LinuxQMISDK

SLQS04.00.05

Generated by Doxygen 1.8.6

Wed Mar 8 2017 07:17:34

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 **33**

5.1 File List . . . 33

6 Module Documentation **37**

6.1 Device Connectivity Service (DCS) . . . 37

6.1.1 Detailed Description . . . 37

6.2 Wireless Data Service (WDS) . . . 38

6.2.1 Detailed Description . . . 38

6.3 Device Management Service (DMS) . . . 39

6.3.1 Detailed Description . . . 39

6.4 Network Access Service (NAS) . . . 40

6.4.1 Detailed Description . . . 40

6.5 CallBacK registration (CBK) . . . 41

6.5.1 Detailed Description . . . 41

6.6 Short Message Service (SMS) . . . 42

6.6.1 Detailed Description . . . 42

6.7 Position Determination Service (PDS) . . . 43
6.7.1 Detailed Description . . . 43
6.8 Card Application Toolkit (CAT) . . . 44
6.8.1 Detailed Description . . . 44
6.9 Remote Management Service (RMS) . . . 45
6.9.1 Detailed Description . . . 45
6.10 Firmware Management Service (FMS) . . . 46
6.10.1 Detailed Description . . . 46
6.11 Open Mobile Alliance Service (OMA) . . . 47
6.11.1 Detailed Description . . . 47
6.12 Specific Absorption Rate (SAR) . . . 48
6.12.1 Detailed Description . . . 48
6.13 SWI Open Mobile Alliance Service (SWIOMA) . . . 49
6.13.1 Detailed Description . . . 49
6.14 Voice Service (VOICE) . . . 50
6.14.1 Detailed Description . . . 50
6.15 Non-service specific APIs (SWI) . . . 51
6.15.1 Detailed Description . . . 51
6.16 User Identity Module Service (UIM) . . . 52
6.16.1 Detailed Description . . . 52
6.17 Audio Service (AUDIO) . . . 53
6.17.1 Detailed Description . . . 53
6.18 Quality of Service (QOS) . . . 54
6.18.1 Detailed Description . . . 54
6.19 IMS Service (IMS) . . . 55
6.19.1 Detailed Description . . . 55
6.20 SWI Audio Service(SWIAUDIO) . . . 56
6.20.1 Detailed Description . . . 56
6.21 Location Service(LOC) . . . 57
6.21.1 Detailed Description . . . 57
6.22 Thermal Mitigation Device(TMD) . . . 58
6.22.1 Detailed Description . . . 58
6.23 AVMS (SWIAVMS) . . . 59
6.23.1 Detailed Description . . . 59

7 Namespace Documentation **61**

7.1 Tables Namespace Reference . . . 61
7.1.1 Detailed Description . . . 61

8 Data Structure Documentation **63**

8.1 _getIndicationRegResp Struct Reference . . . 63

8.1.1 Detailed Description . . . 63
8.1.2 Field Documentation . . . 64
8.1.2.1 pRegCallStatInfoEvt . . . 64
8.1.2.2 pRegTransLayerInfoEvt . . . 64
8.1.2.3 pRegTransNWRegInfoEvt . . . 64
8.2 _GetProfileSettingIn Struct Reference . . . 64
8.2.1 Detailed Description . . . 64
8.2.2 Field Documentation . . . 64
8.2.2.1 ProfileID . . . 64
8.2.2.2 ProfileType . . . 64
8.3 _GetProfileSettingOut Struct Reference . . . 64
8.3.1 Detailed Description . . . 64
8.3.2 Field Documentation . . . 65
8.3.2.1 curProfile . . . 65
8.3.2.2 pExtErrCode . . . 65
8.4 _getResetInfoNotification Struct Reference . . . 65
8.4.1 Detailed Description . . . 65
8.4.2 Field Documentation . . . 66
8.4.2.1 source . . . 66
8.4.2.2 type . . . 66
8.5 _getTransLayerInfoResp Struct Reference . . . 66
8.5.1 Detailed Description . . . 66
8.5.2 Field Documentation . . . 66
8.5.2.1 pRegInd . . . 66
8.5.2.2 pTransLayerInfo . . . 66
8.6 _getTransNWRegInfoResp Struct Reference . . . 66
8.6.1 Detailed Description . . . 66
8.6.2 Field Documentation . . . 67
8.6.2.1 pRegStatus . . . 67
8.7 _MitigationDevInfo Struct Reference . . . 67
8.7.1 Detailed Description . . . 67
8.7.2 Field Documentation . . . 67
8.7.2.1 deviceID . . . 67
8.7.2.2 deviceIdLen . . . 67
8.8 _modemTempNotification Struct Reference . . . 67
8.8.1 Detailed Description . . . 68
8.8.2 Field Documentation . . . 68
8.8.2.1 ModemTemperature . . . 68
8.8.2.2 ModemTempState . . . 68
8.9 _packetSrvStatus Struct Reference . . . 68

8.9.1 Detailed Description . . . 68
8.9.2 Field Documentation . . . 70
8.9.2.1 bearerID . . . 70
8.9.2.2 connStatus . . . 70
8.9.2.3 ipFamily . . . 70
8.9.2.4 pQmiInterfaceInfo . . . 70
8.9.2.5 reconfigReqd . . . 70
8.9.2.6 sessionEndReason . . . 70
8.9.2.7 techName . . . 70
8.9.2.8 verboseSessnEndReason . . . 70
8.9.2.9 verboseSessnEndReasonType . . . 70
8.10 _qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . 70
8.10.1 Detailed Description . . . 70
8.10.2 Field Documentation . . . 71
8.10.2.1 activated_ind . . . 71
8.10.2.2 CDMABroadcastConfig . . . 71
8.10.2.3 num_instances . . . 71
8.11 _qaQmi3GPPBroadcastCfgInfo Struct Reference . . . 71
8.11.1 Detailed Description . . . 71
8.11.2 Field Documentation . . . 71
8.11.2.1 activated_ind . . . 71
8.11.2.2 broadcastConfig . . . 71
8.11.2.3 num_instances . . . 71
8.12 _QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp Struct Reference . . . 71
8.12.1 Detailed Description . . . 72
8.12.2 Field Documentation . . . 73
8.12.2.1 bBinaryType . . . 73
8.12.2.2 bSerity . . . 73
8.12.2.3 bState . . . 73
8.12.2.4 bUserInputRequest . . . 73
8.12.2.5 szDescription . . . 73
8.12.2.6 szName . . . 73
8.12.2.7 szVersion . . . 73
8.12.2.8 TlvPresent . . . 73
8.12.2.9 ulPkgDownloadComplete . . . 73
8.12.2.10 ulPkgDownloadSize . . . 73
8.12.2.11 wDescriptionLength . . . 73
8.12.2.12 wNameLength . . . 73
8.12.2.13 wUpdateCompeteStatus . . . 73
8.12.2.14 wUserInputTimeout . . . 73

8.12.2.15 wVersionLength . . . 73
8.13 _QmiSwiAvmsEventReportConfigResp Struct Reference . . . 73
8.13.1 Detailed Description . . . 73
8.13.2 Field Documentation . . . 74
8.13.2.1 bState . . . 74
8.13.2.2 bUserInputRequest . . . 74
8.13.2.3 szAlertMsg . . . 74
8.13.2.4 TlvPresent . . . 74
8.13.2.5 wAlertMsgLength . . . 74
8.13.2.6 wUserInputTimeout . . . 74
8.14 _QmiSwiAvmsEventReportConnectionRequestResp Struct Reference . . . 74
8.14.1 Detailed Description . . . 74
8.14.2 Field Documentation . . . 75
8.14.2.1 bUserInputRequest . . . 75
8.14.2.2 TlvPresent . . . 75
8.14.2.3 wUserInputTimeout . . . 75
8.15 _QmiSwiAvmsEventReportDataSessionStatusResp Struct Reference . . . 75
8.15.1 Detailed Description . . . 75
8.15.2 Field Documentation . . . 75
8.15.2.1 bType . . . 75
8.15.2.2 TlvPresent . . . 75
8.15.2.3 wErrorCode . . . 75
8.16 _QmiSwiAvmsEventReportHTTPStatusResp Struct Reference . . . 75
8.16.1 Detailed Description . . . 75
8.16.2 Field Documentation . . . 76
8.16.2.1 TlvPresent . . . 76
8.16.2.2 wHTTPStatus . . . 77
8.17 _QmiSwiAvmsEventReportNotificationResp Struct Reference . . . 77
8.17.1 Detailed Description . . . 77
8.17.2 Field Documentation . . . 77
8.17.2.1 bNotification . . . 77
8.17.2.2 TlvPresent . . . 78
8.17.2.3 wSessionStatus . . . 78
8.18 _QmiSwiAvmsEventReportPackageIDResp Struct Reference . . . 78
8.18.1 Detailed Description . . . 78
8.18.2 Field Documentation . . . 78
8.18.2.1 bPackageID . . . 78
8.18.2.2 TlvPresent . . . 78
8.19 _QmiSwiAvmsEventReportRegStatusResp Struct Reference . . . 78
8.19.1 Detailed Description . . . 78

8.19.2 Field Documentation . . . 78
8.19.2.1 bRegStatus . . . 78
8.19.2.2 TlvPresent . . . 78
8.20 _QmiSwiAvmsEventReportSessionTypeResp Struct Reference . . . 78
8.20.1 Detailed Description . . . 79
8.20.2 Field Documentation . . . 79
8.20.2.1 bType . . . 79
8.20.2.2 TlvPresent . . . 79
8.21 _QmiSwiAvmsEventReportWAMSParamChangeResp Struct Reference . . . 79
8.21.1 Detailed Description . . . 79
8.21.2 Field Documentation . . . 79
8.21.2.1 TlvPresent . . . 79
8.21.2.2 wWamsChangeMask . . . 79
8.22 _QmiSwiAvmsSetSettingsAPNInfo Struct Reference . . . 79
8.22.1 Field Documentation . . . 80
8.22.1.1 bAPNLength . . . 80
8.22.1.2 bPWDLength . . . 80
8.22.1.3 bUnameLength . . . 80
8.22.1.4 szAPN . . . 80
8.22.1.5 szPWD . . . 80
8.22.1.6 szUname . . . 80
8.23 _QmiSwiAvmsSetSettingsConnectionRetryTimers Struct Reference . . . 80
8.23.1 Field Documentation . . . 80
8.23.1.1 Timers . . . 80
8.24 _QmiSwiAvmsSetSettingsPeriodInfo Struct Reference . . . 80
8.24.1 Field Documentation . . . 80
8.24.1.1 ulMax . . . 80
8.24.1.2 ulMin . . . 80
8.25 _setIndicationRegReq Struct Reference . . . 80
8.25.1 Detailed Description . . . 81
8.25.2 Field Documentation . . . 81
8.25.2.1 pRegCallStatInfoEvt . . . 81
8.25.2.2 pRegTransLayerInfoEvt . . . 81
8.25.2.3 pRegTransNWRegInfoEvt . . . 81
8.26 _slqs3GPPConfigItem Struct Reference . . . 81
8.26.1 Detailed Description . . . 82
8.26.2 Field Documentation . . . 83
8.26.2.1 LTEAttachProfileListLen . . . 83
8.26.2.2 p3gppRelease . . . 83
8.26.2.3 pDefaultPDNEnabled . . . 83

8.26.2.4 pLTEAttachProfile . . . 83
8.26.2.5 pLTEAttachProfileList . . . 83
8.26.2.6 pProfileList . . . 83
8.27 _SLQSAVMSSetSettingsReqPara Struct Reference . . . 83
8.27.1 Detailed Description . . . 83
8.27.2 Field Documentation . . . 85
8.27.2.1 pAPNInfo . . . 85
8.27.2.2 pAutoConnect . . . 85
8.27.2.3 pAutoReboot . . . 85
8.27.2.4 pConnectionRetryTimers . . . 85
8.27.2.5 pFwAutodownload . . . 85
8.27.2.6 pFwAutoSDM . . . 85
8.27.2.7 pFwAutoUpdate . . . 85
8.27.2.8 pNotifStore . . . 85
8.27.2.9 pPeroidInfo . . . 85
8.27.2.10 pPollingTimer . . . 85
8.28 _SLQSAVMSSettings Struct Reference . . . 85
8.28.1 Field Documentation . . . 85
8.28.1.1 pAPNInfo . . . 85
8.28.1.2 pAutoConnect . . . 85
8.28.1.3 pAutoReboot . . . 85
8.28.1.4 pConnectionRetryTimers . . . 85
8.28.1.5 pFwAutodownload . . . 85
8.28.1.6 pFwAutoSDM . . . 85
8.28.1.7 pFwAutoUpdate . . . 85
8.28.1.8 pNotificationStore . . . 86
8.28.1.9 pOMADMEnabled . . . 86
8.28.1.10 pPeroidsInfo . . . 86
8.28.1.11 pPollingTimer . . . 86
8.29 _SLQSAVMSSettingsAPNInfo Struct Reference . . . 86
8.29.1 Field Documentation . . . 86
8.29.1.1 bAPNLength . . . 86
8.29.1.2 bPWDLength . . . 86
8.29.1.3 bUnameLength . . . 86
8.29.1.4 pAPN . . . 86
8.29.1.5 pPWD . . . 86
8.29.1.6 pUname . . . 86
8.30 _SLQSAVMSSettingsConnectionRetryTimers Struct Reference . . . 86
8.30.1 Field Documentation . . . 86
8.30.1.1 Timers . . . 86

8.31 _SLQSAVMSSettingsPeriodsInfo Struct Reference . . . 86
8.31.1 Field Documentation . . . 87
8.31.1.1 max . . . 87
8.31.1.2 min . . . 87
8.32 _SlqsNas3GppNetworkRAT_ Struct Reference . . . 87
8.32.1 Detailed Description . . . 87
8.32.2 Field Documentation . . . 87
8.32.2.1 MCC . . . 87
8.32.2.2 MNC . . . 87
8.32.2.3 RAT . . . 87
8.33 _slqsNetworkScanInfo Struct Reference . . . 87
8.33.1 Detailed Description . . . 88
8.33.2 Field Documentation . . . 88
8.33.2.1 pNetworkInfo . . . 88
8.33.2.2 pNetworkInfoInstances . . . 88
8.33.2.3 pPCSDigitInfo . . . 88
8.33.2.4 pPCSDigitInstances . . . 88
8.33.2.5 pRATInfo . . . 88
8.33.2.6 pRATInstances . . . 88
8.33.2.7 pScanResult . . . 88
8.34 _SLQSOMADMSessionInfo Struct Reference . . . 89
8.34.1 Detailed Description . . . 89
8.34.2 Field Documentation . . . 91
8.34.2.1 pDate . . . 91
8.34.2.2 pDateLength . . . 91
8.34.2.3 pPkgDescLength . . . 91
8.34.2.4 pPkgDescription . . . 91
8.34.2.5 pPkgName . . . 91
8.34.2.6 pPkgNameLength . . . 91
8.34.2.7 pRetryCount . . . 91
8.34.2.8 pSessionState . . . 91
8.34.2.9 pSessionType . . . 91
8.34.2.10 pSeverity . . . 91
8.34.2.11 pSource . . . 91
8.34.2.12 pSourceLength . . . 91
8.34.2.13 pStatus . . . 91
8.34.2.14 pTime . . . 91
8.34.2.15 pTimeLength . . . 91
8.34.2.16 pUpdateCompleteStatus . . . 91
8.35 _SLQSOMADMSettings Struct Reference . . . 91

8.35.1 Detailed Description . . . 91
8.35.2 Field Documentation . . . 93
8.35.2.1 pAutosdm . . . 93
8.35.2.2 pFOTAdownload . . . 93
8.35.2.3 pFOTAUpdate . . . 93
8.35.2.4 pFwAutoCheck . . . 93
8.35.2.5 pOMADMEnabled . . . 93
8.36 _SLQSOMADMSettingsReqParams Struct Reference . . . 93
8.36.1 Detailed Description . . . 93
8.36.2 Field Documentation . . . 93
8.36.2.1 FOTAdownload . . . 94
8.36.2.2 FOTAUpdate . . . 94
8.36.2.3 pAutosdm . . . 94
8.37 _SLQSOMADMSettingsReqParams3 Struct Reference . . . 94
8.37.1 Detailed Description . . . 94
8.37.2 Field Documentation . . . 94
8.37.2.1 FOTAdownload . . . 95
8.37.2.2 FOTAUpdate . . . 95
8.37.2.3 pAutosdm . . . 95
8.37.2.4 pFwAutoCheck . . . 95
8.38 _SLQSSwiGetHostDevInfoParams Struct Reference . . . 95
8.38.1 Detailed Description . . . 95
8.38.2 Field Documentation . . . 96
8.38.2.1 bManSize . . . 96
8.38.2.2 bModelSize . . . 96
8.38.2.3 bPlasmaIDSize . . . 96
8.38.2.4 bSWVerSize . . . 96
8.38.2.5 pManString . . . 96
8.38.2.6 pModelString . . . 96
8.38.2.7 pPlasmaIDString . . . 96
8.38.2.8 pSWVerString . . . 96
8.39 _SLQSSwiGetOSInfoParams Struct Reference . . . 96
8.39.1 Detailed Description . . . 96
8.39.2 Field Documentation . . . 96
8.39.2.1 bNameSize . . . 96
8.39.2.2 bVersionSize . . . 96
8.39.2.3 pNameString . . . 97
8.39.2.4 pVersionString . . . 97
8.40 _SLQSSwiGetSerialNoExtParams Struct Reference . . . 97
8.40.1 Detailed Description . . . 97

