
omaDmConfigTlv, 621
omaDmConfigTlvExt, 623
omaDmFotaTlv, 625
omaDmFotaTlvExt, 627
unpack_omaDmConfigTlv_t, 1240
unpack_omaDmFotaTlv_t, 1242
Username
unpack_wds_SLQSGetRuntimeSettings_t, 1378
username
unpack_wds_GetDefaultProfile_t, 1357
usersize
unpack_wds_GetDefaultProfile_t, 1357
ussDCS
pack_voice_SLQSOriginateUSSD_t, 743
USSInfo, 1394
voice_USSInfo, 1444
ussData
pack_voice_SLQSOriginateUSSD_t, 743
USSInfo, 1394
voice_USSInfo, 1444
ussLen
pack_voice_SLQSOriginateUSSD_t, 743
USSInfo, 1394
voice_USSInfo, 1444
uusInfo
allCallsUUSInfo, 120
voice_allCallsUUSInfo, 1399

v4sessionId
qaQmiInterfaceInfo, 829
ssdatasession_params, 992
WdsRunTimeSettings, 1543
v6sessionId
qaQmiInterfaceInfo, 829
ssdatasession_params, 992
WdsRunTimeSettings, 1543
VOICE_SUPS_SRV_CLASS_DATA
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_FAX
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_NONE
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_PACKETACCESS
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_PADACCESS
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_SMS
qaGobiApiVoice.h, 2081
VOICE_SUPS_SRV_CLASS_VOICE

qaGobiApiVoice.h, 2081
VDOP
loc_precisionDilution, 393
precisionDilution_s, 807
VOICE_SRV
qaGobiApiCbk.h, 1726
val
IPv6TrafCls, 350
Tos, 1037
unpack_qos_IPv6TrafCls_t, 1256
unpack_qos_Tos_t, 1274
val_3GPP2Pri
unpack_qos_swiQosFlow_t, 1272
val_3GPPImCn
unpack_qos_swiQosFlow_t, 1272
val_3GPPResResidualBER
unpack_qos_swiQosFlow_t, 1272
val_3GPPSigInd
unpack_qos_swiQosFlow_t, 1272
val_3GPPTraHdlPri
unpack_qos_swiQosFlow_t, 1272
ValidMask
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
validMask
loc_satelliteInfo, 395
satelliteInfo, 898
ValidateSPC
qaGobiApiDms.h, 1852
ValidityCW0
LteCQIParm, 414
unpack_nas_SLQSSwiGetLteCQI_t, 1233
ValidityCW1
LteCQIParm, 414
unpack_nas_SLQSSwiGetLteCQI_t, 1233
Value
GetM2MSpkrGainResp, 294
SetM2MSpkrGainReq, 925
value
pack_dms_UIMSetPINProtection_t, 645
pack_dms_UIMVerifyPIN_t, 647
value_length
custSettingInfo, 214
DMScustSettingInfo, 239
pack_dms_SetCustFeaturesV2_t, 635
setCustomSettingV2, 915
verbFailReason
ssdatasession_params, 992
verbFailReasonType
ssdatasession_params, 992
verboseSessnEndReason
_packetSrvStatus, 72
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1381
verboseSessnEndReasonType
_packetSrvStatus, 72
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1381
verifyLeft
personalizationStatus, 794
remainingRetries, 875
uim_personalizationStatus, 1058
uim_remainingRetries, 1062
verifyPIN
pack_uim_VerifyPin_t, 742
UIMVerifyPinReq, 1090
verifyRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1151
unpack_dms_UIMSetControlKeyProtection_t, 1154
unpack_dms_UIMSetPINProtection_t, 1155
verifyUIMPIN, 1396
pinID, 1397
pinLen, 1397
pinVal, 1397
version
omaDmFotaTlv, 625
omaDmFotaTlvExt, 627
swiQosFilter, 1010
unpack_omaDmFotaTlv_t, 1242
unpack_qos_swiQosFilter_t, 1269
versionString
pack_dms_SLQSSwiSetOSInfo_t, 642
unpack_dms_SLQSSwiGetOSInfo_t, 1147
versionlength
omaDmFotaTlv, 625
omaDmFotaTlvExt, 627
unpack_omaDmFotaTlv_t, 1242
vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 659
vertReliability
pack_loc_SLQSLOCInjectPosition_t, 659
vertUnc
pack_loc_SLQSLOCInjectPosition_t, 659
VerticalUncertainty
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
VirtStream
nas_protocolSubtypeElement, 521
protocolSubtypeElement, 824
Voice Service (VOICE), 52
voice.h
LITE_VOICE_SUPS_SRV_CLASS_DATA, 2225
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC, 2225
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC, 2225
LITE_VOICE_SUPS_SRV_CLASS_FAX, 2225
LITE_VOICE_SUPS_SRV_CLASS_NONE, 2225
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS, 2225
LITE_VOICE_SUPS_SRV_CLASS_PADACCESS, 2225
LITE_VOICE_SUPS_SRV_CLASS_SMS, 2225
LITE_VOICE_SUPS_SRV_CLASS_VOICE, 2225
voice.h, 2219
liteServiceClassInformation, 2225

MAXVOICEUSSDLENGTH, 2225
pack_voice_AnswerUSSD, 2225
pack_voice_CancelUSSD, 2225
pack_voice_OriginateUSSD, 2226
pack_voice_SLQSOriginateUSSD, 2226
pack_voice_SLQSVoiceALSSelectLine, 2226
pack_voice_SLQSVoiceALSSetLineSwitching, 2227
pack_voice_SLQSVoiceAnswerCall, 2227
pack_voice_SLQSVoiceBindSubscription, 2228
pack_voice_SLQSVoiceBurstDTMF, 2228
pack_voice_SLQSVoiceDialCall, 2228
pack_voice_SLQSVoiceEndCall, 2229
pack_voice_SLQSVoiceGetAllCallInfo, 2229
pack_voice_SLQSVoiceGetCLIP, 2231
pack_voice_SLQSVoiceGetCLIR, 2231
pack_voice_SLQSVoiceGetCNAP, 2232
pack_voice_SLQSVoiceGetCOLP, 2232
pack_voice_SLQSVoiceGetCOLR, 2232
pack_voice_SLQSVoiceGetCallBarring, 2229
pack_voice_SLQSVoiceGetCallForwardingStatus, 2230
pack_voice_SLQSVoiceGetCallInfo, 2230
pack_voice_SLQSVoiceGetCallWaiting, 2231
pack_voice_SLQSVoiceGetConfig, 2233
pack_voice_SLQSVoiceIndicationRegister, 2233
pack_voice_SLQSVoiceManageCalls, 2233
pack_voice_SLQSVoiceOrigUSSDNoWait, 2234
pack_voice_SLQSVoiceSendFlash, 2234
pack_voice_SLQSVoiceSetCallBarringPassword, 2234
pack_voice_SLQSVoiceSetConfig, 2235
pack_voice_SLQSVoiceSetPreferredPrivacy, 2235
pack_voice_SLQSVoiceSetSUPSService, 2236
pack_voice_SLQSVoiceStartContDTMF, 2236
pack_voice_SLQSVoiceStopContDTMF, 2236
unpack_voice_AnswerUSSD, 2237
unpack_voice_CancelUSSD, 2237
unpack_voice_DTMFEventCallback_ind, 2238
unpack_voice_OTASPStatusCallback_ind, 2238
unpack_voice_OriginateUSSD, 2238
unpack_voice_SLQSOriginateUSSD, 2239
unpack_voice_SLQSVoiceALSSelectLine, 2239
unpack_voice_SLQSVoiceALSSetLineSwitching, 2239
unpack_voice_SLQSVoiceAnswerCall, 2240
unpack_voice_SLQSVoiceBindSubscription, 2240
unpack_voice_SLQSVoiceBurstDTMF, 2240
unpack_voice_SLQSVoiceDialCall, 2241
unpack_voice_SLQSVoiceEndCall, 2241
unpack_voice_SLQSVoiceGetAllCallInfo, 2241
unpack_voice_SLQSVoiceGetCLIP, 2243
unpack_voice_SLQSVoiceGetCLIR, 2244
unpack_voice_SLQSVoiceGetCNAP, 2244
unpack_voice_SLQSVoiceGetCOLP, 2244
unpack_voice_SLQSVoiceGetCOLR, 2245
unpack_voice_SLQSVoiceGetCallBarring, 2242
unpack_voice_SLQSVoiceGetCallForwarding-Status, 2242
unpack_voice_SLQSVoiceGetCallInfo, 2243
unpack_voice_SLQSVoiceGetCallWaiting, 2243
unpack_voice_SLQSVoiceGetConfig, 2245
unpack_voice_SLQSVoiceIndicationRegister, 2245
unpack_voice_SLQSVoiceManageCalls, 2246
unpack_voice_SLQSVoiceOrigUSSDNoWait, 2246
unpack_voice_SLQSVoiceSUPSCallback_ind, 2249
unpack_voice_SLQSVoiceSendFlash, 2246
unpack_voice_SLQSVoiceSetCallBarringPassword, 2247
unpack_voice_SLQSVoiceSetConfig, 2247
unpack_voice_SLQSVoiceSetPreferredPrivacy, 2247
unpack_voice_SLQSVoiceSetSUPSService, 2248
unpack_voice_SLQSVoiceStartContDTMF, 2248
unpack_voice_SLQSVoiceStopContDTMF, 2248
unpack_voice_SUPSNotificationCallback_ind, 2249
unpack_voice_USSDNotificationCallback_ind, 2249
unpack_voice_VoiceInfoRecCallback_ind, 2250
unpack_voice_allCallStatusCallback_ind, 2237
unpack_voice_voicePrivacyChangeCallback_ind, 2250
voice_CLIPResp, 1419
ActiveStatus, 1420
ProvisionStatus, 1420
voice_CLIRResp, 1420
ActiveStatus, 1420
ProvisionStatus, 1420
voice_CNAPResp, 1421
ActiveStatus, 1421
ProvisionStatus, 1421
voice_COLPResp, 1421
ActiveStatus, 1422
ProvisionStatus, 1422
voice_COLRResp, 1422
ActiveStatus, 1423
ProvisionStatus, 1423
voice_CUGInfo, 1424
CUGIndex, 1425
SuppOA, 1425
SuppPrefCUG, 1425
voice_DTMFInfo, 1426
callID, 1427
DTMFEvent, 1427
digitBuff, 1427
digitCnt, 1427
voice_DTMFLengths, 1427
DTMFInterdigitInterval, 1428
DTMFPulseWidth, 1428
voice_ECTNum, 1428
ECTCallState, 1429
number, 1429
presentationInd, 1429