8.40.2 Field Documentation . . . 97
8.40.2.1 meidLength . . . 97
8.40.2.2 pMeidString . . . 97
8.41 _SLQSSwiSetHostDevInfoParams Struct Reference . . . 97
8.41.1 Detailed Description . . . 97
8.41.2 Field Documentation . . . 98
8.41.2.1 bManSize . . . 98
8.41.2.2 bModelSize . . . 98
8.41.2.3 bPlasmaIDSize . . . 98
8.41.2.4 bSWVerSize . . . 98
8.41.2.5 pManString . . . 98
8.41.2.6 pModelString . . . 98
8.41.2.7 pPlasmaIDString . . . 98
8.41.2.8 pSWVerString . . . 98
8.42 _SLQSSwiSetOSInfoParams Struct Reference . . . 98
8.42.1 Detailed Description . . . 99
8.42.2 Field Documentation . . . 99
8.42.2.1 bNameSize . . . 99
8.42.2.2 bVersionSize . . . 99
8.42.2.3 pNameString . . . 99
8.42.2.4 pVersionString . . . 99
8.43 _SwiAvmsEventInd Struct Reference . . . 99
8.43.1 Field Documentation . . . 99
8.43.1.1 BinaryUpdateSessionInfo . . . 99
8.43.1.2 Config . . . 99
8.43.1.3 ConnectionRequest . . . 99
8.43.1.4 DataSessionStatus . . . 100
8.43.1.5 HTTPStatus . . . 100
8.43.1.6 Notification . . . 100
8.43.1.7 PackageID . . . 100
8.43.1.8 RegStatus . . . 100
8.43.1.9 SessionType . . . 100
8.43.1.10 WAMSParaChanged . . . 100
8.44 _sysSelectPrefInfo Struct Reference . . . 100
8.44.1 Detailed Description . . . 100
8.44.2 Field Documentation . . . 104
8.44.2.1 pBandPref . . . 104
8.44.2.2 pEmerMode . . . 104
8.44.2.3 pGWAcqOrderPref . . . 104
8.44.2.4 pLTEBandPref . . . 104

8.44.2.5 pModePref . . . 104
8.44.2.6 pNetSelPref . . . 104
8.44.2.7 pPRLPref . . . 104
8.44.2.8 pRoamPref . . . 104
8.44.2.9 pSrvDomainPref . . . 104
8.45 _sysSelectPrefParams Struct Reference . . . 105
8.45.1 Detailed Description . . . 105
8.45.2 Field Documentation . . . 110
8.45.2.1 pAcqOrderPref . . . 110
8.45.2.2 pBandPref . . . 110
8.45.2.3 pChgDuration . . . 110
8.45.2.4 pCSGID . . . 110
8.45.2.5 pEmerMode . . . 110
8.45.2.6 pGWAcqOrderPref . . . 110
8.45.2.7 pLTEBandPref . . . 110
8.45.2.8 pMNCIncPCSDigStat . . . 110
8.45.2.9 pModePref . . . 110
8.45.2.10 pNetSelPref . . . 110
8.45.2.11 pPRLPref . . . 110
8.45.2.12 pRAT . . . 110
8.45.2.13 pRoamPref . . . 111
8.45.2.14 pSrvDomainPref . . . 111
8.45.2.15 pSrvRegRestriction . . . 111
8.45.2.16 pTdscdmaBandPref . . . 111
8.46 _transLayerinfo Struct Reference . . . 111
8.46.1 Detailed Description . . . 111
8.46.2 Field Documentation . . . 111
8.46.2.1 TransCap . . . 111
8.46.2.2 TransType . . . 111
8.47 _transLayerInfoNotification Struct Reference . . . 111
8.47.1 Detailed Description . . . 111
8.47.2 Field Documentation . . . 112
8.47.2.1 pTransLayerInfo . . . 112
8.47.2.2 regInd . . . 112
8.48 _transNWRegInfoNotification Struct Reference . . . 112
8.48.1 Detailed Description . . . 112
8.48.2 Field Documentation . . . 112
8.48.2.1 NWRegStat . . . 112
8.49 accelAcceptReady_s Struct Reference . . . 113
8.49.1 Detailed Description . . . 113

8.49.2 Field Documentation . . . 113
8.49.2.1 batchPerSec . . . 113
8.49.2.2 injectEnable . . . 113
8.49.2.3 samplesPerBatch . . . 113
8.50 accelTempAcceptReady_s Struct Reference . . . 113
8.50.1 Detailed Description . . . 113
8.50.2 Field Documentation . . . 114
8.50.2.1 batchPerSec . . . 114
8.50.2.2 injectEnable . . . 114
8.50.2.3 samplesPerBatch . . . 114
8.51 acqOrderPref Struct Reference . . . 114
8.51.1 Detailed Description . . . 114
8.51.2 Field Documentation . . . 115
8.51.2.1 acqOrdeLen . . . 115
8.51.2.2 pAcqOrder . . . 115
8.52 ActPilotPNElement Struct Reference . . . 115
8.52.1 Detailed Description . . . 115
8.52.2 Field Documentation . . . 115
8.52.2.1 ActSetPilotPN . . . 115
8.52.2.2 ActSetPilotPNStrength . . . 115
8.53 AddCDMASysInfo Struct Reference . . . 115
8.53.1 Detailed Description . . . 115
8.53.2 Field Documentation . . . 116
8.53.2.1 geoSysIdx . . . 116
8.53.2.2 regPrd . . . 116
8.54 AddSysInfo Struct Reference . . . 116
8.54.1 Detailed Description . . . 116
8.54.2 Field Documentation . . . 116
8.54.2.1 cellBroadcastCap . . . 116
8.54.2.2 geoSysIdx . . . 116
8.55 airTimer Struct Reference . . . 116
8.55.1 Detailed Description . . . 116
8.55.2 Field Documentation . . . 117
8.55.2.1 airTimerValue . . . 117
8.55.2.2 namID . . . 117
8.56 allCallsAlphaIDInfo Struct Reference . . . 117
8.56.1 Detailed Description . . . 117
8.56.2 Field Documentation . . . 117
8.56.2.1 AlphaIDInfo . . . 117
8.56.2.2 callID . . . 117

8.57 allCallsDiagInfo Struct Reference . . . 117
8.57.1 Detailed Description . . . 118
8.57.2 Field Documentation . . . 118
8.57.2.1 callID . . . 118
8.57.2.2 DiagInfo . . . 118
8.58 allCallsUUSInfo Struct Reference . . . 118
8.58.1 Detailed Description . . . 118
8.58.2 Field Documentation . . . 118
8.58.2.1 callID . . . 118
8.58.2.2 uusInfo . . . 118
8.59 alphaIDInfo Struct Reference . . . 118
8.59.1 Detailed Description . . . 119
8.59.2 Field Documentation . . . 119
8.59.2.1 alphaDcs . . . 119
8.59.2.2 alphaLen . . . 119
8.59.2.3 alphaText . . . 119
8.60 altitudeSrcInfo Struct Reference . . . 119
8.60.1 Detailed Description . . . 119
8.60.2 Field Documentation . . . 120
8.60.2.1 coverage . . . 120
8.60.2.2 linkage . . . 120
8.60.2.3 source . . . 120
8.61 altSrcInfo_t Struct Reference . . . 120
8.61.1 Detailed Description . . . 120
8.61.2 Field Documentation . . . 121
8.61.2.1 coverage . . . 121
8.61.2.2 linkage . . . 121
8.61.2.3 source . . . 121
8.62 appStats Struct Reference . . . 121
8.62.1 Detailed Description . . . 122
8.62.2 Field Documentation . . . 124
8.62.2.1 aidLength . . . 124
8.62.2.2 aidVal . . . 124
8.62.2.3 appState . . . 124
8.62.2.4 appType . . . 124
8.62.2.5 persoFeature . . . 124
8.62.2.6 persoRetries . . . 124
8.62.2.7 persoState . . . 124
8.62.2.8 persoUnblockRetries . . . 124
8.62.2.9 pin1Retries . . . 124

8.62.2.10 pin1State . . . 124
8.62.2.11 pin2Retries . . . 124
8.62.2.12 pin2State . . . 124
8.62.2.13 puk1Retries . . . 124
8.62.2.14 puk2Retries . . . 124
8.62.2.15 univPin . . . 124
8.63 appStatus Struct Reference . . . 124
8.63.1 Detailed Description . . . 125
8.63.2 Field Documentation . . . 127
8.63.2.1 aidLength . . . 127
8.63.2.2 aidVal . . . 127
8.63.2.3 appState . . . 127
8.63.2.4 appType . . . 127
8.63.2.5 persoFeature . . . 127
8.63.2.6 persoRetries . . . 127
8.63.2.7 persoState . . . 127
8.63.2.8 persoUnblockRetries . . . 127
8.63.2.9 pin1Retries . . . 127
8.63.2.10 pin1State . . . 127
8.63.2.11 pin2Retries . . . 127
8.63.2.12 pin2State . . . 127
8.63.2.13 puk1Retries . . . 127
8.63.2.14 puk2Retries . . . 127
8.63.2.15 univPin . . . 127
8.64 arrAlertingPattern Struct Reference . . . 127
8.64.1 Detailed Description . . . 128
8.64.2 Field Documentation . . . 128
8.64.2.1 alertingPattern . . . 128
8.64.2.2 callID . . . 128
8.64.2.3 numInstances . . . 128
8.65 arrAlertingType Struct Reference . . . 128
8.65.1 Detailed Description . . . 128
8.65.2 Field Documentation . . . 129
8.65.2.1 AlertingType . . . 129
8.65.2.2 callID . . . 129
8.65.2.3 numInstances . . . 129
8.66 arrAlphaID Struct Reference . . . 129
8.66.1 Detailed Description . . . 129
8.66.2 Field Documentation . . . 129
8.66.2.1 allCallsAlphaIDInfoArr . . . 129

8.66.2.2 numInstances . . . 129
8.67 arrCalledPartyNum Struct Reference . . . 129
8.67.1 Detailed Description . . . 130
8.67.2 Field Documentation . . . 130
8.67.2.1 CalledPartyNum . . . 130
8.67.2.2 numInstances . . . 130
8.68 arrCallEndReason Struct Reference . . . 130
8.68.1 Detailed Description . . . 130
8.68.2 Field Documentation . . . 131
8.68.2.1 callEndReason . . . 131
8.68.2.2 callID . . . 131
8.68.2.3 numInstances . . . 131
8.69 arrCallInfo Struct Reference . . . 131
8.69.1 Detailed Description . . . 131
8.69.2 Field Documentation . . . 131
8.69.2.1 getAllCallInfo . . . 131
8.69.2.2 numInstances . . . 131
8.70 arrConnectPartyNum Struct Reference . . . 131
8.70.1 Detailed Description . . . 131
8.70.2 Field Documentation . . . 132
8.70.2.1 ConnectedPartyNum . . . 132
8.70.2.2 numInstances . . . 132
8.71 arrDiagInfo Struct Reference . . . 132
8.71.1 Detailed Description . . . 132
8.71.2 Field Documentation . . . 132
8.71.2.1 DiagInfo . . . 132
8.71.2.2 numInstances . . . 132
8.72 arrRedirPartyNum Struct Reference . . . 132
8.72.1 Detailed Description . . . 132
8.72.2 Field Documentation . . . 133
8.72.2.1 numInstances . . . 133
8.72.2.2 RedirPartyNum . . . 133
8.73 arrRemotePartyName Struct Reference . . . 133
8.73.1 Detailed Description . . . 133
8.73.2 Field Documentation . . . 133
8.73.2.1 GetAllCallRmtPtyName . . . 133
8.73.2.2 numInstances . . . 133
8.74 arrRemotePartyNum Struct Reference . . . 133
8.74.1 Detailed Description . . . 134
8.74.2 Field Documentation . . . 134

8.74.2.1 numInstances . . . 134
8.74.2.2 RmtPtyNum . . . 134
8.75 arrSvcOption Struct Reference . . . 134
8.75.1 Detailed Description . . . 134
8.75.2 Field Documentation . . . 134
8.75.2.1 callID . . . 134
8.75.2.2 numInstances . . . 134
8.75.2.3 srvOption . . . 135
8.76 arrUUSInfo Struct Reference . . . 135
8.76.1 Detailed Description . . . 135
8.76.2 Field Documentation . . . 135
8.76.2.1 AllCallsUUSInfo . . . 135
8.76.2.2 numInstances . . . 135
8.77 authenticateResult Struct Reference . . . 135
8.77.1 Detailed Description . . . 135
8.77.2 Field Documentation . . . 136
8.77.2.1 content . . . 136
8.77.2.2 contentLen . . . 136
8.78 authenticationData Struct Reference . . . 136
8.78.1 Detailed Description . . . 136
8.78.2 Field Documentation . . . 136
8.78.2.1 context . . . 137
8.78.2.2 data . . . 137
8.78.2.3 dataLen . . . 137
8.79 BandCapabilityResp Struct Reference . . . 137
8.79.1 Detailed Description . . . 137
8.79.2 Field Documentation . . . 140
8.79.2.1 bandCapability . . . 140
8.79.2.2 pLteBandCapability . . . 140
8.79.2.3 pTdsBandCapability . . . 140
8.80 BdsSV Struct Reference . . . 140
8.80.1 Detailed Description . . . 140
8.80.2 Field Documentation . . . 140
8.80.2.1 id . . . 140
8.80.2.2 mask . . . 140
8.81 BdsSVInfo Struct Reference . . . 140
8.81.1 Detailed Description . . . 141
8.81.2 Field Documentation . . . 141
8.81.2.1 len . . . 141
8.81.2.2 pSV . . . 141

8.82 BroadcastConfig Struct Reference . . . 141
8.82.1 Detailed Description . . . 141
8.82.2 Field Documentation . . . 142
8.82.2.1 fromServiceId . . . 142
8.82.2.2 selected . . . 142
8.82.2.3 toServiceId . . . 142
8.83 burstDTMFInfo Struct Reference . . . 142
8.83.1 Detailed Description . . . 142
8.83.2 Field Documentation . . . 142
8.83.2.1 digitCnt . . . 142
8.83.2.2 pCallID . . . 142
8.83.2.3 pDigitBuff . . . 142
8.84 CallBarringSysInfo Struct Reference . . . 142
8.84.1 Detailed Description . . . 143
8.84.2 Field Documentation . . . 143
8.84.2.1 csBarStatus . . . 143
8.84.2.2 psBarStatus . . . 143
8.85 callBarStatus Struct Reference . . . 143
8.85.1 Detailed Description . . . 143
8.85.2 Field Documentation . . . 144
8.85.2.1 csBarStatus . . . 144
8.85.2.2 psBarStatus . . . 144
8.86 calledPartyInfo Struct Reference . . . 144
8.86.1 Detailed Description . . . 144
8.86.2 Field Documentation . . . 145
8.86.2.1 number . . . 145
8.86.2.2 numLen . . . 145
8.86.2.3 numPlan . . . 145
8.86.2.4 numType . . . 146
8.86.2.5 PI . . . 146
8.86.2.6 SI . . . 146
8.87 calledPartySubAdd Struct Reference . . . 146
8.87.1 Detailed Description . . . 146
8.87.2 Field Documentation . . . 146
8.87.2.1 extBit . . . 146
8.87.2.2 oddEvenInd . . . 146
8.87.2.3 subAddr . . . 146
8.87.2.4 subAddrLen . . . 146
8.87.2.5 subAddrType . . . 147
8.88 callerIDInfo Struct Reference . . . 147

8.88.1 Detailed Description . . . 147
8.88.2 Field Documentation . . . 147
8.88.2.1 callerID . . . 147
8.88.2.2 callerIDLen . . . 147
8.88.2.3 PI . . . 147
8.89 callFwdTypeAndPlan Struct Reference . . . 147
8.89.1 Detailed Description . . . 147
8.89.2 Field Documentation . . . 148
8.89.2.1 numberPlan . . . 148
8.89.2.2 numberType . . . 148
8.90 callFWExtInfo Struct Reference . . . 148
8.90.1 Detailed Description . . . 148
8.90.2 Field Documentation . . . 150
8.90.2.1 noReplyTimer . . . 150
8.90.2.2 number . . . 150
8.90.2.3 numLen . . . 150
8.90.2.4 numPlan . . . 150
8.90.2.5 numType . . . 150
8.90.2.6 PI . . . 150
8.90.2.7 SI . . . 150
8.90.2.8 SvcClass . . . 150
8.90.2.9 SvcStatus . . . 150
8.91 callFWInfo Struct Reference . . . 150
8.91.1 Detailed Description . . . 150
8.91.2 Field Documentation . . . 151
8.91.2.1 noReplyTimer . . . 151
8.91.2.2 number . . . 151
8.91.2.3 numLen . . . 151
8.91.2.4 SvcClass . . . 151
8.91.2.5 SvcStatus . . . 151
8.92 callInfo Struct Reference . . . 151
8.92.1 Detailed Description . . . 151
8.92.2 Field Documentation . . . 152
8.92.2.1 callID . . . 152
8.92.2.2 callState . . . 153
8.92.2.3 callType . . . 153
8.92.2.4 direction . . . 153
8.92.2.5 mode . . . 153
8.93 callingPartyInfo Struct Reference . . . 153
8.93.1 Detailed Description . . . 153