voice_NSSAudioCtrl, 1434
downLink, 1434
upLink, 1434
voice_SUPSInfo, 1443
isModByCC, 1443
svcType, 1443
voice_USSDNotificationNetworkInfo, 1443
networkInfo, 1444
tlvPresent, 1444
voice_USSInfo, 1444
ussDCS, 1444
ussData, 1444
ussLen, 1444
voice_UUSInfo, 1444
UUSData, 1445
UUSDatalen, 1445
UUSDcs, 1445
UUSType, 1445
voice_airTimer, 1397
airTimerValue, 1397
namID, 1397
voice_allCallsAlphaIDInfo, 1397
AlphaIDInfo, 1398
callID, 1398
voice_allCallsDiagInfo, 1398
callID, 1398
DiagInfo, 1398
voice_allCallsUUSInfo, 1398
callID, 1399
uusInfo, 1399
voice_alphaIDInfo, 1399
alphaDcs, 1400
alphaLen, 1400
alphaText, 1400
voice_arrAlertingPattern, 1400
alertingPattern, 1400
callID, 1400
numInstances, 1400
voice_arrAlertingType, 1400
AlertingType, 1401
callID, 1401
numInstances, 1401
voice_arrAlphaID, 1401
allCallsAlphaIDInfoArr, 1402
numInstances, 1402
voice_arrCallEndReason, 1402
callEndReason, 1403
callID, 1403
numInstances, 1403
voice_arrCallInfo, 1403
getAllCallInfo, 1403
numInstances, 1403
voice_arrCalledPartyNum, 1402
CalledPartyNum, 1402
numInstances, 1402
voice_arrConnectPartyNum, 1403
ConnectedPartyNum, 1404
numInstances, 1404
voice_arrDiagInfo, 1404
DiagInfo, 1404
numInstances, 1404
voice_arrRedirPartyNum, 1404
numInstances, 1405
RedirPartyNum, 1405
voice_arrRemotePartyName, 1405
GetAllCallRmtPtyName, 1405
numInstances, 1405
voice_arrRemotePartyNum, 1406
numInstances, 1406
RmtPtyNum, 1406
voice_arrSvcOption, 1406
callID, 1407
numInstances, 1407
srvOption, 1407
voice_arrUUSInfo, 1407
AllCallsUUSInfo, 1407
numInstances, 1407
voice_burstDTMFInfo, 1407
digitCnt, 1408
pCallID, 1408
pDigitBuff, 1408
voice_callFWExtInfo, 1412
noReplyTimer, 1414
numLen, 1414
numPlan, 1414
numType, 1414
number, 1414
PI, 1414
SI, 1414
SvcClass, 1414
SvcStatus, 1414
voice_callFWInfo, 1414
noReplyTimer, 1415
numLen, 1415
number, 1415
SvcClass, 1415
SvcStatus, 1415
voice_callFwdTypeAndPlan, 1411
numberPlan, 1412
numberType, 1412
voice_callInfo, 1415
callID, 1416
callState, 1416
callType, 1417
direction, 1417
mode, 1417
voice_calledPartyInfo, 1408
numLen, 1409
numPlan, 1409
numType, 1409
number, 1409
PI, 1410
SI, 1410
voice_calledPartySubAdd, 1410
extBit, 1410
oddEvenInd, 1410

subAddr, 1410
subAddrLen, 1410
subAddrType, 1410
voice_callerIDInfo, 1411
callerID, 1411
callerIDLen, 1411
PI, 1411
voice_callingPartyInfo, 1417
numLen, 1418
numPlan, 1418
numType, 1418
number, 1418
PI, 1418
SI, 1418
voice_ccSUPSType, 1418
reason, 1419
svcType, 1419
voice_connectNumInfo, 1423
callerID, 1424
callerIDLen, 1424
numPlan, 1424
numPresInd, 1424
numType, 1424
screeningInd, 1424
voice_curAMRConfig, 1425
gsmAmrStat, 1426
wcdmaAmrStat, 1426
voice_diagInfo, 1426
diagInfoLen, 1426
diagnosticInfo, 1426
voice_extDispRecInfo, 1429
dispType, 1429
extDispInfo, 1429
extDispInfoLen, 1429
voice_getAllCallInformation, 1429
ALS, 1430
CallInfo, 1430
isEmpty, 1430
voice_getAllCallRmtPtyName, 1430
callID, 1430
RemotePartyName, 1431
voice_getAllCallRmtPtyNum, 1431
callID, 1431
RemotePartyNum, 1431
voice_getCallFWExtInfo, 1431
CallFWExtInfo, 1432
numInstances, 1432
voice_getCallFWInfo, 1432
CallFWInfo, 1432
numInstances, 1432
voice_lineCtrlInfo, 1432
polarityIncluded, 1433
pwrDenialTime, 1433
revPolarity, 1433
toggleMode, 1433
voice_newPwdData, 1433
newPwd, 1433
newPwdAgain, 1433
voice_peerNumberInfo, 1434
callID, 1436
numLen, 1436
numPI, 1436
numPlan, 1436
numSI, 1436
numType, 1436
number, 1436
voice_prefVoiceSO, 1436
evrcCapability, 1438
homeOrigVoiceSO, 1438
homePageVoiceSO, 1438
namID, 1438
roamOrigVoiceSO, 1438
voice_redirNumInfo, 1438
numLen, 1439
numPlan, 1439
numType, 1439
number, 1439
PI, 1439
reason, 1439
SI, 1439
voice_remotePartyName, 1440
callerName, 1440
codingScheme, 1440
nameLen, 1440
namePI, 1440
voice_remotePartyNum, 1440
numLen, 1441
presentationInd, 1441
remPartyNumber, 1441
voice_roamTimer, 1441
namID, 1442
roamTimerValue, 1442
voice_signalInfo, 1442
alertPitch, 1442
signal, 1442
signalType, 1442
voiceALSSelectLineInfo, 1446
lineValue, 1446
voiceALSSetLineSwitchInfo, 1446
switchOption, 1446
voiceAnswerCall, 1446
pCallId, 1447
voiceBindSubscriptionInfo, 1447
subsType, 1447
voiceBurstDTMFInfo, 1447
BurstDTMFInfo, 1448
pBurstDTMFLengths, 1448
voiceCallInfoReq, 1448
callID, 1448
voiceCallInfoResp, 1448
pAlertType, 1450
pAlertingPattern, 1450
pAlphaIDInfo, 1450
pCallInfo, 1450
pConnectNumInfo, 1450
pDiagInfo, 1450