8.93.2 Field Documentation . . . 154
8.93.2.1 number . . . 154
8.93.2.2 numLen . . . 154
8.93.2.3 numPlan . . . 154
8.93.2.4 numType . . . 154
8.93.2.5 PI . . . 154
8.93.2.6 SI . . . 154
8.94 cardResult Struct Reference . . . 154
8.94.1 Detailed Description . . . 155
8.94.2 Field Documentation . . . 155
8.94.2.1 sw1 . . . 155
8.94.2.2 sw2 . . . 155
8.95 cardStatus Struct Reference . . . 155
8.95.1 Detailed Description . . . 155
8.95.2 Field Documentation . . . 156
8.95.2.1 index1xPri . . . 156
8.95.2.2 index1xSec . . . 156
8.95.2.3 indexGwPri . . . 156
8.95.2.4 indexGwSec . . . 156
8.95.2.5 numSlot . . . 156
8.95.2.6 SlotInfo . . . 156
8.96 CarrierImage_t Struct Reference . . . 156
8.96.1 Detailed Description . . . 156
8.96.2 Field Documentation . . . 157
8.96.2.1 m_FwBuildId . . . 157
8.96.2.2 m_FwImageId . . . 157
8.96.2.3 m_nCarrierId . . . 157
8.96.2.4 m_nFolderId . . . 157
8.96.2.5 m_nStorage . . . 157
8.96.2.6 m_PriBuildId . . . 157
8.96.2.7 m_PriImageId . . . 157
8.97 CatAlphaIdentifierTlv Struct Reference . . . 157
8.97.1 Detailed Description . . . 158
8.97.2 Field Documentation . . . 158
8.97.2.1 AlphaID . . . 158
8.97.2.2 AlphaIDLength . . . 158
8.97.2.3 ReferenceID . . . 158
8.98 CatCommonEventTlv Struct Reference . . . 158
8.98.1 Field Documentation . . . 158
8.98.1.1 CatEvent . . . 158

8.98.1.2 EventID . . . 158
8.98.1.3 EventLength . . . 158
8.98.1.4 TlvPresent . . . 158
8.99 CatEndProactiveSessionTlv Struct Reference . . . 158
8.99.1 Detailed Description . . . 158
8.99.2 Field Documentation . . . 159
8.99.2.1 EndProactiveSession . . . 159
8.100 CATEventDataType Struct Reference . . . 159
8.100.1 Field Documentation . . . 159
8.100.1.1 eventMask . . . 159
8.100.1.2 pErrorMask . . . 159
8.101 CatEventIDDataTlv Struct Reference . . . 159
8.101.1 Detailed Description . . . 159
8.101.2 Field Documentation . . . 159
8.101.2.1 Data . . . 159
8.101.2.2 DataLength . . . 159
8.101.2.3 ReferenceID . . . 159
8.102 CatEventListTlv Struct Reference . . . 159
8.102.1 Detailed Description . . . 160
8.102.2 Field Documentation . . . 160
8.102.2.1 SetupEventList . . . 160
8.103 CatRefreshTlv Struct Reference . . . 160
8.103.1 Detailed Description . . . 160
8.103.2 Field Documentation . . . 160
8.103.2.1 RefreshMode . . . 160
8.103.2.2 RefreshStage . . . 160
8.104 ccSUPSType Struct Reference . . . 160
8.104.1 Detailed Description . . . 161
8.104.2 Field Documentation . . . 161
8.104.2.1 reason . . . 161
8.104.2.2 svcType . . . 161
8.105 CDMABroadcastConfig Struct Reference . . . 161
8.105.1 Detailed Description . . . 161
8.105.2 Field Documentation . . . 162
8.105.2.1 language . . . 162
8.105.2.2 selected . . . 162
8.105.2.3 serviceCategory . . . 162
8.106 CDMAChannel Struct Reference . . . 162
8.106.1 Detailed Description . . . 162
8.106.2 Field Documentation . . . 162

8.106.2.1 priChA . . . 162
8.106.2.2 priChB . . . 162
8.106.2.3 secChA . . . 163
8.106.2.4 secChB . . . 163
8.107 CDMAECIOThresh Struct Reference . . . 163
8.107.1 Detailed Description . . . 163
8.107.2 Field Documentation . . . 163
8.107.2.1 CDMAECIOThreshListLen . . . 163
8.107.2.2 pCDMAECIOThreshList . . . 163
8.108 CDMAInfo Struct Reference . . . 163
8.108.1 Detailed Description . . . 163
8.108.2 Field Documentation . . . 164
8.108.2.1 baseId . . . 164
8.108.2.2 baseLat . . . 164
8.108.2.3 baseLong . . . 164
8.108.2.4 nid . . . 164
8.108.2.5 refpn . . . 164
8.108.2.6 sid . . . 164
8.109 cdmaMsgDecodingParams Struct Reference . . . 164
8.109.1 Detailed Description . . . 165
8.109.2 Field Documentation . . . 166
8.109.2.1 absoluteValidity . . . 166
8.109.2.2 mcTimeStamp . . . 166
8.109.2.3 messageLength . . . 166
8.109.2.4 pAlertPriority . . . 166
8.109.2.5 pCallbkAddr . . . 166
8.109.2.6 pCallbkAddrLength . . . 166
8.109.2.7 pDisplayMode . . . 166
8.109.2.8 pLanguage . . . 167
8.109.2.9 pMessage . . . 167
8.109.2.10 pMessageID . . . 167
8.109.2.11 pPriority . . . 167
8.109.2.12 pPrivacy . . . 167
8.109.2.13 pReadAcknowledgementReq . . . 167
8.109.2.14 pRelativeValidity . . . 167
8.109.2.15 pSenderAddr . . . 167
8.109.2.16 pSenderAddrLength . . . 167
8.109.2.17 pTextMsg . . . 167
8.109.2.18 pTextMsgLength . . . 167
8.109.2.19 pUserAcknowledgementReq . . . 167

8.110 cdmaMsgEncodingParams Struct Reference . . . 167
8.110.1 Detailed Description . . . 167
8.110.2 Field Documentation . . . 168
8.110.2.1 messageId . . . 168
8.110.2.2 pCallbackAddr . . . 168
8.110.2.3 pDestAddr . . . 168
8.110.2.4 pEncodingAlphabet . . . 168
8.110.2.5 pMessage . . . 168
8.110.2.6 pMessageSize . . . 168
8.110.2.7 pPriority . . . 168
8.110.2.8 pRelValidity . . . 169
8.110.2.9 pTextMsg . . . 169
8.110.2.10 textMsgLength . . . 169
8.111 CDMARSSIThresh Struct Reference . . . 169
8.111.1 Detailed Description . . . 169
8.111.2 Field Documentation . . . 169
8.111.2.1 CDMARSSIThreshListLen . . . 169
8.111.2.2 pCDMARSSIThreshList . . . 169
8.112 CDMASSInfo Struct Reference . . . 169
8.112.1 Detailed Description . . . 169
8.112.2 Field Documentation . . . 170
8.112.2.1 ecio . . . 170
8.112.2.2 rssi . . . 170
8.113 cdmaSSInfo Struct Reference . . . 170
8.113.1 Detailed Description . . . 170
8.113.2 Field Documentation . . . 170
8.113.2.1 ecio . . . 170
8.113.2.2 rssi . . . 170
8.114 CDMASysInfo Struct Reference . . . 170
8.114.1 Detailed Description . . . 171
8.114.2 Field Documentation . . . 173
8.114.2.1 baseId . . . 173
8.114.2.2 baseLat . . . 173
8.114.2.3 baseLong . . . 173
8.114.2.4 bsInfoValid . . . 173
8.114.2.5 bsPRev . . . 173
8.114.2.6 bsPRevValid . . . 173
8.114.2.7 ccsSupported . . . 173
8.114.2.8 ccsSupportedValid . . . 173
8.114.2.9 cdmaSysIdValid . . . 173

8.114.2.10 isSysPrlMatch . . . 173
8.114.2.11 isSysPrlMatchValid . . . 173
8.114.2.12 MCC . . . 173
8.114.2.13 MNC . . . 173
8.114.2.14 networkID . . . 173
8.114.2.15 networkIdValid . . . 174
8.114.2.16 packetZone . . . 174
8.114.2.17 packetZoneValid . . . 174
8.114.2.18 pRevInUse . . . 174
8.114.2.19 pRevInUseValid . . . 174
8.114.2.20 sysInfoCDMA . . . 174
8.114.2.21 systemID . . . 174
8.115 CDMASysInfoExt Struct Reference . . . 174
8.115.1 Detailed Description . . . 174
8.115.2 Field Documentation . . . 174
8.115.2.1 imsi_11_12 . . . 174
8.115.2.2 MCC . . . 174
8.116 CellDb Struct Reference . . . 174
8.116.1 Detailed Description . . . 174
8.116.2 Field Documentation . . . 175
8.116.2.1 mask . . . 175
8.117 cellParams Struct Reference . . . 175
8.117.1 Detailed Description . . . 175
8.117.2 Field Documentation . . . 176
8.117.2.1 pci . . . 176
8.117.2.2 rsrp . . . 176
8.117.2.3 rsrq . . . 176
8.117.2.4 rssi . . . 176
8.117.2.5 srxlev . . . 176
8.118 changeUIMPIN Struct Reference . . . 176
8.118.1 Detailed Description . . . 176
8.118.2 Field Documentation . . . 177
8.118.2.1 oldPINLen . . . 177
8.118.2.2 oldPINVal . . . 177
8.118.2.3 pinID . . . 177
8.118.2.4 pinLen . . . 177
8.118.2.5 pinVal . . . 177
8.119 channelRate Struct Reference . . . 177
8.119.1 Detailed Description . . . 177
8.119.2 Field Documentation . . . 177

8.119.2.1 CurrChanRxRate . . . 177
8.119.2.2 CurrChanTxRate . . . 177
8.120 ChannelRate Struct Reference . . . 177
8.120.1 Detailed Description . . . 178
8.120.2 Field Documentation . . . 178
8.120.2.1 CurrChanRxRate . . . 178
8.120.2.2 CurrChanTxRate . . . 178
8.120.2.3 MaxChanRxRate . . . 178
8.120.2.4 MaxChanTxRate . . . 178
8.121 CLIPResp Struct Reference . . . 178
8.121.1 Detailed Description . . . 178
8.121.2 Field Documentation . . . 179
8.121.2.1 ActiveStatus . . . 179
8.121.2.2 ProvisionStatus . . . 179
8.122 CLIRResp Struct Reference . . . 179
8.122.1 Detailed Description . . . 179
8.122.2 Field Documentation . . . 179
8.122.2.1 ActiveStatus . . . 179
8.122.2.2 ProvisionStatus . . . 179
8.123 ClkInfo Struct Reference . . . 180
8.123.1 Detailed Description . . . 180
8.123.2 Field Documentation . . . 181
8.123.2.1 mask . . . 181
8.124 CNAPResp Struct Reference . . . 181
8.124.1 Detailed Description . . . 181
8.124.2 Field Documentation . . . 181
8.124.2.1 ActiveStatus . . . 181
8.124.2.2 ProvisionStatus . . . 181
8.125 COLPResp Struct Reference . . . 181
8.125.1 Detailed Description . . . 181
8.125.2 Field Documentation . . . 182
8.125.2.1 ActiveStatus . . . 182
8.125.2.2 ProvisionStatus . . . 182
8.126 COLRResp Struct Reference . . . 182
8.126.1 Detailed Description . . . 182
8.126.2 Field Documentation . . . 183
8.126.2.1 ActiveStatus . . . 183
8.126.2.2 ProvisionStatus . . . 183
8.127 CommInfo Struct Reference . . . 183
8.127.1 Detailed Description . . . 183

8.127.2 Field Documentation . . . 184
8.127.2.1 imsRegState . . . 184
8.127.2.2 modemMode . . . 184
8.127.2.3 psState . . . 184
8.127.2.4 systemMode . . . 184
8.127.2.5 temperature . . . 184
8.128 ConnectionStatus Struct Reference . . . 184
8.128.1 Detailed Description . . . 185
8.128.2 Field Documentation . . . 185
8.128.2.1 MDMCallDuration . . . 185
8.128.2.2 MDMConnStatus . . . 185
8.129 connectionStatus Struct Reference . . . 185
8.129.1 Detailed Description . . . 185
8.129.2 Field Documentation . . . 185
8.129.2.1 MDMCallDuration . . . 185
8.129.2.2 MDMConnStatus . . . 185
8.130 connectNumInfo Struct Reference . . . 185
8.130.1 Detailed Description . . . 186
8.130.2 Field Documentation . . . 187
8.130.2.1 callerID . . . 187
8.130.2.2 callerIDLen . . . 187
8.130.2.3 numPlan . . . 187
8.130.2.4 numPresInd . . . 187
8.130.2.5 numType . . . 187
8.130.2.6 screeningInd . . . 187
8.131 CrashInfo Struct Reference . . . 187
8.131.1 Detailed Description . . . 187
8.131.2 Field Documentation . . . 188
8.131.2.1 crashData . . . 188
8.131.2.2 crashId . . . 188
8.131.2.3 crashStrLen . . . 188
8.131.2.4 gcDumpStrLen . . . 188
8.131.2.5 numCrashes . . . 188
8.131.2.6 pCrashString . . . 188
8.131.2.7 pGCDumpString . . . 188
8.132 crashInfoParams Struct Reference . . . 188
8.132.1 Detailed Description . . . 188
8.132.2 Field Documentation . . . 189
8.132.2.1 crashInfo . . . 189
8.132.2.2 crashStatus . . . 189

8.133 CrashInfoParams Struct Reference . . . 189
8.133.1 Detailed Description . . . 189
8.133.2 Field Documentation . . . 189
8.133.2.1 pCrashInfo . . . 189
8.133.2.2 pDevCrashStatus . . . 189
8.134 crashInformation Struct Reference . . . 189
8.134.1 Detailed Description . . . 190
8.134.2 Field Documentation . . . 190
8.134.2.1 crashData . . . 190
8.134.2.2 crashId . . . 190
8.134.2.3 crashString . . . 190
8.134.2.4 crashStrlen . . . 190
8.134.2.5 gcdumpString . . . 190
8.134.2.6 gcdumpStrlen . . . 190
8.134.2.7 numCrashes . . . 190
8.135 CreateProfileIn Struct Reference . . . 190
8.135.1 Detailed Description . . . 191
8.135.2 Field Documentation . . . 191
8.135.2.1 curProfile . . . 191
8.135.2.2 pProfileID . . . 191
8.135.2.3 pProfileType . . . 191
8.136 CreateProfileOut Struct Reference . . . 191
8.136.1 Detailed Description . . . 191
8.136.2 Field Documentation . . . 192
8.136.2.1 pExtErrorCode . . . 192
8.136.2.2 pProfileIndex . . . 192
8.136.2.3 pProfileType . . . 192
8.137 CsgId Struct Reference . . . 192
8.137.1 Detailed Description . . . 192
8.137.2 Field Documentation . . . 192
8.137.2.1 csgId . . . 192
8.137.2.2 TlvPresent . . . 192
8.138 CSGID Struct Reference . . . 192
8.138.1 Detailed Description . . . 193
8.138.2 Field Documentation . . . 193
8.138.2.1 id . . . 193
8.138.2.2 mcc . . . 193
8.138.2.3 mnc . . . 193
8.138.2.4 mncPcsDigits . . . 193
8.138.2.5 rat . . . 193

8.139 CUGInfo Struct Reference . . . 193
8.139.1 Detailed Description . . . 194
8.139.2 Field Documentation . . . 194
8.139.2.1 CUGIndex . . . 194
8.139.2.2 SuppOA . . . 194
8.139.2.3 SuppPrefCUG . . . 194
8.140 curAMRConfig Struct Reference . . . 194
8.140.1 Detailed Description . . . 194
8.140.2 Field Documentation . . . 195
8.140.2.1 gsmAmrStat . . . 195
8.140.2.2 wcdmaAmrStat . . . 195
8.141 CurrDataSysStat Struct Reference . . . 195
8.141.1 Detailed Description . . . 195
8.141.2 Field Documentation . . . 196
8.141.2.1 pCurrNetworkInfo . . . 196
8.141.2.2 pNetworkInfoLen . . . 196
8.141.2.3 pPrefNetwork . . . 196
8.142 currentCatEvent Union Reference . . . 196
8.142.1 Detailed Description . . . 196
8.142.2 Field Documentation . . . 196
8.142.2.1 CatAlphaIdtfr . . . 196
8.142.2.2 CatEndPS . . . 196
8.142.2.3 CatEventLst . . . 196
8.142.2.4 CatEvIDData . . . 196
8.142.2.5 CatRefresh . . . 196
8.143 CurrentImgList Struct Reference . . . 196
8.143.1 Detailed Description . . . 197
8.143.2 Field Documentation . . . 197
8.143.2.1 carrier . . . 197
8.143.2.2 fwvers . . . 197
8.143.2.3 numEntries . . . 197
8.143.2.4 pCurrImgInfo . . . 197
8.143.2.5 pkgver . . . 197
8.143.2.6 priver . . . 197
8.144 currentPLMN Struct Reference . . . 197
8.144.1 Detailed Description . . . 198
8.144.2 Field Documentation . . . 198
8.144.2.1 MCC . . . 198
8.144.2.2 MNC . . . 198
8.144.2.3 netDescr . . . 198

8.144.2.4 netDescrLength . . . 198
8.145 CurrImageInfo Struct Reference . . . 198
8.145.1 Detailed Description . . . 199
8.145.2 Field Documentation . . . 199
8.145.2.1 buildID . . . 199
8.145.2.2 buildIDLen . . . 199
8.145.2.3 imageType . . . 199
8.145.2.4 uniqueID . . . 199
8.146 CurrNetworkInfo Struct Reference . . . 199
8.146.1 Detailed Description . . . 199
8.146.2 Field Documentation . . . 200
8.146.2.1 NetworkType . . . 201
8.146.2.2 RATMask . . . 201
8.146.2.3 SOMask . . . 201
8.147 currNetworkInfo Struct Reference . . . 201
8.147.1 Detailed Description . . . 201
8.147.2 Field Documentation . . . 201
8.147.2.1 NetworkType . . . 201
8.147.2.2 RATMask . . . 201
8.147.2.3 SOMask . . . 201
8.148 custFeaturesInfo Struct Reference . . . 201
8.148.1 Detailed Description . . . 201
8.148.2 Field Documentation . . . 203
8.148.2.1 GpsEnable . . . 203
8.148.2.2 pDHCPRelayEnabled . . . 203
8.148.2.3 pDisableIMSI . . . 203
8.148.2.4 pGPSLPM . . . 203
8.148.2.5 pGPSSel . . . 203
8.148.2.6 pIPFamSupport . . . 203
8.148.2.7 pIsVoiceEnabled . . . 203
8.148.2.8 pRMAutoConnect . . . 203
8.148.2.9 pSMSSupport . . . 204
8.149 custFeaturesSetting Struct Reference . . . 204
8.149.1 Detailed Description . . . 204
8.149.2 Field Documentation . . . 205
8.149.2.1 pDHCPRelayEnabled . . . 205
8.149.2.2 pGPSEnable . . . 205
8.149.2.3 pGPSLPM . . . 205
8.149.2.4 pGPSSel . . . 205
8.149.2.5 pIsVoiceEnabled . . . 205

8.150 custSettingInfo Struct Reference . . . 205
8.150.1 Detailed Description . . . 205
8.150.2 Field Documentation . . . 206
8.150.2.1 cust_attr . . . 206
8.150.2.2 cust_id . . . 206
8.150.2.3 cust_value . . . 206
8.150.2.4 id_length . . . 206
8.150.2.5 value_length . . . 206
8.151 custSettingList Struct Reference . . . 206
8.151.1 Detailed Description . . . 206
8.151.2 Field Documentation . . . 206
8.151.2.1 custSetting . . . 206
8.151.2.2 list_type . . . 207
8.151.2.3 num_instances . . . 207
8.152 dataBearers Struct Reference . . . 207
8.152.1 Detailed Description . . . 207
8.152.2 Field Documentation . . . 207
8.152.2.1 dataBearerMask . . . 207
8.152.2.2 pCurDataBearerTechnology . . . 207
8.152.2.3 pLastCallDataBearerTechnology . . . 207
8.153 DataBearerTech Struct Reference . . . 207
8.153.1 Detailed Description . . . 207
8.153.2 Field Documentation . . . 209
8.153.2.1 ratValue . . . 209
8.153.2.2 soMask . . . 209
8.153.2.3 techType . . . 209
8.154 DataBearerTechExt Struct Reference . . . 209
8.154.1 Detailed Description . . . 210
8.154.2 Field Documentation . . . 210
8.154.2.1 pBearerTech . . . 210
8.154.2.2 pLastBearerTech . . . 210
8.155 dataBearerTechnology Struct Reference . . . 210
8.155.1 Detailed Description . . . 210
8.155.2 Field Documentation . . . 211
8.155.2.1 currentNetwork . . . 211
8.155.2.2 ratMask . . . 211
8.155.2.3 soMask . . . 211
8.156 dataRate Struct Reference . . . 211
8.156.1 Detailed Description . . . 211
8.156.2 Field Documentation . . . 212

8.156.2.1 dataRateMax . . . 212
8.156.2.2 guaranteedRate . . . 212
8.157 dataSrvCapabilities Struct Reference . . . 212
8.157.1 Detailed Description . . . 212
8.157.2 Field Documentation . . . 212
8.157.2.1 dataCapabilities . . . 212
8.157.2.2 dataCapabilitiesLen . . . 213
8.158 DataStatusDetail Struct Reference . . . 213
8.158.1 Detailed Description . . . 213
8.158.2 Field Documentation . . . 214
8.158.2.1 IPAddress . . . 214
8.158.2.2 LastErrCode . . . 214
8.159 DataULongLongTlv Struct Reference . . . 215
8.159.1 Field Documentation . . . 215
8.159.1.1 TlvPresent . . . 215
8.159.1.2 ullData . . . 215
8.160 DataULongTlv Struct Reference . . . 215
8.160.1 Field Documentation . . . 215
8.160.1.1 TlvPresent . . . 215
8.160.1.2 ulData . . . 215
8.161 DcsUsbPortNames Struct Reference . . . 215
8.161.1 Field Documentation . . . 215
8.161.1.1 AtCmdPort . . . 215
8.161.1.2 DmPort . . . 215
8.161.1.3 NmeaPort . . . 215
8.162 delAssistDataStatus Struct Reference . . . 215
8.162.1 Detailed Description . . . 216
8.162.2 Field Documentation . . . 216
8.162.2.1 status . . . 216
8.163 depersonalizationInformation Struct Reference . . . 216
8.163.1 Detailed Description . . . 216
8.163.2 Field Documentation . . . 217
8.163.2.1 ckLen . . . 217
8.163.2.2 ckVal . . . 217
8.163.2.3 feature . . . 217
8.163.2.4 operation . . . 217
8.164 detailSvcInfo Struct Reference . . . 217
8.164.1 Detailed Description . . . 217
8.164.2 Field Documentation . . . 218
8.164.2.1 hdrHybrid . . . 218

8.164.2.2 hdrSrvStatus . . . 218
8.164.2.3 isSysForbidden . . . 218
8.164.2.4 srvCapability . . . 219
8.164.2.5 srvStatus . . . 219
8.165 DeviceConfigDetail Struct Reference . . . 219
8.165.1 Detailed Description . . . 219
8.165.2 Field Documentation . . . 219
8.165.2.1 Chipset . . . 219
8.165.2.2 HWVersion . . . 220
8.165.2.3 QLIC . . . 220
8.165.2.4 Technology . . . 220
8.166 DHCPOption Struct Reference . . . 220
8.166.1 Detailed Description . . . 220
8.166.2 Field Documentation . . . 220
8.166.2.1 optCode . . . 220
8.166.2.2 optValLen . . . 220
8.166.2.3 pOptVal . . . 220
8.167 DHCPOptionList Struct Reference . . . 220
8.167.1 Detailed Description . . . 220
8.167.2 Field Documentation . . . 221
8.167.2.1 numOpt . . . 221
8.167.2.2 pOptions . . . 221
8.168 diagInfo Struct Reference . . . 221
8.168.1 Detailed Description . . . 221
8.168.2 Field Documentation . . . 221
8.168.2.1 diagInfoLen . . . 221
8.168.2.2 diagnosticInfo . . . 221
8.169 dirNum Struct Reference . . . 221
8.169.1 Detailed Description . . . 222
8.169.2 Field Documentation . . . 222
8.169.2.1 dirNum . . . 222
8.169.2.2 dirNumLen . . . 222
8.170 dms_ActivationStatusTlv Struct Reference . . . 222
8.170.1 Detailed Description . . . 222
8.170.2 Field Documentation . . . 222
8.170.2.1 activationStatus . . . 222
8.170.2.2 TlvPresent . . . 223
8.171 dms_OperatingModeTlv Struct Reference . . . 223
8.171.1 Detailed Description . . . 223
8.171.2 Field Documentation . . . 223

8.171.2.1 operatingMode . . . 223
8.171.2.2 TlvPresent . . . 223
8.172 dmsCurrentPRLInfo Struct Reference . . . 223
8.172.1 Detailed Description . . . 223
8.172.2 Field Documentation . . . 224
8.172.2.1 pPRLPreference . . . 224
8.172.2.2 pPRLVersion . . . 224
8.173 DMScustSettingInfo Struct Reference . . . 224
8.173.1 Detailed Description . . . 224
8.173.2 Field Documentation . . . 224
8.173.2.1 cust_attr . . . 224
8.173.2.2 cust_id . . . 224
8.173.2.3 cust_value . . . 224
8.173.2.4 id_length . . . 224
8.173.2.5 value_length . . . 224
8.174 DMScustSettingList Struct Reference . . . 225
8.174.1 Detailed Description . . . 225
8.174.2 Field Documentation . . . 225
8.174.2.1 custSetting . . . 225
8.174.2.2 list_type . . . 225
8.174.2.3 num_instances . . . 225
8.175 DMSgetCustomFeatureV2 Struct Reference . . . 225
8.175.1 Detailed Description . . . 225
8.175.2 Field Documentation . . . 226
8.175.2.1 pCustSettingInfo . . . 226
8.175.2.2 pCustSettingList . . . 226
8.175.2.3 pGetCustomInput . . . 226
8.176 DMSgetCustomInput Struct Reference . . . 226
8.176.1 Detailed Description . . . 226
8.176.2 Field Documentation . . . 226
8.176.2.1 cust_id . . . 226
8.176.2.2 list_type . . . 226
8.177 dmsIndicationRegisterReq Struct Reference . . . 226
8.177.1 Detailed Description . . . 226
8.177.2 Field Documentation . . . 227
8.177.2.1 pSwiGetResetInd . . . 227
8.178 dmsSwiGetResetInfo Struct Reference . . . 227
8.178.1 Detailed Description . . . 227
8.178.2 Field Documentation . . . 227
8.178.2.1 source . . . 228

8.178.2.2 type . . . 228
8.179 Domain Struct Reference . . . 228
8.179.1 Detailed Description . . . 228
8.179.2 Field Documentation . . . 228
8.179.2.1 domainLen . . . 228
8.179.2.2 domainName . . . 228
8.180 DomainNameList Struct Reference . . . 228
8.180.1 Detailed Description . . . 228
8.180.2 Field Documentation . . . 228
8.180.2.1 domain . . . 229
8.180.2.2 numInstances . . . 229
8.181 DRCParams Struct Reference . . . 229
8.181.1 Detailed Description . . . 229
8.181.2 Field Documentation . . . 229
8.181.2.1 DRCCover . . . 229
8.181.2.2 DRCValue . . . 229
8.182 DTMFInfo Struct Reference . . . 229
8.182.1 Detailed Description . . . 229
8.182.2 Field Documentation . . . 230
8.182.2.1 callID . . . 230
8.182.2.2 digitBuff . . . 230
8.182.2.3 digitCnt . . . 230
8.182.2.4 DTMFEvent . . . 230
8.183 DTMFLengths Struct Reference . . . 230
8.183.1 Detailed Description . . . 230
8.183.2 Field Documentation . . . 231
8.183.2.1 DTMFInterdigitInterval . . . 231
8.183.2.2 DTMFPulseWidth . . . 231
8.184 DUNCallInfoInd Struct Reference . . . 231
8.184.1 Field Documentation . . . 231
8.184.1.1 CallEndReason . . . 231
8.184.1.2 ChannelRate . . . 231
8.184.1.3 DataBearerTech . . . 231
8.184.1.4 DormancyStatus . . . 231
8.184.1.5 MdmConnStatus . . . 231
8.184.1.6 RXOKBytesCount . . . 231
8.184.1.7 TXOKBytesCount . . . 231
8.185 dunchannelRate Struct Reference . . . 232
8.185.1 Detailed Description . . . 232
8.185.2 Field Documentation . . . 232

8.185.2.1 CurrChanRxRate . . . 232
8.185.2.2 CurrChanTxRate . . . 232
8.185.2.3 MaxChanRxRate . . . 232
8.185.2.4 MaxChanTxRate . . . 232
8.186 ecioListElement Struct Reference . . . 232
8.186.1 Detailed Description . . . 232
8.186.2 Field Documentation . . . 233
8.186.2.1 ecio . . . 233
8.186.2.2 radioIf . . . 233
8.187 ECIOThresh Struct Reference . . . 233
8.187.1 Detailed Description . . . 233
8.187.2 Field Documentation . . . 233
8.187.2.1 ECIOThresListLen . . . 233
8.187.2.2 pECIOThresList . . . 234
8.188 ECTNum Struct Reference . . . 234
8.188.1 Detailed Description . . . 234
8.188.2 Field Documentation . . . 234
8.188.2.1 ECTCallState . . . 234
8.188.2.2 number . . . 234
8.188.2.3 presentationInd . . . 234
8.189 encryptedPIN1 Struct Reference . . . 234
8.189.1 Detailed Description . . . 234
8.189.2 Field Documentation . . . 235
8.189.2.1 pin1Len . . . 235
8.189.2.2 pin1Val . . . 235
8.190 eriDataparams Struct Reference . . . 235
8.190.1 Field Documentation . . . 235
8.190.1.1 eriData . . . 235
8.190.1.2 eriDataLen . . . 235
8.191 ERIFileparams Struct Reference . . . 235
8.191.1 Detailed Description . . . 235
8.191.2 Field Documentation . . . 236
8.191.2.1 pFile . . . 236
8.191.2.2 pFileSize . . . 236
8.192 errorRateListElement Struct Reference . . . 236
8.192.1 Detailed Description . . . 236
8.192.2 Field Documentation . . . 236
8.192.2.1 errorRate . . . 237
8.192.2.2 radioIf . . . 237
8.193 eTWSPLMNInfoTlv Struct Reference . . . 237

8.193.1 Detailed Description . . . 237
8.193.2 Field Documentation . . . 237
8.193.2.1 ETWSPLMNInfo . . . 237
8.193.2.2 TlvPresent . . . 237
8.194 extDispRecInfo Struct Reference . . . 237
8.194.1 Detailed Description . . . 237
8.194.2 Field Documentation . . . 238
8.194.2.1 dispType . . . 238
8.194.2.2 extDispInfo . . . 238
8.194.2.3 extDispInfoLen . . . 238
8.195 FactorySequenceNumber Struct Reference . . . 238
8.195.1 Detailed Description . . . 238
8.195.2 Field Documentation . . . 238
8.195.2.1 FSNumber . . . 238
8.196 fileAttributes Struct Reference . . . 238
8.196.1 Detailed Description . . . 239
8.196.2 Field Documentation . . . 241
8.196.2.1 fileID . . . 241
8.196.2.2 fileSize . . . 241
8.196.2.3 fileType . . . 241
8.196.2.4 rawLen . . . 241
8.196.2.5 rawValue . . . 241
8.196.2.6 recordCount . . . 241
8.196.2.7 recordSize . . . 241
8.196.2.8 secActivate . . . 241
8.196.2.9 secActivateMask . . . 241
8.196.2.10 secDeactivate . . . 241
8.196.2.11 secDeactivateMask . . . 241
8.196.2.12 secIncrease . . . 241
8.196.2.13 secIncreaseMask . . . 241
8.196.2.14 secRead . . . 241
8.196.2.15 secReadMask . . . 241
8.196.2.16 secWrite . . . 241
8.196.2.17 secWriteMask . . . 241
8.197 fileInfo Struct Reference . . . 241
8.197.1 Detailed Description . . . 242
8.197.2 Field Documentation . . . 242
8.197.2.1 fileID . . . 242
8.197.2.2 path . . . 242
8.197.2.3 pathLen . . . 242

8.198 FirmwareUpdatStat Struct Reference . . . 242
8.198.1 Detailed Description . . . 242
8.198.2 Field Documentation . . . 243
8.198.2.1 pImgType . . . 243
8.198.2.2 pLogString . . . 244
8.198.2.3 pLogStringLen . . . 244
8.198.2.4 pRefData . . . 244
8.198.2.5 pRefString . . . 244
8.198.2.6 pRefStringLen . . . 244
8.198.2.7 ResCode . . . 244
8.199 FMSImageElement Struct Reference . . . 244
8.199.1 Detailed Description . . . 244
8.199.2 Field Documentation . . . 244
8.199.2.1 buildId . . . 244
8.199.2.2 buildIdLength . . . 244
8.199.2.3 imageId . . . 244
8.199.2.4 imageType . . . 244
8.200 FMSImageIdElement Struct Reference . . . 244
8.200.1 Detailed Description . . . 245
8.200.2 Field Documentation . . . 245
8.200.2.1 buildID . . . 245
8.200.2.2 buildIDLength . . . 245
8.200.2.3 failureCount . . . 245
8.200.2.4 imageID . . . 245
8.200.2.5 storageIndex . . . 245
8.201 FMSImageIDEntries Struct Reference . . . 245
8.201.1 Detailed Description . . . 246
8.201.2 Field Documentation . . . 246
8.201.2.1 executingImage . . . 246
8.201.2.2 imageIDElement . . . 246
8.201.2.3 imageIDSize . . . 246
8.201.2.4 imageType . . . 246
8.201.2.5 maxImages . . . 246
8.202 FMSImageList Struct Reference . . . 246
8.202.1 Detailed Description . . . 246
8.202.2 Field Documentation . . . 247
8.202.2.1 imageIDEntries . . . 247
8.202.2.2 listSize . . . 247
8.203 FMSPrefImageList Struct Reference . . . 247
8.203.1 Detailed Description . . . 247