pOTASPStatus, 1450
pRemotePartyName, 1450
pRemotePartyNum, 1450
pSrvOpt, 1451
pUUSInfo, 1451
pVoicePrivacy, 1451
voiceCallRequestParams, 1451
callNumber, 1452
pCLIRType, 1452
pCUGInfo, 1452
pCallPartySubAdd, 1452
pCallType, 1452
pEmergencyCategory, 1452
pSvcType, 1452
pUUSInFo, 1452
voiceCallResponseParams, 1452
pAlphaIDInfo, 1453
pCCResultType, 1453
pCCSUPSType, 1453
pCallID, 1453
voiceContDTMFinfo, 1453
DTMFdigit, 1454
pCallID, 1454
voiceDTMFEventInfo, 1454
DTMFInformation, 1455
pOffLength, 1455
pOnLength, 1455
voiceFlashInfo, 1455
pCallID, 1455
pFlashPayLd, 1455
pFlashType, 1455
voiceGetAllCallInfo, 1455
pArrAlertingPattern, 1458
pArrAlertingType, 1458
pArrAlphaID, 1458
pArrCallEndReason, 1458
pArrCallInfo, 1458
pArrCalledPartyNum, 1458
pArrConnectPartyNum, 1458
pArrDiagInfo, 1458
pArrRedirPartyNum, 1458
pArrRemotePartyName, 1458
pArrRemotePartyNum, 1458
pArrSvcOption, 1458
pArrUUSInfo, 1458
pOTASPStatus, 1458
pVoicePrivacy, 1458
voiceGetCLIPResp, 1464
pAlphaIDInfo, 1465
pCCResType, 1465
pCCSUPSType, 1465
pCLIPResp, 1465
pCallID, 1465
pFailCause, 1465
voiceGetCLIRResp, 1465
pAlphaIDInfo, 1466
pCCResType, 1466
pCCSUPSType, 1466
pCLIRResp, 1466
pCallID, 1466
pFailCause, 1466
voiceGetCNAPResp, 1466
pAlphaIDInfo, 1467
pCCResType, 1467
pCCSUPSType, 1467
pCNAPResp, 1467
pCallID, 1467
pFailCause, 1467
voiceGetCOLPResp, 1467
pAlphaIDInfo, 1468
pCCResType, 1469
pCCSUPSType, 1469
pCOLPResp, 1469
pCallID, 1468
pFailCause, 1469
voiceGetCOLRResp, 1469
pAlphaIDInfo, 1470
pCCResType, 1470
pCCSUPSType, 1470
pCOLRResp, 1470
pCallID, 1470
pFailCause, 1470
voiceGetCallBarringReq, 1458
pSvcClass, 1459
reason, 1459
voiceGetCallBarringResp, 1459
pAlphaIDInfo, 1460
pCCResType, 1460
pCCSUPSType, 1460
pCallID, 1460
pFailCause, 1460
pSvcClass, 1460
voiceGetCallFWReq, 1460
pSvcClass, 1461
Reason, 1461
voiceGetCallFWResp, 1461
pAlphaIDInfo, 1462
pCCResType, 1462
pCCSUPSType, 1462
pCallID, 1462
pFailCause, 1462
pGetCallFWExtInfo, 1462
pGetCallFWInfo, 1462
voiceGetCallWaitInfo, 1462
pAlphaIDInfo, 1463
pCCResType, 1464
pCCSUPSType, 1464
pCallID, 1464
pFailCause, 1464
pSvcClass, 1464
voiceGetConfigReq, 1470
pAMRStatus, 1471
pAirTimer, 1471
pAutoAnswer, 1471
pNamID, 1471
pPrefVoicePrivacy, 1471

pPrefVoiceSO, 1471
pRoamTimer, 1471
pTTYMode, 1472
pVoiceDomainPref, 1472
voiceGetConfigResp, 1472
pAirTimerCnt, 1473
pAutoAnswerStat, 1473
pCurAMRConfig, 1473
pCurPrefVoiceSO, 1473
pCurVoiceDomainPref, 1473
pCurVoicePrivacyPref, 1473
pCurrTTYMode, 1473
pRoamTimerCnt, 1473
voiceIndicationRegisterInfo, 1473
pRegDTMFEvents, 1474
pRegVoicePrivacyEvents, 1474
pSuppsNotifEvents, 1474
voiceInfoRec, 1474
callID, 1476
pCLIRCause, 1476
pCallWaitInd, 1476
pCalledPartyInfo, 1476
pCallerIDInfo, 1476
pCallerNameInfo, 1476
pCallingPartyInfo, 1476
pConnectNumInfo, 1476
pDispInfo, 1476
pExtDispInfo, 1476
pExtDispRecInfo, 1476
pLineCtrlInfo, 1476
pNSSAudioCtrl, 1476
pNSSRelease, 1476
pRedirNumInfo, 1476
pSignalInfo, 1477
voiceManageCallsReq, 1477
pCallID, 1477
SUPSType, 1477
voiceManageCallsResp, 1477
pFailCause, 1478
VoiceNumber
unpack_dms_GetVoiceNumber_t, 1126
voiceNumberSize
unpack_dms_GetVoiceNumber_t, 1126
voiceOTASPStatusInfo, 1478
callID, 1479
OTASPStatus, 1479
voiceOrigUSSDNoWaitInfo, 1478
USSInformation, 1478
voicePrivacy
unpack_voice_voicePrivacyChangeCallback_ind_t, 1351
voicePrivacyInfo, 1480
voicePrivacyInfo, 1479
callID, 1480
voicePrivacy, 1480
voiceSUPSInfo, 1492
pAlphaIDInfo, 1494
pCLIPstatus, 1494
pCLIRstatus, 1494
pCNAPstatus, 1494
pCOLPstatus, 1494
pCOLRstatus, 1494
pCallBarPasswd, 1494
pCallFWNum, 1494
pCallFWTimerVal, 1494
pCallFwdInfo, 1494
pCallID, 1494
pDataSrc, 1494
pFailCause, 1494
pNewPwdData, 1494
pReason, 1494
pSvcClass, 1494
pUSSInfo, 1494
SUPSInformation, 1494
voiceSUPSNotification, 1494
callID, 1496
notifType, 1496
pCUGIndex, 1496
pECTNum, 1496
voiceSetAllCallStatusCbkInfo, 1480
arrCallInfomation, 1482
pArrAlertingPattern, 1482
pArrAlertingType, 1482
pArrAlphaID, 1482
pArrCallEndReason, 1482
pArrCalledPartyNum, 1482
pArrConnectPartyNum, 1482
pArrDiagInfo, 1482
pArrRedirPartyNum, 1482
pArrRemotePartyName, 1482
pArrRemotePartyNum, 1482
pArrSvcOption, 1482
voiceSetCallBarringPwdInfo, 1482
newPasswd, 1483
newPasswdAgain, 1483
oldPasswd, 1483
Reason, 1483
voiceSetCallBarringPwdResp, 1483
pAlphaIDInfo, 1484
pCCResType, 1484
pCCSUPSType, 1484
pCallID, 1484
pFailCause, 1484
voiceSetConfigReq, 1484
pAirTimerConfig, 1485
pAutoAnswer, 1486
pPrefVoiceDomain, 1486
pPrefVoiceSO, 1486
pRoamTimerConfig, 1486
pTTYMode, 1486
voiceSetConfigResp, 1486
pAirTimerStatus, 1487
pAutoAnsStatus, 1487
pPrefVoiceSOStatus, 1487
pRoamTimerStatus, 1487
pTTYConfigStatus, 1487

pVoiceDomainPrefStatus, 1487
voiceSetPrefPrivacy, 1487
privacyPref, 1488
voiceSetSUPSServiceReq, 1488
pCallBarringPasswd, 1490
pCallForwardingNumber, 1490
pCallFwdTypeAndPlan, 1490
pServiceClass, 1490
pTimerVal, 1490
reason, 1490
voiceSvc, 1490
voiceSetSUPSServiceResp, 1490
pAlphaIDInfo, 1491
pCCResultType, 1491
pCCSUPSType, 1491
pCallID, 1491
pFailCause, 1491
voiceStopContDTMFinfo, 1491
callID, 1492
voiceSvc
pack_voice_SLQSVoiceSetSUPSService_t, 759
voiceSetSUPSServiceReq, 1490
VolValue
SetAudioVolTLBConfigReq, 914
VoltStat
dms_VoltageTlv, 238
VoltTlv
unpack_dms_SwiEventReportCallBack_ind_t, 1150
Voltage
dms_VoltageTlv, 238
Volume
GetAudioProfileResp, 274
GetAudioVolTLBConfigReq, 275
GetM2MAudioProfileResp, 291
SetAudioProfileReq, 913
SetAudioVolTLBConfigReq, 914
voteForInit
registerRefresh, 875
uim_registerRefresh, 1062

WAMSParaChanged
_SwiAvmsEventInd, 102
wAlertMsgLength
_QmiSwiAvmsEventReportConfigResp, 76
WCDMACellInfo
lteWcdmaCellInfo, 435
nas_lteWcdmaCellInfo, 500
WCDMAECIOThresh, 1497
pWCDMAECIOThreshList, 1497
WCDMAECIOThreshListLen, 1497
WCDMAECIOThreshListLen
nas_WCDMAECIOThresh, 560
WCDMAECIOThresh, 1497
WCDMAInfoLTENeighborCell, 1497
UMTSLTENbrCell, 1498
umtsLTENbrCellLen, 1498
wcdmaRRCState, 1498
WCDMARSSIThresh, 1502
pWCDMARSSIThreshList, 1503
WCDMARSSIThreshListLen, 1503
WCDMARSSIThreshListLen
nas_WCDMARSSIThresh, 562
WCDMARSSIThresh, 1503
WCDMASSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1225
WCDMASysInfo, 1503
cellId, 1506
cellIdValid, 1506
hsCallStatus, 1506
hsCallStatusValid, 1506
hsInd, 1506
hsIndValid, 1506
lac, 1506
lacValid, 1506
MCC, 1506
MNC, 1506
networkIdValid, 1506
psc, 1506
pscValid, 1506
regRejectInfoValid, 1506
rejCause, 1506
rejectSrvDomain, 1506
sysInfoWCDMA, 1506
WDS_IsGobiDevice
qaGobiApiWds.h, 2141
WDS_SRV
qaGobiApiCbk.h, 1727
WDSGetLoopbackData, 1538
ByteLoopbackMode, 1538
ByteLoopbackMultiplier, 1538
WDSSWICurrentChannelRates, 1546
current_channel_rx_rate, 1546
current_channel_tx_rate, 1546
max_channel_rx_rate, 1546
max_channel_tx_rate, 1546
WDSSetLoopbackData, 1545
pLoopbackMode, 1546
pLoopbackMultiplier, 1546
wDescriptionLength
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 75
wErrorCode
_QmiSwiAvmsEventReportDataSessionStatusResp, 77
wHTTPStatus
_QmiSwiAvmsEventReportHTTPStatusResp, 78
wNameLength
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 75
WORD
SwiDataTypes.h, 2187
wSessionStatus
_QmiSwiAvmsEventReportNotificationResp, 80
wUpdateCompeteStatus
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 75