8.203.2 Field Documentation . . . 247
8.203.2.1 listEntries . . . 247
8.203.2.2 listSize . . . 247
8.204 fwinfo_s Struct Reference . . . 247
8.204.1 Detailed Description . . . 247
8.204.2 Field Documentation . . . 248
8.204.2.1 Carrier . . . 248
8.204.2.2 FirmwareID . . . 248
8.204.2.3 GPSCapability . . . 248
8.204.2.4 Region . . . 248
8.204.2.5 Technology . . . 248
8.205 GERANInfo Struct Reference . . . 248
8.205.1 Detailed Description . . . 248
8.205.2 Field Documentation . . . 249
8.205.2.1 arfcn . . . 249
8.205.2.2 bsic . . . 249
8.205.2.3 cellID . . . 249
8.205.2.4 insNmrCellInfo . . . 250
8.205.2.5 lac . . . 250
8.205.2.6 nmrInst . . . 250
8.205.2.7 plmn . . . 250
8.205.2.8 rxLev . . . 250
8.205.2.9 timingAdvance . . . 250
8.206 geranInstInfo Struct Reference . . . 250
8.206.1 Detailed Description . . . 250
8.206.2 Field Documentation . . . 250
8.206.2.1 geranArfcn . . . 250
8.206.2.2 geranBsicBcc . . . 250
8.206.2.3 geranBsicNcc . . . 250
8.206.2.4 geranRssi . . . 250
8.207 getAllCallInformation Struct Reference . . . 251
8.207.1 Detailed Description . . . 251
8.207.2 Field Documentation . . . 251
8.207.2.1 ALS . . . 251
8.207.2.2 CallInfo . . . 251
8.207.2.3 isEmpty . . . 251
8.208 getAllCallRmtPtyName Struct Reference . . . 251
8.208.1 Detailed Description . . . 251
8.208.2 Field Documentation . . . 252
8.208.2.1 callID . . . 252

8.208.2.2 RemotePartyName . . . 252
8.209 getAllCallRmtPtyNum Struct Reference . . . 252
8.209.1 Detailed Description . . . 252
8.209.2 Field Documentation . . . 252
8.209.2.1 callID . . . 252
8.209.2.2 RemotePartyNum . . . 252
8.210 GetAudioPathConfigReq Struct Reference . . . 252
8.210.1 Detailed Description . . . 252
8.210.2 Field Documentation . . . 253
8.210.2.1 Item . . . 253
8.210.2.2 Profile . . . 253
8.211 GetAudioPathConfigResp Struct Reference . . . 253
8.211.1 Detailed Description . . . 253
8.211.2 Field Documentation . . . 254
8.211.2.1 pCodecSTGain . . . 255
8.211.2.2 pDTMFTXGain . . . 255
8.211.2.3 pECMode . . . 255
8.211.2.4 pMICGainSelect . . . 255
8.211.2.5 pNSEnable . . . 255
8.211.2.6 pRXAGCList . . . 255
8.211.2.7 pRXAVCAGCSwitch . . . 255
8.211.2.8 pRXAVCList . . . 255
8.211.2.9 pRXPCMIIRFltr . . . 255
8.211.2.10 pTXAGCList . . . 255
8.211.2.11 pTXAVCSwitch . . . 255
8.211.2.12 pTXGain . . . 255
8.211.2.13 pTXPCMIIRFltr . . . 255
8.212 GetAudioProfileReq Struct Reference . . . 255
8.212.1 Detailed Description . . . 255
8.212.2 Field Documentation . . . 255
8.212.2.1 Generator . . . 255
8.213 GetAudioProfileResp Struct Reference . . . 255
8.213.1 Detailed Description . . . 256
8.213.2 Field Documentation . . . 256
8.213.2.1 EarMute . . . 256
8.213.2.2 MicMute . . . 256
8.213.2.3 Profile . . . 256
8.213.2.4 Volume . . . 256
8.214 GetAudioVoITLBConfigReq Struct Reference . . . 257
8.214.1 Detailed Description . . . 257

8.214.2 Field Documentation . . . 257
8.214.2.1 Generator . . . 257
8.214.2.2 Item . . . 257
8.214.2.3 Profile . . . 257
8.214.2.4 Volume . . . 257
8.215 GetAudioVolTLBConfigResp Struct Reference . . . 257
8.215.1 Detailed Description . . . 258
8.215.2 Field Documentation . . . 258
8.215.2.1 ResCode . . . 258
8.216 getCallFWExtInfo Struct Reference . . . 258
8.216.1 Detailed Description . . . 258
8.216.2 Field Documentation . . . 258
8.216.2.1 CallFWExtInfo . . . 258
8.216.2.2 numInstances . . . 258
8.217 getCallFWInfo Struct Reference . . . 258
8.217.1 Detailed Description . . . 258
8.217.2 Field Documentation . . . 259
8.217.2.1 CallFWInfo . . . 259
8.217.2.2 numInstances . . . 259
8.218 getCustomFeatureV2 Struct Reference . . . 259
8.218.1 Detailed Description . . . 259
8.218.2 Field Documentation . . . 259
8.218.2.1 pCustSettingInfo . . . 259
8.218.2.2 pCustSettingList . . . 259
8.218.2.3 pGetCustomInput . . . 259
8.219 getCustomInput Struct Reference . . . 260
8.219.1 Detailed Description . . . 260
8.219.2 Field Documentation . . . 260
8.219.2.1 cust_id . . . 260
8.219.2.2 list_type . . . 260
8.220 getDUNCallInfoReq Struct Reference . . . 260
8.220.1 Detailed Description . . . 260
8.220.2 Field Documentation . . . 261
8.220.2.1 Mask . . . 261
8.220.2.2 pReportChannelRate . . . 261
8.220.2.3 pReportConnStatus . . . 261
8.220.2.4 pReportDataBearerTech . . . 261
8.220.2.5 pReportDormStatus . . . 261
8.220.2.6 pTransferStatInd . . . 261
8.221 getDUNCallInfoResp Struct Reference . . . 262

8.221.1 Detailed Description . . . 262
8.221.2 Field Documentation . . . 264
8.221.2.1 pCallEndReason . . . 264
8.221.2.2 pChannelRate . . . 264
8.221.2.3 pConnectionStatus . . . 264
8.221.2.4 pDataBearerTech . . . 264
8.221.2.5 pDormancyStatus . . . 264
8.221.2.6 pLastCallDataBearerTech . . . 264
8.221.2.7 pLastCallRXOKBytesCnt . . . 264
8.221.2.8 pLastCallTXOKBytesCnt . . . 264
8.221.2.9 pMdmCallDurationActive . . . 264
8.221.2.10 pRXOKBytesCount . . . 264
8.221.2.11 pTXOKBytesCount . . . 264
8.222 getDyingGaspCfg Struct Reference . . . 264
8.222.1 Detailed Description . . . 265
8.222.2 Field Documentation . . . 265
8.222.2.1 pDestSMSContent . . . 265
8.222.2.2 pDestSMSNum . . . 265
8.223 getDyingGaspStatistics Struct Reference . . . 265
8.223.1 Detailed Description . . . 265
8.223.2 Field Documentation . . . 265
8.223.2.1 pSMSAttemptedFlag . . . 265
8.223.2.2 pTimeStamp . . . 266
8.224 GetErrRateResp Struct Reference . . . 266
8.224.1 Detailed Description . . . 266
8.224.2 Field Documentation . . . 266
8.224.2.1 pCDMAFrameErrRate . . . 266
8.224.2.2 pGSMBER . . . 266
8.224.2.3 pHDRPackErrRate . . . 266
8.224.2.4 pWCDMABER . . . 266
8.225 GetHRPDStatsResp Struct Reference . . . 266
8.225.1 Detailed Description . . . 267
8.225.2 Field Documentation . . . 267
8.225.2.1 pDRCParams . . . 267
8.225.2.2 pPilotSetData . . . 267
8.225.2.3 pUATI . . . 267
8.226 GetIMSSMSConfigParams Struct Reference . . . 267
8.226.1 Detailed Description . . . 267
8.226.2 Field Documentation . . . 268
8.226.2.1 pPhoneCtxtURI . . . 268

8.226.2.2 pPhoneCtxtURILen . . . 268
8.226.2.3 pSettingResp . . . 268
8.226.2.4 pSMSFormat . . . 268
8.226.2.5 pSMSOverIPNwInd . . . 268
8.227 GetIMSUserConfigParams Struct Reference . . . 268
8.227.1 Detailed Description . . . 268
8.227.2 Field Documentation . . . 269
8.227.2.1 pIMSDomain . . . 269
8.227.2.2 pIMSDomainLen . . . 269
8.227.2.3 pSettingResp . . . 269
8.228 GetIMSVoIPConfigResp Struct Reference . . . 269
8.228.1 Detailed Description . . . 269
8.228.2 Field Documentation . . . 271
8.228.2.1 pAmrMode . . . 271
8.228.2.2 pAmrOctetAligned . . . 271
8.228.2.3 pAmrWbEnable . . . 271
8.228.2.4 pAmrWBMode . . . 271
8.228.2.5 pAmrWBOctetAligned . . . 271
8.228.2.6 pMinSessionExpiryTimer . . . 271
8.228.2.7 pRingBackTimer . . . 271
8.228.2.8 pRingingTimer . . . 271
8.228.2.9 pRTPRTCPInactTimer . . . 271
8.228.2.10 pScrAmrEnable . . . 271
8.228.2.11 pScrAmrWbEnable . . . 271
8.228.2.12 pSessionExpiryTimer . . . 271
8.228.2.13 pSettingResp . . . 271
8.229 GetInstIDResp Struct Reference . . . 271
8.229.1 Field Documentation . . . 271
8.229.1.1 pInstanceID . . . 271
8.229.1.2 pIPFamily . . . 272
8.230 GetM2MAudioProfileReq Struct Reference . . . 272
8.230.1 Detailed Description . . . 272
8.230.2 Field Documentation . . . 272
8.230.2.1 pGenerator . . . 272
8.231 GetM2MAudioProfileResp Struct Reference . . . 272
8.231.1 Detailed Description . . . 272
8.231.2 Field Documentation . . . 273
8.231.2.1 CwtMute . . . 273
8.231.2.2 EarMute . . . 273
8.231.2.3 Generator . . . 273

8.231.2.4 MicMute . . . 273
8.231.2.5 Profile . . . 273
8.231.2.6 Volume . . . 273
8.232 GetM2MAudioVolumeReq Struct Reference . . . 273
8.232.1 Detailed Description . . . 273
8.232.2 Field Documentation . . . 274
8.232.2.1 Generator . . . 274
8.232.2.2 Profile . . . 274
8.233 GetM2MAudioVolumeResp Struct Reference . . . 274
8.233.1 Detailed Description . . . 274
8.233.2 Field Documentation . . . 274
8.233.2.1 Level . . . 274
8.234 GetM2MAVMuteReq Struct Reference . . . 274
8.234.1 Detailed Description . . . 274
8.234.2 Field Documentation . . . 274
8.234.2.1 Profile . . . 275
8.235 GetM2MAVMuteResp Struct Reference . . . 275
8.235.1 Detailed Description . . . 275
8.235.2 Field Documentation . . . 275
8.235.2.1 CwtMute . . . 275
8.235.2.2 EarMute . . . 275
8.235.2.3 MicMute . . . 275
8.236 GetM2MSpkrGainReq Struct Reference . . . 275
8.236.1 Detailed Description . . . 275
8.236.2 Field Documentation . . . 276
8.236.2.1 Profile . . . 276
8.237 GetM2MSpkrGainResp Struct Reference . . . 276
8.237.1 Detailed Description . . . 276
8.237.2 Field Documentation . . . 276
8.237.2.1 Value . . . 276
8.238 getMsgWaitingInfo Struct Reference . . . 276
8.238.1 Detailed Description . . . 276
8.238.2 Field Documentation . . . 277
8.238.2.1 msgWaitInfo . . . 277
8.238.2.2 numInstances . . . 277
8.239 GetNetworkTimeResp Struct Reference . . . 277
8.239.1 Detailed Description . . . 277
8.239.2 Field Documentation . . . 277
8.239.2.1 p3GPP2TimeInfo . . . 277
8.239.2.2 p3GPPTimeInfo . . . 277

8.240 GetRegMgrConfigParams Struct Reference . . . 277
8.240.1 Detailed Description . . . 277
8.240.2 Field Documentation . . . 278
8.240.2.1 pIMSTestMode . . . 278
8.240.2.2 pPCSCFPort . . . 278
8.240.2.3 pPriCSCFPortName . . . 278
8.240.2.4 pPriCSCFPortNameLen . . . 278
8.240.2.5 pSettingResp . . . 278
8.241 GetSessionIDResp Struct Reference . . . 278
8.241.1 Field Documentation . . . 278
8.241.1.1 pSessionIDv4 . . . 278
8.241.1.2 pSessionIDv6 . . . 278
8.242 GetSIPConfigResp Struct Reference . . . 278
8.242.1 Detailed Description . . . 279
8.242.2 Field Documentation . . . 279
8.242.2.1 pSettingResp . . . 279
8.242.2.2 pSigCompEnabled . . . 279
8.242.2.3 pSIPLocalPort . . . 279
8.242.2.4 pSubscribeTimer . . . 279
8.242.2.5 pTimerSIPReg . . . 279
8.242.2.6 pTimerT1 . . . 279
8.242.2.7 pTimerT2 . . . 279
8.242.2.8 pTimerTf . . . 279
8.243 GnssData Struct Reference . . . 280
8.243.1 Detailed Description . . . 280
8.243.2 Field Documentation . . . 281
8.243.2.1 mask . . . 281
8.244 gnssSvInfoNotification Struct Reference . . . 281
8.244.1 Detailed Description . . . 282
8.244.2 Field Documentation . . . 282
8.244.2.1 bAltitudeAssumed . . . 282
8.244.2.2 pSatelliteInfo . . . 282
8.245 GPRSQoS Struct Reference . . . 282
8.245.1 Detailed Description . . . 282
8.245.2 Field Documentation . . . 283
8.245.2.1 delayClass . . . 283
8.245.2.2 meanThroughputClass . . . 283
8.245.2.3 peakThroughputClass . . . 283
8.245.2.4 precedenceClass . . . 283
8.245.2.5 reliabilityClass . . . 283

8.246 GPRSRequestedQoS Struct Reference . . . 283
8.246.1 Detailed Description . . . 283
8.246.2 Field Documentation . . . 284
8.246.2.1 delayClass . . . 284
8.246.2.2 meanThroughputClass . . . 284
8.246.2.3 peakThroughputClass . . . 284
8.246.2.4 precedenceClass . . . 284
8.246.2.5 reliabilityClass . . . 284
8.247 GPSStateInfo Struct Reference . . . 284
8.247.1 Detailed Description . . . 284
8.247.2 Field Documentation . . . 287
8.247.2.1 Altitude . . . 287
8.247.2.2 EngineState . . . 287
8.247.2.3 glo_almanac_sv_msk . . . 287
8.247.2.4 glo_ephemeris_sv_msk . . . 287
8.247.2.5 glo_health_sv_msk . . . 287
8.247.2.6 glo_visible_sv_msk . . . 287
8.247.2.7 gps_almanac_sv_msk . . . 287
8.247.2.8 gps_ephemeris_sv_msk . . . 287
8.247.2.9 gps_health_sv_msk . . . 287
8.247.2.10 gps_visible_sv_msk . . . 287
8.247.2.11 HorizontalUncertainty . . . 287
8.247.2.12 iono_valid . . . 287
8.247.2.13 Latitude . . . 287
8.247.2.14 Longitude . . . 287
8.247.2.15 sbas_almanac_sv_msk . . . 287
8.247.2.16 sbas_ephemeris_sv_msk . . . 287
8.247.2.17 sbas_health_sv_msk . . . 288
8.247.2.18 sbas_visible_sv_msk . . . 288
8.247.2.19 Time_uncert_ms . . . 288
8.247.2.20 TimeStmp_gps_week . . . 288
8.247.2.21 TimeStmp_tow_ms . . . 288
8.247.2.22 ValidMask . . . 288
8.247.2.23 VerticalUncertainty . . . 288
8.247.2.24 xtra_start_gps_minutes . . . 288
8.247.2.25 xtra_start_gps_week . . . 288
8.247.2.26 xtra_valid_duration_hours . . . 288
8.248 gpsTime_s Struct Reference . . . 288
8.248.1 Detailed Description . . . 288
8.248.2 Field Documentation . . . 288