wUserInputTimeout
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 75
_QmiSwiAvmsEventReportConfigResp, 76
_QmiSwiAvmsEventReportConnectionRequestResp, 77
wVersionLength
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 75
wWamsChangeMask
_QmiSwiAvmsEventReportWAMSParamChangeResp, 81
wake_mask
pack_rms_SetSMSWake_t, 714
unpack_rms_GetSMSWake_t, 1275
wcdmaAmrStat
curAMRConfig, 202
voice_curAMRConfig, 1426
wcdmaCellInfo, 1496
cpich_ecno, 1496
cpich_rscp, 1496
psc, 1496
srxlev, 1497
wcdmaLongMsgDecodingParams, 1498
Date, 1499
pIsUDHPresent, 1499
pMessage, 1500
pPartNum, 1500
pReferenceNum, 1500
pScAddr, 1500
pScAddrLength, 1500
pSenderAddr, 1500
pSenderAddrLength, 1500
pTextMsg, 1500
pTextMsgLength, 1500
pTotalNum, 1500
Time, 1500
wcdmaMsgDecodingParams, 1500
Date, 1501
pMessage, 1501
pScAddr, 1501
pScAddrLength, 1501
pSenderAddr, 1501
pSenderAddrLength, 1501
pTextMsg, 1501
pTextMsgLength, 1501
Time, 1501
wcdmaMsgEncodingParams, 1501
alphabet, 1502
messageSize, 1502
pDestAddr, 1502
pPDUMessage, 1502
pTextMsg, 1502
wcdmaRRCState
nas_WCDMAInfoLTENeighborCell, 561
WCDMAInfoLTENeighborCell, 1498
wcdmaUARFCN, 1506
status, 1507
uarfcn, 1507
Wdisable
unpack_dms_SLQSDmsSwiGetPCInfo_t, 1133
wds.h
QMI_LITE_WDS_CURRENT_CALL_DB_MASK, 2257
QMI_LITE_WDS_LAST_CALL_DB_MASK, 2257
wds.h, 2250
BYT_STAT_STAT_MASK, 2256
liteQmiDataBearerMasks, 2257
PACK_WDS_IPV4, 2256
PACK_WDS_IPV6, 2257
pack_wds_DHCPv4ClientLeaseChange, 2257
pack_wds_GetAutoconnect, 2258
pack_wds_GetByteTotals, 2258
pack_wds_GetConnectionRate, 2258
pack_wds_GetDataBearerTechnology, 2259
pack_wds_GetDefaultProfile, 2259
pack_wds_GetDefaultProfileNum, 2260
pack_wds_GetDormancyState, 2260
pack_wds_GetLastMobileIPError, 2260
pack_wds_GetMobileIP, 2261
pack_wds_GetMobileIPProfile, 2261
pack_wds_GetPacketStatistics, 2262
pack_wds_GetPacketStatus, 2262
pack_wds_GetSessionDuration, 2263
pack_wds_GetSessionState, 2263
pack_wds_RMSetTransferStatistics, 2263
pack_wds_SLQSCreateProfile, 2267
pack_wds_SLQSDeleteProfile, 2267
pack_wds_SLQSGet3GPPConfigItem, 2268
pack_wds_SLQSGetCurrDataSystemStat, 2268
pack_wds_SLQSGetCurrentChannelRate, 2269
pack_wds_SLQSGetDUNCallInfo, 2269
pack_wds_SLQSGetDataBearerTechnology, 2269
pack_wds_SLQSGetProfileSettings, 2270
pack_wds_SLQSGetRuntimeSettings, 2270
pack_wds_SLQSModifyProfile, 2271
pack_wds_SLQSResetPacketStatics, 2271
pack_wds_SLQSSGetDHCPv4ClientConfig, 2273
pack_wds_SLQSSGetLoopback, 2273
pack_wds_SLQSSSetDHCPv4ClientConfig, 2274
pack_wds_SLQSSSetLoopback, 2274
pack_wds_SLQSSet3GPPConfigItem, 2272
pack_wds_SLQSSetIPFamilyPreference, 2272
pack_wds_SLQSSetWdsEventCallback, 2272
pack_wds_SLQSStartDataSession, 2274
pack_wds_SLQSStopDataSession, 2275
pack_wds_SLQSWdsGoActive, 2275
pack_wds_SLQSWdsGoDormant, 2276
pack_wds_SLQSWdsSetEventReport, 2276
pack_wds_SLQSWdsSwiPDPRuntimeSettings, 2277
pack_wds_SetAutoconnect, 2264
pack_wds_SetDefaultProfile, 2264
pack_wds_SetDefaultProfileNum, 2265
pack_wds_SetMobileIP, 2265
pack_wds_SetMobileIPParameters, 2265

pack_wds_SetMobileIPProfile, 2266
pack_wds_SetMuxID, 2266
unpack_wds_DHCPv4ClientLease_ind, 2277
unpack_wds_DHCPv4ClientLeaseChange, 2277
unpack_wds_GetAutoconnect, 2278
unpack_wds_GetByteTotals, 2278
unpack_wds_GetConnectionRate, 2278
unpack_wds_GetDataBearerTechnology, 2279
unpack_wds_GetDefaultProfile, 2279
unpack_wds_GetDefaultProfileNum, 2279
unpack_wds_GetDormancyState, 2280
unpack_wds_GetLastMobileIPError, 2280
unpack_wds_GetMobileIP, 2280
unpack_wds_GetMobileIPProfile, 2281
unpack_wds_GetPacketStatistics, 2281
unpack_wds_GetPacketStatus, 2281
unpack_wds_GetSessionDuration, 2282
unpack_wds_GetSessionState, 2282
unpack_wds_RMSetTransferStatistics, 2282
unpack_wds_RMTransferStatistics_ind, 2283
unpack_wds_SLQSCreateProfile, 2285
unpack_wds_SLQSDUNCallInfoCallBack_ind, 2286
unpack_wds_SLQSDeleteProfile, 2286
unpack_wds_SLQSGet3GPPConfigItem, 2286
unpack_wds_SLQSGetCurrDataSystemStat, 2287
unpack_wds_SLQSGetCurrentChannelRate, 2287
unpack_wds_SLQSGetDUNCallInfo, 2288
unpack_wds_SLQSGetDataBearerTechnology, 2287
unpack_wds_SLQSGetProfileSettings, 2288
unpack_wds_SLQSGetRuntimeSettings, 2288
unpack_wds_SLQSModifyProfile, 2289
unpack_wds_SLQSResetPacketStatics, 2289
unpack_wds_SLQSSGetDHCPv4ClientConfig, 2291
unpack_wds_SLQSSGetLoopback, 2291
unpack_wds_SLQSSSetDHCPv4ClientConfig, 2292
unpack_wds_SLQSSSetLoopback, 2292
unpack_wds_SLQSSet3GPPConfigItem, 2289
unpack_wds_SLQSSetIPFamilyPreference, 2289
unpack_wds_SLQSSetPacketSrvStatusCallback, 2290
unpack_wds_SLQSSetWdsEventCallback, 2290
unpack_wds_SLQSSetWdsEventCallback_ind, 2290
unpack_wds_SLQSStartDataSession, 2292
unpack_wds_SLQSStopDataSession, 2293
unpack_wds_SLQSWdsGoActive, 2293
unpack_wds_SLQSWdsGoDormant, 2293
unpack_wds_SLQSWdsSetEventReport, 2293
unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 2294
unpack_wds_SetAutoconnect, 2283
unpack_wds_SetDefaultProfile, 2283
unpack_wds_SetDefaultProfileNum, 2284
unpack_wds_SetMobileIP, 2284
unpack_wds_SetMobileIPParameters, 2284
unpack_wds_SetMobileIPProfile, 2285
unpack_wds_SetMuxID, 2285
UnpackQmiProfileInfo, 2257
wds_ChannelRateTlv, 1507
ChannelRate, 1508
TlvPresent, 1508
wds_ConnStatusTlv, 1508
MDMConnStatus, 1508
TlvPresent, 1508
wds_DHCPLeaseOptTlv, 1512
numOpt, 1512
optList, 1512
optListData, 1512
TlvPresent, 1512
wds_DHCPLeaseStateTlv, 1513
leaseState, 1513
TlvPresent, 1513
wds_DHCPOpt, 1513
optCode, 1514
optValLen, 1514
pOptVal, 1514
wds_DHCPProfileIdTlv, 1514
profileId, 1514
profileType, 1514
TlvPresent, 1514
wds_DHCPv4HWConfig, 1514
chaddr, 1515
chaddrLen, 1515
hwType, 1515
wds_DHCPv4Option, 1515
optCode, 1515
optVal, 1515
optValLen, 1516
wds_DHCPv4OptionList, 1516
numOpt, 1516
pOptList, 1516
wds_DHCPv4ProfileId, 1516
profileId, 1516
profileType, 1517
wds_DataBearTechTlv, 1510
DataBearerTech, 1511
TlvPresent, 1511
wds_DataULongLongTlv, 1511
TlvPresent, 1511
ullData, 1511
wds_DataULongTlv, 1512
TlvPresent, 1512
ulData, 1512
wds_Domain, 1517
domainLen, 1517
domainName, 1517
wds_DomainNameList, 1517
domain, 1518
numInstances, 1518
wds_DormStatTlv, 1518
DormancyStat, 1518
TlvPresent, 1518