8.248.2.1 gpsTimeOfWeekMs . . . 288
8.248.2.2 gpsWeek . . . 288
8.249 gsmCellInfo Struct Reference . . . 288
8.249.1 Detailed Description . . . 289
8.249.2 Field Documentation . . . 289
8.249.2.1 arfcn . . . 289
8.249.2.2 band1900 . . . 289
8.249.2.3 bsicId . . . 289
8.249.2.4 cellIdValid . . . 289
8.249.2.5 rssi . . . 289
8.249.2.6 srxlev . . . 289
8.250 GSMRSSIThresh Struct Reference . . . 290
8.250.1 Detailed Description . . . 290
8.250.2 Field Documentation . . . 290
8.250.2.1 GSMRSSIThreshListLen . . . 290
8.250.2.2 pGSMRSSIThreshList . . . 290
8.251 GSMSrvStatusInfo Struct Reference . . . 290
8.251.1 Detailed Description . . . 290
8.251.2 Field Documentation . . . 291
8.251.2.1 isPrefDataPath . . . 291
8.251.2.2 srvStatus . . . 291
8.251.2.3 trueSrvStatus . . . 291
8.252 GSMSysInfo Struct Reference . . . 291
8.252.1 Detailed Description . . . 291
8.252.2 Field Documentation . . . 293
8.252.2.1 cellId . . . 293
8.252.2.2 cellIdValid . . . 293
8.252.2.3 dtmSupp . . . 293
8.252.2.4 dtmSuppValid . . . 293
8.252.2.5 egprsSupp . . . 293
8.252.2.6 egprsSuppValid . . . 293
8.252.2.7 lac . . . 294
8.252.2.8 lacValid . . . 294
8.252.2.9 MCC . . . 294
8.252.2.10 MNC . . . 294
8.252.2.11 networkIdValid . . . 294
8.252.2.12 regRejectInfoValid . . . 294
8.252.2.13 rejCause . . . 294
8.252.2.14 rejectSrvDomain . . . 294
8.252.2.15 sysInfoGSM . . . 294

8.253 gyroAcceptReady_s Struct Reference . . . 294
8.253.1 Detailed Description . . . 294
8.253.2 Field Documentation . . . 294
8.253.2.1 batchPerSec . . . 294
8.253.2.2 injectEnable . . . 295
8.253.2.3 samplesPerBatch . . . 295
8.254 gyroTempAcceptReady_s Struct Reference . . . 295
8.254.1 Detailed Description . . . 295
8.254.2 Field Documentation . . . 295
8.254.2.1 batchPerSec . . . 295
8.254.2.2 injectEnable . . . 295
8.254.2.3 samplesPerBatch . . . 295
8.255 HDRECIOThresh Struct Reference . . . 295
8.255.1 Detailed Description . . . 296
8.255.2 Field Documentation . . . 296
8.255.2.1 HDRECIOThreshListLen . . . 296
8.255.2.2 pHDRECIOThreshList . . . 296
8.256 HDRIOThresh Struct Reference . . . 296
8.256.1 Detailed Description . . . 296
8.256.2 Field Documentation . . . 296
8.256.2.1 HDRIOThreshListLen . . . 296
8.256.2.2 pHDRIOThreshList . . . 296
8.257 HDRPersonalityInd Struct Reference . . . 296
8.257.1 Field Documentation . . . 297
8.257.1.1 pCurrentPersonality . . . 297
8.257.1.2 pPersonalityListLength . . . 297
8.257.1.3 pProtocolSubtypeElement . . . 297
8.258 HDRPersonalityResp Struct Reference . . . 297
8.258.1 Detailed Description . . . 297
8.258.2 Field Documentation . . . 297
8.258.2.1 pCurrentPersonality . . . 297
8.258.2.2 pPersonalityListLength . . . 297
8.258.2.3 pProtocolSubtypeElement . . . 297
8.259 HDRProtSubtypResp Struct Reference . . . 297
8.259.1 Detailed Description . . . 298
8.259.2 Field Documentation . . . 298
8.259.2.1 pAppSubType . . . 298
8.259.2.2 pCurrentPrsnlty . . . 298
8.259.2.3 pPersonalityListLength . . . 298
8.259.2.4 pProtoSubTypElmnt . . . 298

8.260 HDRRSSIThresh Struct Reference . . . 298
8.260.1 Detailed Description . . . 298
8.260.2 Field Documentation . . . 299
8.260.2.1 HDRRSSIThreshListLen . . . 299
8.260.2.2 pHDRRSSIThreshList . . . 299
8.261 HDRSINRThresh Struct Reference . . . 299
8.261.1 Detailed Description . . . 299
8.261.2 Field Documentation . . . 299
8.261.2.1 HDRSINRThresListLen . . . 299
8.261.2.2 pHDRSINRThresList . . . 299
8.262 HDRSINRThreshold Struct Reference . . . 299
8.262.1 Detailed Description . . . 300
8.262.2 Field Documentation . . . 300
8.262.2.1 HDRSINRThreshListLen . . . 300
8.262.2.2 pHDRSINRThreshList . . . 300
8.263 HDRSSInfo Struct Reference . . . 300
8.263.1 Detailed Description . . . 300
8.263.2 Field Documentation . . . 301
8.263.2.1 ecio . . . 301
8.263.2.2 io . . . 301
8.263.2.3 rssi . . . 301
8.263.2.4 sinr . . . 301
8.264 hdrSSInfo Struct Reference . . . 301
8.264.1 Detailed Description . . . 301
8.264.2 Field Documentation . . . 302
8.264.2.1 ecio . . . 302
8.264.2.2 io . . . 302
8.264.2.3 rssi . . . 302
8.264.2.4 sinr . . . 302
8.265 HDRSysInfo Struct Reference . . . 302
8.265.1 Detailed Description . . . 302
8.265.2 Field Documentation . . . 304
8.265.2.1 hdrActiveProt . . . 304
8.265.2.2 hdrActiveProtValid . . . 304
8.265.2.3 hdrPersonality . . . 304
8.265.2.4 hdrPersonalityValid . . . 304
8.265.2.5 is856SysId . . . 304
8.265.2.6 is856SysIdValid . . . 304
8.265.2.7 isSysPrlMatch . . . 304
8.265.2.8 isSysPrlMatchValid . . . 304

8.265.2.9 sysInfoHDR . . . 304
8.266 homeSIDNID Struct Reference . . . 304
8.266.1 Detailed Description . . . 304
8.266.2 Field Documentation . . . 304
8.266.2.1 numInstances . . . 304
8.266.2.2 SidNid . . . 304
8.267 hotSwapStatus Struct Reference . . . 304
8.267.1 Detailed Description . . . 305
8.267.2 Field Documentation . . . 305
8.267.2.1 hotSwap . . . 305
8.267.2.2 hotSwapLength . . . 305
8.268 image_info_t Struct Reference . . . 305
8.268.1 Field Documentation . . . 305
8.268.1.1 buildID . . . 305
8.268.1.2 buildIDLen . . . 305
8.268.1.3 imageType . . . 305
8.268.1.4 uniqueID . . . 305
8.269 ImageElement Struct Reference . . . 305
8.269.1 Detailed Description . . . 306
8.269.2 Field Documentation . . . 306
8.269.2.1 buildId . . . 306
8.269.2.2 buildIdLength . . . 306
8.269.2.3 imageId . . . 306
8.269.2.4 imageType . . . 306
8.270 ImageIdElement Struct Reference . . . 306
8.270.1 Detailed Description . . . 306
8.270.2 Field Documentation . . . 307
8.270.2.1 buildID . . . 307
8.270.2.2 buildIDLength . . . 307
8.270.2.3 failureCount . . . 307
8.270.2.4 imageID . . . 307
8.270.2.5 storageIndex . . . 307
8.271 ImageIDEntries Struct Reference . . . 307
8.271.1 Detailed Description . . . 307
8.271.2 Field Documentation . . . 308
8.271.2.1 executingImage . . . 308
8.271.2.2 imageIDElement . . . 308
8.271.2.3 imageIDSize . . . 308
8.271.2.4 imageType . . . 308
8.271.2.5 maxImages . . . 308

8.272 ImageList Struct Reference . . . 308
8.272.1 Detailed Description . . . 308
8.272.2 Field Documentation . . . 308
8.272.2.1 imageIDEntries . . . 308
8.272.2.2 listSize . . . 308
8.273 IMSAIndRegisterInfo Struct Reference . . . 308
8.273.1 Detailed Description . . . 309
8.273.2 Field Documentation . . . 309
8.273.2.1 pPdpStatusConfig . . . 309
8.273.2.2 pRatHandoverStatusConfig . . . 309
8.273.2.3 pRegStatusConfig . . . 309
8.273.2.4 pServiceStatusConfig . . . 309
8.274 imsaPdpStatusInfo Struct Reference . . . 309
8.274.1 Detailed Description . . . 310
8.274.2 Field Documentation . . . 310
8.274.2.1 connetionState . . . 310
8.274.2.2 pFailErrorCode . . . 310
8.275 imsaRatStatusInfo Struct Reference . . . 310
8.275.1 Detailed Description . . . 310
8.275.2 Field Documentation . . . 311
8.275.2.1 pErrorCodeStr . . . 311
8.275.2.2 pRATStatus . . . 311
8.275.2.3 pSrcRAT . . . 311
8.275.2.4 pTgtRAT . . . 311
8.276 IMSARegistrationStatus Struct Reference . . . 311
8.276.1 Detailed Description . . . 311
8.276.2 Field Documentation . . . 312
8.276.2.1 pImsRegErrCode . . . 312
8.276.2.2 pImsRegStatus . . . 312
8.276.2.3 pNewImsRegStatus . . . 312
8.277 imsaRegStatusInfo Struct Reference . . . 312
8.277.1 Detailed Description . . . 312
8.277.2 Field Documentation . . . 312
8.277.2.1 pbIMSRegistered . . . 312
8.277.2.2 pImsRegStatus . . . 312
8.277.2.3 pRegStatusErrorCode . . . 312
8.278 IMSAServiceStatus Struct Reference . . . 312
8.278.1 Detailed Description . . . 313
8.278.2 Field Documentation . . . 314
8.278.2.1 pSmsServiceRat . . . 314

8.278.2.2 pSmsServiceStatus . . . 314
8.278.2.3 pUtServiceRat . . . 314
8.278.2.4 pUtServiceStatus . . . 314
8.278.2.5 pVoipServiceRat . . . 314
8.278.2.6 pVoipServiceStatus . . . 314
8.278.2.7 pVsServiceRat . . . 314
8.278.2.8 pVsServiceStatus . . . 314
8.278.2.9 pVtServiceRat . . . 314
8.278.2.10 pVtServiceStatus . . . 314
8.279 IMSASupportedFieldsResp Struct Reference . . . 314
8.279.1 Detailed Description . . . 315
8.279.2 Field Documentation . . . 315
8.279.2.1 pIndFieldsList . . . 315
8.279.2.2 pReqFieldsList . . . 315
8.279.2.3 pRespFieldsList . . . 315
8.280 IMSASupportedMsgInfo Struct Reference . . . 315
8.280.1 Detailed Description . . . 315
8.280.2 Field Documentation . . . 315
8.280.2.1 pSupportedMsgList . . . 315
8.281 imsaSvcStatusInfo Struct Reference . . . 315
8.281.1 Detailed Description . . . 316
8.281.2 Field Documentation . . . 316
8.281.2.1 pSMSSvcRAT . . . 316
8.281.2.2 pSMSSvcStatus . . . 316
8.281.2.3 pUTSvcRAT . . . 316
8.281.2.4 pUTSvcStatus . . . 316
8.281.2.5 pVOIPSvcRAT . . . 316
8.281.2.6 pVOIPSvcStatus . . . 316
8.281.2.7 pVTSvcRAT . . . 316
8.281.2.8 pVTSvcStatus . . . 316
8.282 imsCfgIndRegisterInfo Struct Reference . . . 316
8.282.1 Detailed Description . . . 317
8.282.2 Field Documentation . . . 317
8.282.2.1 pRegMgrConfigEvents . . . 317
8.282.2.2 pSIPConfigEvents . . . 317
8.282.2.3 pSMSConfigEvents . . . 317
8.282.2.4 pUserConfigEvents . . . 318
8.282.2.5 pVoIPConfigEvents . . . 318
8.283 imsRegMgrConfigInfo Struct Reference . . . 318
8.283.1 Detailed Description . . . 318

8.283.2 Field Documentation . . . 318
8.283.2.1 pCSCFPortName . . . 318
8.283.2.2 pIMSTestMode . . . 318
8.283.2.3 pPriCSCFPort . . . 318
8.284 imsSIPConfigInfo Struct Reference . . . 318
8.284.1 Detailed Description . . . 319
8.284.2 Field Documentation . . . 319
8.284.2.1 pSigCompEnabled . . . 319
8.284.2.2 pSIPLocalPort . . . 319
8.284.2.3 pSubscribeTimer . . . 319
8.284.2.4 pTimerSIPReg . . . 319
8.284.2.5 pTimerT1 . . . 319
8.284.2.6 pTimerT2 . . . 319
8.284.2.7 pTimerTf . . . 319
8.285 imsSMSConfigInfo Struct Reference . . . 319
8.285.1 Detailed Description . . . 320
8.285.2 Field Documentation . . . 320
8.285.2.1 pPhoneCtxtURI . . . 320
8.285.2.2 pSMSFormat . . . 320
8.285.2.3 pSMSOverIPNwInd . . . 320
8.286 imsUserConfigInfo Struct Reference . . . 320
8.286.1 Detailed Description . . . 320
8.286.2 Field Documentation . . . 320
8.286.2.1 pIMSDomain . . . 320
8.287 imsVoIPConfigInfo Struct Reference . . . 321
8.287.1 Detailed Description . . . 321
8.287.2 Field Documentation . . . 323
8.287.2.1 pAmrMode . . . 323
8.287.2.2 pAmrOctetAligned . . . 323
8.287.2.3 pAmrWbEnable . . . 323
8.287.2.4 pAmrWBMode . . . 323
8.287.2.5 pAmrWBOctetAligned . . . 323
8.287.2.6 pMinSessionExpiryTimer . . . 323
8.287.2.7 pRingBackTimer . . . 323
8.287.2.8 pRingingTimer . . . 323
8.287.2.9 pRTPRTCPInactTimer . . . 323
8.287.2.10 pScrAmrEnable . . . 323
8.287.2.11 pScrAmrWbEnable . . . 323
8.287.2.12 pSessionExpiryTimer . . . 323
8.288 IndFieldsList Struct Reference . . . 323

8.288.1 Detailed Description . . . 323
8.288.2 Field Documentation . . . 323
8.288.2.1 indicationFields . . . 323
8.288.2.2 indicationFieldsLen . . . 323
8.289 infoInterFreq Struct Reference . . . 324
8.289.1 Detailed Description . . . 324
8.289.2 Field Documentation . . . 324
8.289.2.1 cell_resel_priority . . . 324
8.289.2.2 cellInterFreqParams . . . 324
8.289.2.3 cells_len . . . 324
8.289.2.4 earfcn . . . 325
8.289.2.5 threshXHigh . . . 325
8.289.2.6 threshXLow . . . 325
8.290 IOThresh Struct Reference . . . 325
8.290.1 Detailed Description . . . 325
8.290.2 Field Documentation . . . 325
8.290.2.1 IOThresListLen . . . 325
8.290.2.2 pIOThresList . . . 325
8.291 IPv4Addr Struct Reference . . . 325
8.291.1 Detailed Description . . . 325
8.291.2 Field Documentation . . . 326
8.291.2.1 addr . . . 326
8.291.2.2 subnetMask . . . 326
8.292 IPv6Addr Struct Reference . . . 326
8.292.1 Detailed Description . . . 326
8.292.2 Field Documentation . . . 326
8.292.2.1 addr . . . 326
8.292.2.2 prefixLen . . . 326
8.293 IPV6AddressInfo Struct Reference . . . 326
8.293.1 Detailed Description . . . 326
8.293.2 Field Documentation . . . 327
8.293.2.1 IPAddressV6 . . . 327
8.293.2.2 IPV6PrefixLen . . . 327
8.294 ipv6AddressInfo Struct Reference . . . 327
8.294.1 Detailed Description . . . 327
8.294.2 Field Documentation . . . 327
8.294.2.1 IPAddressV6 . . . 327
8.294.2.2 IPV6PrefixLen . . . 327
8.295 IPV6GWAddressInfo Struct Reference . . . 327
8.295.1 Detailed Description . . . 327

8.295.2 Field Documentation . . . 328
8.295.2.1 gwAddressV6 . . . 328
8.295.2.2 gwV6PrefixLen . . . 328
8.296 IPv6TrafCls Struct Reference . . . 328
8.296.1 Detailed Description . . . 328
8.296.2 Field Documentation . . . 328
8.296.2.1 mask . . . 328
8.296.2.2 val . . . 328
8.297 LibPackGPRSRequestedQoS Struct Reference . . . 328
8.297.1 Detailed Description . . . 329
8.297.2 Field Documentation . . . 329
8.297.2.1 delayClass . . . 329
8.297.2.2 meanThroughputClass . . . 329
8.297.2.3 peakThroughputClass . . . 329
8.297.2.4 precedenceClass . . . 329
8.297.2.5 reliabilityClass . . . 329
8.298 LibpackProfile3GPP Struct Reference . . . 329
8.298.1 Detailed Description . . . 330
8.298.2 Field Documentation . . . 334
8.298.2.1 pAddrAllocPref . . . 334
8.298.2.2 pAPNClass . . . 334
8.298.2.3 pAPNDisabledFlag . . . 334
8.298.2.4 pAPNName . . . 334
8.298.2.5 pAPNnameSize . . . 334
8.298.2.6 pAuthenticationPref . . . 334
8.298.2.7 pGPRSMinimumQoS . . . 334
8.298.2.8 pGPRSRequestedQos . . . 334
8.298.2.9 pImCnFlag . . . 334
8.298.2.10 pIPv4AddrPref . . . 334
8.298.2.11 pIPv6AddPref . . . 334
8.298.2.12 pPassword . . . 334
8.298.2.13 pPasswordSize . . . 334
8.298.2.14 pPcscfAddrUsingDhcp . . . 334
8.298.2.15 pPcscfAddrUsingPCO . . . 334
8.298.2.16 pPDNInactivTimeout . . . 334
8.298.2.17 pPdpAccessConFlag . . . 334
8.298.2.18 pPdpContext . . . 334
8.298.2.19 pPdpDataCompType . . . 334
8.298.2.20 pPdpHdrCompType . . . 334
8.298.2.21 pPDPtype . . . 334

8.298.2.22 pPriDNSIPv4AddPref . . . 334
8.298.2.23 pPriDNSIPv6addpref . . . 334
8.298.2.24 pPrimaryID . . . 335
8.298.2.25 pProfilename . . . 335
8.298.2.26 pProfilenameSize . . . 335
8.298.2.27 pQosClassID . . . 335
8.298.2.28 pSecDNSIPv4AddPref . . . 335
8.298.2.29 pSecDNSIPv6addpref . . . 335
8.298.2.30 pSecondaryFlag . . . 335
8.298.2.31 pTFTID1Params . . . 335
8.298.2.32 pTFTID2Params . . . 335
8.298.2.33 pUMTSMinQoS . . . 335
8.298.2.34 pUMTSMinQosSigInd . . . 335
8.298.2.35 pUMTSReqQoS . . . 335
8.298.2.36 pUMTSReqQoSSigInd . . . 335
8.298.2.37 pUsername . . . 335
8.298.2.38 pUsernameSize . . . 335
8.299 LibpackProfile3GPP2 Struct Reference . . . 335
8.299.1 Detailed Description . . . 336
8.299.2 Field Documentation . . . 339
8.299.2.1 pAllowLinger . . . 339
8.299.2.2 pAPNClass3GPP2 . . . 339
8.299.2.3 pAPNEnabled3GPP2 . . . 339
8.299.2.4 pApnString . . . 339
8.299.2.5 pApnStringSize . . . 339
8.299.2.6 pAppPriority . . . 339
8.299.2.7 pAppType . . . 339
8.299.2.8 pAuthPassword . . . 339
8.299.2.9 pAuthPasswordSize . . . 339
8.299.2.10 pAuthProtocol . . . 340
8.299.2.11 pAuthRetryCount . . . 340
8.299.2.12 pAuthTimeout . . . 340
8.299.2.13 pDataMode . . . 340
8.299.2.14 pDataRate . . . 340
8.299.2.15 pIpcpAckTimeout . . . 340
8.299.2.16 pIpcpCreqRetryCount . . . 340
8.299.2.17 pIsPcscfAddressNedded . . . 340
8.299.2.18 pLcpAckTimeout . . . 340
8.299.2.19 pLcpCreqRetryCount . . . 340
8.299.2.20 pNegoDnsSrvrPref . . . 340

8.299.2.21 pPDNInactivTimeout3GPP2 . . . 340
8.299.2.22 pPdnType . . . 340
8.299.2.23 pPppSessCloseTimer1x . . . 340
8.299.2.24 pPppSessCloseTimerDO . . . 340
8.299.2.25 pPrimaryV4DnsAddress . . . 340
8.299.2.26 pPriV6DnsAddress . . . 340
8.299.2.27 pRATType . . . 340
8.299.2.28 pSecondaryV4DnsAddress . . . 340
8.299.2.29 pSecV6DnsAddress . . . 340
8.299.2.30 pUserId . . . 340
8.299.2.31 pUserIdSize . . . 340
8.300 LibPackprofile_3GPP Struct Reference . . . 340
8.300.1 Detailed Description . . . 341
8.300.2 Field Documentation . . . 345
8.300.2.1 pAddrAllocPref . . . 345
8.300.2.2 pAPNClass . . . 345
8.300.2.3 pAPNDisabledFlag . . . 345
8.300.2.4 pAPNName . . . 345
8.300.2.5 pAPNnameSize . . . 345
8.300.2.6 pAuthenticationPref . . . 345
8.300.2.7 pGPRSMinimumQoS . . . 345
8.300.2.8 pGPRSRequestedQos . . . 345
8.300.2.9 pImCnFlag . . . 345
8.300.2.10 pIPv4AddrPref . . . 345
8.300.2.11 pIPv6AddPref . . . 345
8.300.2.12 pPassword . . . 345
8.300.2.13 pPasswordSize . . . 345
8.300.2.14 pPcscfAddrUsingDhcp . . . 345
8.300.2.15 pPcscfAddrUsingPCO . . . 345
8.300.2.16 pPDNInactivTimeout . . . 345
8.300.2.17 pPdpAccessConFlag . . . 345
8.300.2.18 pPdpContext . . . 345
8.300.2.19 pPdpDataCompType . . . 345
8.300.2.20 pPdpHdrCompType . . . 345
8.300.2.21 pPDPtype . . . 345
8.300.2.22 pPriDNSIPv4AddPref . . . 345
8.300.2.23 pPriDNSIPv6addpref . . . 345
8.300.2.24 pPrimaryID . . . 346
8.300.2.25 pProfilename . . . 346
8.300.2.26 pProfilenameSize . . . 346

8.300.2.27 pQosClassID . . . 346
8.300.2.28 pSecDNSIPv4AddPref . . . 346
8.300.2.29 pSecDNSIPv6addpref . . . 346
8.300.2.30 pSecondaryFlag . . . 346
8.300.2.31 pTFTID1Params . . . 346
8.300.2.32 pTFTID2Params . . . 346
8.300.2.33 pUMTSMinQoS . . . 346
8.300.2.34 pUMTSMinQosSigInd . . . 346
8.300.2.35 pUMTSReqQoS . . . 346
8.300.2.36 pUMTSReqQoSSigInd . . . 346
8.300.2.37 pUsername . . . 346
8.300.2.38 pUsernameSize . . . 346
8.301 LibPackprofile_3GPP2 Struct Reference . . . 346
8.301.1 Detailed Description . . . 347
8.301.2 Field Documentation . . . 350
8.301.2.1 pAllowLinger . . . 350
8.301.2.2 pAPNClass3GPP2 . . . 350
8.301.2.3 pAPNEnabled3GPP2 . . . 350
8.301.2.4 pApnString . . . 350
8.301.2.5 pApnStringSize . . . 350
8.301.2.6 pAppPriority . . . 350
8.301.2.7 pAppType . . . 350
8.301.2.8 pAuthPassword . . . 350
8.301.2.9 pAuthPassword_tSize . . . 350
8.301.2.10 pAuthProtocol . . . 351
8.301.2.11 pAuthRetryCount . . . 351
8.301.2.12 pAuthTimeout . . . 351
8.301.2.13 pDataMode . . . 351
8.301.2.14 pDataRate . . . 351
8.301.2.15 pIpcpAckTimeout . . . 351
8.301.2.16 pIpcpCreqRetryCount . . . 351
8.301.2.17 pIsPcscfAddressNedded . . . 351
8.301.2.18 pLcpAckTimeout . . . 351
8.301.2.19 pLcpCreqRetryCount . . . 351
8.301.2.20 pNegoDnsSrvrPref . . . 351
8.301.2.21 pPDNInactivTimeout3GPP2 . . . 351
8.301.2.22 pPdnType . . . 351
8.301.2.23 pPppSessCloseTimer1x . . . 351
8.301.2.24 pPppSessCloseTimerDO . . . 351
8.301.2.25 pPrimaryV4DnsAddress . . . 351

8.301.2.26 pPriV6DnsAddress . . . 351
8.301.2.27 pRATType . . . 351
8.301.2.28 pSecondaryV4DnsAddress . . . 351
8.301.2.29 pSecV6DnsAddress . . . 351
8.301.2.30 pUserId . . . 351
8.301.2.31 pUserIdSize . . . 351
8.302 LibPackQosClassID Struct Reference . . . 351
8.302.1 Detailed Description . . . 352
8.302.2 Field Documentation . . . 352
8.302.2.1 gDlBitRate . . . 352
8.302.2.2 gUlBitRate . . . 352
8.302.2.3 maxDlBitRate . . . 352
8.302.2.4 maxUlBitRate . . . 352
8.302.2.5 QCI . . . 352
8.303 LibPackTFTIDParams Struct Reference . . . 352
8.303.1 Detailed Description . . . 353
8.303.2 Field Documentation . . . 353
8.303.2.1 destPortRangeEnd . . . 354
8.303.2.2 destPortRangeStart . . . 354
8.303.2.3 eValid . . . 354
8.303.2.4 filterId . . . 354
8.303.2.5 flowLabel . . . 354
8.303.2.6 IPSECSPI . . . 354
8.303.2.7 ipVersion . . . 354
8.303.2.8 nextHeader . . . 354
8.303.2.9 pSourceIP . . . 354
8.303.2.10 sourceIPMask . . . 354
8.303.2.11 srcPortRangeEnd . . . 354
8.303.2.12 srcPortRangeStart . . . 354
8.303.2.13 tosMask . . . 354
8.304 LibPackUMTSQoS Struct Reference . . . 354
8.304.1 Detailed Description . . . 354
8.304.2 Field Documentation . . . 356
8.304.2.1 deliveryErrSDU . . . 356
8.304.2.2 grntDownlinkBitrate . . . 356
8.304.2.3 grntUplinkBitrate . . . 356
8.304.2.4 maxDownlinkBitrate . . . 356
8.304.2.5 maxSDUSize . . . 356
8.304.2.6 maxUplinkBitrate . . . 356
8.304.2.7 qosDeliveryOrder . . . 356

8.304.2.8 resBerRatio . . . 356
8.304.2.9 sduErrorRatio . . . 356
8.304.2.10 trafficClass . . . 356
8.304.2.11 trafficPriority . . . 356
8.304.2.12 transferDelay . . . 356
8.305 LibPackUMTSReqQoSSigInd Struct Reference . . . 356
8.305.1 Detailed Description . . . 357
8.305.2 Field Documentation . . . 357
8.305.2.1 SigInd . . . 357
8.305.2.2 UMTSReqQoS . . . 357
8.306 lineCtrlInfo Struct Reference . . . 357
8.306.1 Detailed Description . . . 357
8.306.2 Field Documentation . . . 358
8.306.2.1 polarityIncluded . . . 358
8.306.2.2 pwrDenialTime . . . 358
8.306.2.3 revPolarity . . . 358
8.306.2.4 toggleMode . . . 358
8.307 loc_BdsSV Struct Reference . . . 358
8.307.1 Detailed Description . . . 358
8.307.2 Field Documentation . . . 358
8.307.2.1 id . . . 358
8.307.2.2 mask . . . 358
8.308 loc_BdsSVInfo Struct Reference . . . 358
8.308.1 Detailed Description . . . 358
8.308.2 Field Documentation . . . 359
8.308.2.1 len . . . 359
8.308.2.2 pSV . . . 359
8.309 loc_CellDb Struct Reference . . . 359
8.309.1 Detailed Description . . . 359
8.309.2 Field Documentation . . . 359
8.309.2.1 mask . . . 359
8.310 loc_ClkInfo Struct Reference . . . 359
8.310.1 Detailed Description . . . 360
8.310.2 Field Documentation . . . 360
8.310.2.1 mask . . . 361
8.311 loc_GnssData Struct Reference . . . 361
8.311.1 Detailed Description . . . 361
8.311.2 Field Documentation . . . 362
8.311.2.1 mask . . . 362
8.312 loc_gpsTime Struct Reference . . . 362

8.312.1 Detailed Description . . . 363
8.312.2 Field Documentation . . . 363
8.312.2.1 gpsTimeOfWeekMs . . . 363
8.312.2.2 gpsWeek . . . 363
8.313loc_LocApplicationInfo Struct Reference . . . 363
8.313.1 Detailed Description . . . 363
8.313.2 Field Documentation . . . 364
8.313.2.1 appNameLength . . . 364
8.313.2.2 appProviderLength . . . 364
8.313.2.3 appVersionLength . . . 364
8.313.2.4 appVersionValid . . . 364
8.313.2.5 pAppName . . . 364
8.313.2.6 pAppProvider . . . 364
8.313.2.7 pAppVersion . . . 364
8.314loc_precisionDilution Struct Reference . . . 364
8.314.1 Detailed Description . . . 364
8.314.2 Field Documentation . . . 365
8.314.2.1 HDOP . . . 365
8.314.2.2 PDOP . . . 365
8.314.2.3 VDOP . . . 365
8.315loc_satelliteInfo Struct Reference . . . 365
8.315.1 Detailed Description . . . 365
8.315.2 Field Documentation . . . 367
8.315.2.1 azimuth . . . 367
8.315.2.2 elevation . . . 367
8.315.2.3 gnssSvId . . . 367
8.315.2.4 healthStatus . . . 367
8.315.2.5 snr . . . 367
8.315.2.6 svInfoMask . . . 367
8.315.2.7 svListLen . . . 367
8.315.2.8 svStatus . . . 367
8.315.2.9 system . . . 367
8.315.2.10validMask . . . 367
8.316loc_sensorDataUsage Struct Reference . . . 367
8.316.1 Detailed Description . . . 367
8.316.2 Field Documentation . . . 368
8.316.2.1 aidingIndicatorMask . . . 368
8.316.2.2 usageMask . . . 368
8.317loc_SV Struct Reference . . . 368
8.317.1 Detailed Description . . . 368

8.317.2 Field Documentation . . . 369
8.317.2.1 id . . . 369
8.317.2.2 mask . . . 369
8.317.2.3 system . . . 369
8.318loc_SVInfo Struct Reference . . . 369
8.318.1 Detailed Description . . . 369
8.318.2 Field Documentation . . . 369
8.318.2.1 len . . . 369
8.318.2.2 pSV . . . 369
8.319loc_svUsedforFix Struct Reference . . . 369
8.319.1 Detailed Description . . . 370
8.319.2 Field Documentation . . . 370
8.319.2.1 gnssSvUsedList . . . 370
8.319.2.2 gnssSvUsedList_len . . . 370
8.320LocApplicationInfo Struct Reference . . . 370
8.320.1 Detailed Description . . . 370
8.320.2 Field Documentation . . . 371
8.320.2.1 appNameLength . . . 371
8.320.2.2 appProviderLength . . . 371
8.320.2.3 appVersionLength . . . 371
8.320.2.4 appVersionValid . . . 371
8.320.2.5 pAppName . . . 371
8.320.2.6 pAppProvider . . . 371
8.320.2.7 pAppVersion . . . 371
8.321LocDelAssDataReq Struct Reference . . . 371
8.321.1 Detailed Description . . . 371
8.321.2 Field Documentation . . . 372
8.321.2.1 pBdsSVInfo . . . 372
8.321.2.2 pCellDb . . . 372
8.321.2.3 pClkInfo . . . 372
8.321.2.4 pGnssData . . . 372
8.321.2.5 pSVInfo . . . 372
8.322LOCEventRegisterReqResp Struct Reference . . . 372
8.322.1 Detailed Description . . . 372
8.322.2 Field Documentation . . . 374
8.322.2.1 eventRegister . . . 374
8.323LOCExtPowerStateReqResp Struct Reference . . . 374
8.323.1 Detailed Description . . . 374
8.323.2 Field Documentation . . . 374
8.323.2.1 extPowerState . . . 374

8.324 LocInjectPositionReq Struct Reference . . . 374
8.324.1 Detailed Description . . . 375
8.324.2 Field Documentation . . . 377
8.324.2.1 pAltitudeSrcInfo . . . 377
8.324.2.2 pAltitudeWrtEllipsoid . . . 377
8.324.2.3 pAltitudeWrtMeanSeaLevel . . . 377
8.324.2.4 pHorConfidence . . . 378
8.324.2.5 pHorReliability . . . 378
8.324.2.6 pHorUncCircular . . . 378
8.324.2.7 pLatitude . . . 378
8.324.2.8 pLongitude . . . 378
8.324.2.9 pPositionSrc . . . 378
8.324.2.10 pRawHorConfidence . . . 378
8.324.2.11 pRawHorUncCircular . . . 378
8.324.2.12 pTimestampAge . . . 378
8.324.2.13 pTimestampUtc . . . 378
8.324.2.14 pVertConfidence . . . 378
8.324.2.15 pVertReliability . . . 378
8.324.2.16 pVertUnc . . . 378
8.325 LocInjectSensorDataReq Struct Reference . . . 378
8.325.1 Detailed Description . . . 378
8.325.2 Field Documentation . . . 379
8.325.2.1 pAcceleroData . . . 379
8.325.2.2 pAcceleroTempData . . . 379
8.325.2.3 pAcceleroTimeSrc . . . 379
8.325.2.4 pGyroData . . . 379
8.325.2.5 pGyroTempData . . . 379
8.325.2.6 pGyroTimeSrc . . . 379
8.325.2.7 pOpaqueIdentifier . . . 379
8.326 LocSetCradleMountReq Struct Reference . . . 379
8.326.1 Detailed Description . . . 380
8.326.2 Field Documentation . . . 380
8.326.2.1 pConfidence . . . 380
8.326.2.2 state . . . 380
8.327 LOCStartReq Struct Reference . . . 380
8.327.1 Detailed Description . . . 380
8.327.2 Field Documentation . . . 381
8.327.2.1 pApplicationInfo . . . 381
8.327.2.2 pConfigAltitudeAssumed . . . 381
8.327.2.3 pHorizontalAccuracyLvl . . . 382