wds_GPRSQoS, 1518
delayClass, 1519
meanThroughputClass, 1519
peakThroughputClass, 1519
precedenceClass, 1519
reliabilityClass, 1519
wds_IPV6AddressInfo, 1520
IPAddressV6, 1520
IPV6PrefixLen, 1520
wds_IPV6GWAddressInfo, 1520
gwAddressV6, 1521
gwV6PrefixLen, 1521
wds_IPv4AdTlv, 1519
IPv4Addr, 1519
TlvPresent, 1519
wds_LastMdmCallEndRsnTlv, 1521
CallEndReason, 1521
TlvPresent, 1521
wds_PCSCFFQDNAddress, 1521
fqdnAddr, 1522
fqdnLen, 1522
wds_PCSCFFQDNAddressList, 1522
numInstances, 1522
pcsfFQDNAddress, 1522
wds_PCSCFIPv4ServerAddressList, 1522
numInstances, 1523
pscsfIPv4Addr, 1523
wds_ProfileIdentifier, 1523
profileIndex, 1523
profileType, 1523
wds_RXBytesOKTlv, 1524
RxByteOKCnt, 1524
TlvPresent, 1524
wds_TXBytesOKTlv, 1526
TlvPresent, 1526
TxByteOKCnt, 1526
wds_TrStatInd, 1525
statsMask, 1525
statsPeriod, 1525
wds_UMTSMinQoS, 1526
deliveryErrSDU, 1528
grntDownlinkBitrate, 1528
grntUplinkBitrate, 1528
maxDownlinkBitrate, 1528
maxSDUSize, 1528
maxUplinkBitrate, 1528
qosDeliveryOrder, 1528
resBerRatio, 1528
sduErrorRatio, 1528
trafficClass, 1528
trafficPriority, 1528
transferDelay, 1528
wds_channelRate, 1507
CurrChanRxRate, 1507
CurrChanTxRate, 1507
wds_currNetworkInfo, 1508
NetworkType, 1510
RATMask, 1510
SOMask, 1510
wds_profileInfo, 1523
SlqsProfile3GPP, 1524
SlqsProfile3GPP2, 1524
wds_transferStatInd, 1524
StatsMask, 1525
StatsPeriod, 1525
WdsByteTotals, 1528
ByteTotalsElmntsV4, 1529
ByteTotalsElmntsV6, 1529
pV4sessionId, 1529
pV6sessionId, 1529
WdsByteTotalsElmnts, 1529
pRXTotalBytes, 1530
pTXTotalBytes, 1530
WdsClientLeaseChange, 1530
pEnableNotification, 1530
WdsConnectionRate, 1530
ConnRateElmntsV4, 1531
ConnRateElmntsV6, 1531
pV4sessionId, 1531
pV6sessionId, 1531
WdsConnectionRateElmnts, 1531
pCurrentChannelRXRate, 1532
pCurrentChannelTXRate, 1532
pMaxChannelRXRate, 1532
pMaxChannelTXRate, 1532
WdsDHCPv4ClientLeaseInd, 1532
pIPv4Addr, 1532
pLeaseState, 1532
pOptList, 1532
pProfileId, 1533
WdsDHCPv4Config, 1533
pHwConfig, 1533
pProfileId, 1533
pRequestOptionList, 1533
WdsDHCPv4HWConfig, 1534
chaddr, 1534
chaddrLen, 1534
hwType, 1534
WdsDHCPv4Option, 1534
optCode, 1535
optVal, 1535
optValLen, 1535
WdsDHCPv4OptionList, 1536
numOpt, 1536
pOptList, 1536
WdsDHCPv4ProfileId, 1537
profileId, 1537
profileType, 1538
wdsDhcpv4HwConfig, 1533
chaddr, 1534
chaddrLen, 1534
hwType, 1534
wdsDhcpv4Option, 1535
optCode, 1535
optVal, 1535
optValLen, 1535

wdsDhcpv4OptionList, 1536
numOpt, 1536
pOptList, 1537
wdsDhcpv4ProfileId, 1537
profileId, 1537
profileType, 1537
WdsIpAddressInfoReq, 1538
ip, 1539
pv4sessionId, 1539
pv6sessionId, 1539
WdsPktStatisticsElmnts, 1539
pRXDroppedCount, 1540
pRXOKBytesLastCall, 1540
pRXOkBytesCount, 1540
pRXPacketErrors, 1540
pRXPacketOverflows, 1540
pRXPacketSuccesses, 1540
pTXDroppedCount, 1541
pTXOKBytesLastCall, 1541
pTXOkBytesCount, 1541
pTXPacketErrors, 1541
pTXPacketOverflows, 1541
pTXPacketSuccesses, 1541
WdsPktStatisticsReq, 1541
pStatMask, 1541
WdsPktStatisticsResp, 1541
pV4sessionId, 1542
pV6sessionId, 1542
PktStatElmntsV4, 1542
PktStatElmntsV6, 1542
WdsProfileParam, 1542
SlqsProfile3GPP, 1542
SlqsProfile3GPP2, 1543
WdsRunTimeSettings, 1543
rts, 1543
v4sessionId, 1543
v6sessionId, 1543
wdsSetEventReportReq, 1543
pCurrChannelRateInd, 1545
pCurrDataBearerTechInd, 1545
pCurrPrefDataSysInd, 1545
pDataBearerTechInd, 1545
pDataCallStatusChangeInd, 1545
pDataSystemStatusChangeInd, 1545
pDormancyStatusInd, 1545
pEVDOPageMonPerChangeInd, 1545
pMIPStatusInd, 1545
pTransferStatInd, 1545
Wireless Data Service (WDS), 40

xAxis
sensorData, 901
sensorData_t, 902
xferStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1383
xid
pack_loc_SLQSLOCGetBestAvailPos_t, 653
pack_qmi_t, 712
unpack_qmi_t, 1254
xtra_start_gps_minutes
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
xtra_start_gps_week
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254
xtra_valid_duration_hours
GPSStateInfo, 306
unpack_pds_SLQSGetGPSStateInfo_t, 1254

yAxis
sensorData, 901
sensorData_t, 902
year
nas_timeInfo, 553
nas_UniversalTime, 559
timeInfo, 1033
UniversalTime, 1101

zAxis
sensorData, 901
sensorData_t, 902