8.327.2.4 pIntermediateReportState . . . 382
8.327.2.5 pMinIntervalTime . . . 382
8.327.2.6 pRecurrenceType . . . 382
8.327.2.7 SessionId . . . 382
8.328 LOCStopReq Struct Reference . . . 382
8.328.1 Detailed Description . . . 382
8.328.2 Field Documentation . . . 382
8.328.2.1 sessionId . . . 382
8.329 LteCQIParm Struct Reference . . . 382
8.329.1 Detailed Description . . . 382
8.329.2 Field Documentation . . . 383
8.329.2.1 CQIValueCW0 . . . 383
8.329.2.2 CQIValueCW1 . . . 383
8.329.2.3 ValidityCW0 . . . 383
8.329.2.4 ValidityCW1 . . . 383
8.330 lteEARFCN Struct Reference . . . 383
8.330.1 Detailed Description . . . 383
8.330.2 Field Documentation . . . 383
8.330.2.1 earfcn0 . . . 384
8.330.2.2 earfcn1 . . . 384
8.330.2.3 status . . . 384
8.331 lteGsmCellInfo Struct Reference . . . 384
8.331.1 Detailed Description . . . 384
8.331.2 Field Documentation . . . 385
8.331.2.1 cellReselPriority . . . 385
8.331.2.2 cells_len . . . 385
8.331.2.3 GsmCellInfo . . . 385
8.331.2.4 nccPermitted . . . 385
8.331.2.5 threshGsmHigh . . . 385
8.331.2.6 threshGsmLow . . . 385
8.332 LTEInfo Struct Reference . . . 385
8.332.1 Detailed Description . . . 385
8.332.2 Field Documentation . . . 387
8.332.2.1 band . . . 387
8.332.2.2 bandwidth . . . 387
8.332.2.3 emmConnState . . . 387
8.332.2.4 emmState . . . 387
8.332.2.5 emmSubState . . . 387
8.332.2.6 RXChan . . . 387
8.332.2.7 TXChan . . . 387

8.333 LTEInfoInterfreq Struct Reference . . . 387
8.333.1 Detailed Description . . . 387
8.333.2 Field Documentation . . . 387
8.333.2.1 freqsLen . . . 387
8.333.2.2 InfoInterfreq . . . 387
8.333.2.3 ueInIdle . . . 387
8.334 LTEInfoIntrafreq Struct Reference . . . 387
8.334.1 Detailed Description . . . 388
8.334.2 Field Documentation . . . 389
8.334.2.1 CellParams . . . 389
8.334.2.2 cellReselPriority . . . 389
8.334.2.3 cellsLen . . . 389
8.334.2.4 earfcn . . . 389
8.334.2.5 globalCellId . . . 389
8.334.2.6 plmn . . . 389
8.334.2.7 servingCellId . . . 389
8.334.2.8 sIntraSearch . . . 389
8.334.2.9 sNonIntraSearch . . . 389
8.334.2.10 tac . . . 389
8.334.2.11 threshServingLow . . . 389
8.334.2.12 ueInIdle . . . 390
8.335 LTEInfoNeighboringGSM Struct Reference . . . 390
8.335.1 Detailed Description . . . 390
8.335.2 Field Documentation . . . 390
8.335.2.1 freqsLen . . . 390
8.335.2.2 LteGsmCellInfo . . . 390
8.335.2.3 ueInIdle . . . 390
8.336 LTEInfoNeighboringWCDMA Struct Reference . . . 390
8.336.1 Detailed Description . . . 390
8.336.2 Field Documentation . . . 391
8.336.2.1 freqsLen . . . 391
8.336.2.2 LTEWCDMACellInfo . . . 391
8.336.2.3 ueInIdle . . . 391
8.337 LteNasReleaseInfo_s Struct Reference . . . 391
8.337.1 Detailed Description . . . 391
8.337.2 Field Documentation . . . 391
8.337.2.1 nas_major . . . 391
8.337.2.2 nas_minor . . . 391
8.337.2.3 nas_release . . . 392
8.338 ltePCI Struct Reference . . . 392

8.338.1 Detailed Description . . . 392
8.338.2 Field Documentation . . . 392
8.338.2.1 earfcn . . . 392
8.338.2.2 pci . . . 392
8.338.2.3 status . . . 392
8.339 lteRsrpinformation Struct Reference . . . 392
8.339.1 Detailed Description . . . 392
8.339.2 Field Documentation . . . 393
8.339.2.1 rsrplevel . . . 393
8.340 LTERSRPThresh Struct Reference . . . 393
8.340.1 Detailed Description . . . 393
8.340.2 Field Documentation . . . 393
8.340.2.1 LTERSRPThreshListLen . . . 393
8.340.2.2 pLTERSRPThreshList . . . 393
8.341 LTERSRQThresh Struct Reference . . . 393
8.341.1 Detailed Description . . . 393
8.341.2 Field Documentation . . . 394
8.341.2.1 LTERSRQThreshListLen . . . 394
8.341.2.2 pLTERSRQThreshList . . . 394
8.342 LTERSSIThresh Struct Reference . . . 394
8.342.1 Detailed Description . . . 394
8.342.2 Field Documentation . . . 394
8.342.2.1 LTERSSIThreshListLen . . . 394
8.342.2.2 pLTERSSIThreshList . . . 394
8.343 LteSccRxInfoResp Struct Reference . . . 394
8.343.1 Detailed Description . . . 394
8.343.2 Field Documentation . . . 394
8.343.2.1 pSccRxInfo . . . 395
8.344 LTESigRptCfg Struct Reference . . . 395
8.344.1 Detailed Description . . . 395
8.344.2 Field Documentation . . . 395
8.344.2.1 avgPeriod . . . 395
8.344.2.2 rptRate . . . 395
8.345 LTESigRptConfig Struct Reference . . . 395
8.345.1 Detailed Description . . . 396
8.345.2 Field Documentation . . . 396
8.345.2.1 avgPeriod . . . 396
8.345.2.2 rptRate . . . 396
8.346 lteSnrinformation Struct Reference . . . 396
8.346.1 Detailed Description . . . 396

8.346.2 Field Documentation . . . 397
8.346.2.1 snrlevel . . . 397
8.347 LTESNRThresh Struct Reference . . . 397
8.347.1 Detailed Description . . . 397
8.347.2 Field Documentation . . . 397
8.347.2.1 LTESNRThresListLen . . . 397
8.347.2.2 pLTESNRThresList . . . 397
8.348 LTESNRThreshold Struct Reference . . . 397
8.348.1 Detailed Description . . . 398
8.348.2 Field Documentation . . . 398
8.348.2.1 LTESNRThreshListLen . . . 398
8.348.2.2 pLTESNRThreshList . . . 398
8.349 LTESSInfo Struct Reference . . . 398
8.349.1 Detailed Description . . . 398
8.349.2 Field Documentation . . . 399
8.349.2.1 rsrp . . . 399
8.349.2.2 rsrq . . . 399
8.349.2.3 rssi . . . 399
8.349.2.4 snr . . . 399
8.350 lteSSInfo Struct Reference . . . 399
8.350.1 Detailed Description . . . 399
8.350.2 Field Documentation . . . 399
8.350.2.1 rsrp . . . 399
8.350.2.2 rsrq . . . 399
8.350.2.3 rssi . . . 399
8.350.2.4 snr . . . 399
8.351 LTESysInfo Struct Reference . . . 399
8.351.1 Detailed Description . . . 400
8.351.2 Field Documentation . . . 401
8.351.2.1 cellId . . . 401
8.351.2.2 cellIdValid . . . 401
8.351.2.3 lac . . . 401
8.351.2.4 lacValid . . . 401
8.351.2.5 MCC . . . 401
8.351.2.6 MNC . . . 401
8.351.2.7 networkIdValid . . . 401
8.351.2.8 regRejectInfoValid . . . 401
8.351.2.9 rejCause . . . 401
8.351.2.10 rejectSrvDomain . . . 401
8.351.2.11 sysInfoLTE . . . 401

8.351.2.12 tac . . . 402
8.351.2.13 tacValid . . . 402
8.352 lteWcdmaCellInfo Struct Reference . . . 402
8.352.1 Detailed Description . . . 402
8.352.2 Field Documentation . . . 402
8.352.2.1 cellReselPriority . . . 402
8.352.2.2 cellsLen . . . 402
8.352.2.3 threshXhigh . . . 403
8.352.2.4 threshXlow . . . 403
8.352.2.5 uarfcn . . . 403
8.352.2.6 WCDMACellInfo . . . 403
8.353 messageModeTlv Struct Reference . . . 403
8.353.1 Detailed Description . . . 403
8.353.2 Field Documentation . . . 403
8.353.2.1 MessageModeInfo . . . 403
8.353.2.2 TlvPresent . . . 403
8.354 messageWaitingInfoContent Struct Reference . . . 403
8.354.1 Detailed Description . . . 403
8.354.2 Field Documentation . . . 404
8.354.2.1 activeInd . . . 404
8.354.2.2 msgCount . . . 404
8.354.2.3 msgType . . . 404
8.355 minBasedIMSI Struct Reference . . . 404
8.355.1 Detailed Description . . . 404
8.355.2 Field Documentation . . . 405
8.355.2.1 imsiM1112 . . . 405
8.355.2.2 imsiMS1 . . . 405
8.355.2.3 imsiMS2 . . . 405
8.355.2.4 mccM . . . 405
8.356 mitigationDevList Struct Reference . . . 405
8.356.1 Detailed Description . . . 405
8.356.2 Field Documentation . . . 405
8.356.2.1 maxMitigationLevel . . . 405
8.356.2.2 mitigationDevId . . . 405
8.356.2.3 mitigationDevIdLen . . . 405
8.357 MNRInfo Struct Reference . . . 405
8.357.1 Detailed Description . . . 406
8.357.2 Field Documentation . . . 406
8.357.2.1 mcc . . . 406
8.357.2.2 mnc . . . 406

8.357.2.3 rat . . . 406
8.358 ModifyProfileIn Struct Reference . . . 406
8.358.1 Detailed Description . . . 406
8.358.2 Field Documentation . . . 407
8.358.2.1 curProfile . . . 407
8.358.2.2 pProfileID . . . 407
8.358.2.3 pProfileType . . . 407
8.359 ModifyProfileOut Struct Reference . . . 407
8.359.1 Detailed Description . . . 407
8.359.2 Field Documentation . . . 407
8.359.2.1 pExtErrorCode . . . 407
8.360 msgWaitingInfo Struct Reference . . . 407
8.360.1 Detailed Description . . . 407
8.360.2 Field Documentation . . . 408
8.360.2.1 msgWaitInfo . . . 408
8.360.2.2 numInstances . . . 408
8.361 namName Struct Reference . . . 408
8.361.1 Detailed Description . . . 408
8.361.2 Field Documentation . . . 408
8.361.2.1 namName . . . 408
8.361.2.2 namNameLen . . . 408
8.362 nas_acqOrderPref Struct Reference . . . 408
8.362.1 Detailed Description . . . 408
8.362.2 Field Documentation . . . 409
8.362.2.1 acqOrdeLen . . . 409
8.362.2.2 pAcqOrder . . . 409
8.363 nas_ActPilotPNElement Struct Reference . . . 409
8.363.1 Detailed Description . . . 409
8.363.2 Field Documentation . . . 409
8.363.2.1 ActSetPilotPN . . . 409
8.363.2.2 ActSetPilotPNStrength . . . 409
8.364 nas_AddCDMASysInfo Struct Reference . . . 409
8.364.1 Detailed Description . . . 410
8.364.2 Field Documentation . . . 410
8.364.2.1 geoSysIdx . . . 410
8.364.2.2 regPrd . . . 410
8.365 nas_AddSysInfo Struct Reference . . . 410
8.365.1 Detailed Description . . . 410
8.365.2 Field Documentation . . . 410
8.365.2.1 cellBroadcastCap . . . 410

8.365.2.2 geoSysIdx . . . 411
8.366 nas_CallBarringSysInfo Struct Reference . . . 411
8.366.1 Detailed Description . . . 411
8.366.2 Field Documentation . . . 411
8.366.2.1 csBarStatus . . . 411
8.366.2.2 psBarStatus . . . 411
8.367 nas_callBarStatus Struct Reference . . . 411
8.367.1 Detailed Description . . . 411
8.367.2 Field Documentation . . . 412
8.367.2.1 csBarStatus . . . 412
8.367.2.2 psBarStatus . . . 412
8.368 nas_CDMAChannel Struct Reference . . . 412
8.368.1 Detailed Description . . . 412
8.368.2 Field Documentation . . . 413
8.368.2.1 priChA . . . 413
8.368.2.2 priChB . . . 413
8.368.2.3 secChA . . . 413
8.368.2.4 secChB . . . 413
8.369 nas_CDMAECIOThresh Struct Reference . . . 413
8.369.1 Detailed Description . . . 413
8.369.2 Field Documentation . . . 413
8.369.2.1 CDMAECIOThreshListLen . . . 413
8.369.2.2 pCDMAECIOThreshList . . . 413
8.370 nas_CDMAInfo Struct Reference . . . 413
8.370.1 Detailed Description . . . 414
8.370.2 Field Documentation . . . 414
8.370.2.1 baseId . . . 414
8.370.2.2 baseLat . . . 414
8.370.2.3 baseLong . . . 414
8.370.2.4 nid . . . 414
8.370.2.5 refpn . . . 414
8.370.2.6 sid . . . 414
8.371 nas_CDMARSSIThresh Struct Reference . . . 414
8.371.1 Detailed Description . . . 415
8.371.2 Field Documentation . . . 415
8.371.2.1 CDMARSSIThreshListLen . . . 415
8.371.2.2 pCDMARSSIThreshList . . . 415
8.372 nas_CDMASysInfo Struct Reference . . . 415
8.372.1 Detailed Description . . . 415
8.372.2 Field Documentation . . . 418

8.372.2.1 baseId . . . 418
8.372.2.2 baseLat . . . 418
8.372.2.3 baseLong . . . 418
8.372.2.4 bsInfoValid . . . 418
8.372.2.5 bsPRev . . . 418
8.372.2.6 bsPRevValid . . . 418
8.372.2.7 ccsSupported . . . 418
8.372.2.8 ccsSupportedValid . . . 418
8.372.2.9 cdmaSysIdValid . . . 418
8.372.2.10 isSysPrlMatch . . . 418
8.372.2.11 isSysPrlMatchValid . . . 418
8.372.2.12 MCC . . . 418
8.372.2.13 MNC . . . 418
8.372.2.14 networkID . . . 418
8.372.2.15 networkIdValid . . . 418
8.372.2.16 packetZone . . . 418
8.372.2.17 packetZoneValid . . . 418
8.372.2.18 pRevInUse . . . 419
8.372.2.19 pRevInUseValid . . . 419
8.372.2.20 sysInfoCDMA . . . 419
8.372.2.21 systemID . . . 419
8.373 nas_CDMASysInfoExt Struct Reference . . . 419
8.373.1 Detailed Description . . . 419
8.373.2 Field Documentation . . . 419
8.373.2.1 imsi_11_12 . . . 419
8.373.2.2 MCC . . . 419
8.374 nas_cellParams Struct Reference . . . 419
8.374.1 Detailed Description . . . 419
8.374.2 Field Documentation . . . 420
8.374.2.1 pci . . . 420
8.374.2.2 rsrp . . . 420
8.374.2.3 rsrq . . . 420
8.374.2.4 rssi . . . 420
8.374.2.5 srxlev . . . 420
8.375 nas_CommInfo Struct Reference . . . 420
8.375.1 Detailed Description . . . 420
8.375.2 Field Documentation . . . 422
8.375.2.1 imsRegState . . . 422
8.375.2.2 modemMode . . . 422
8.375.2.3 psState . . . 422

8.375.2.4 systemMode . . . 422
8.375.2.5 temperature . . . 422
8.376 nas_CSGID Struct Reference . . . 422
8.376.1 Detailed Description . . . 422
8.376.2 Field Documentation . . . 422
8.376.2.1 id . . . 423
8.376.2.2 mcc . . . 423
8.376.2.3 mnc . . . 423
8.376.2.4 mncPcsDigits . . . 423
8.376.2.5 rat . . . 423
8.377 nas_currentPLMN Struct Reference . . . 423
8.377.1 Detailed Description . . . 423
8.377.2 Field Documentation . . . 423
8.377.2.1 MCC . . . 423
8.377.2.2 MNC . . . 423
8.377.2.3 netDescr . . . 424
8.377.2.4 netDescrLength . . . 424
8.378 nas_dataSrvCapabilities Struct Reference . . . 424
8.378.1 Detailed Description . . . 424
8.378.2 Field Documentation . . . 424
8.378.2.1 dataCapabilities . . . 424
8.378.2.2 dataCapabilitiesLen . . . 424
8.379 nas_DataStatusDetail Struct Reference . . . 424
8.379.1 Detailed Description . . . 425
8.379.2 Field Documentation . . . 426
8.379.2.1 IPAddress . . . 426
8.379.2.2 LastErrCode . . . 426
8.380 nas_detailSvcInfo Struct Reference . . . 427
8.380.1 Detailed Description . . . 427
8.380.2 Field Documentation . . . 428
8.380.2.1 hdrHybrid . . . 428
8.380.2.2 hdrSrvStatus . . . 428
8.380.2.3 isSysForbidden . . . 428
8.380.2.4 srvCapability . . . 428
8.380.2.5 srvStatus . . . 428
8.381 nas_DeviceConfigDetail Struct Reference . . . 428
8.381.1 Detailed Description . . . 428
8.381.2 Field Documentation . . . 429
8.381.2.1 Chipset . . . 429
8.381.2.2 HWVersion . . . 429

8.381.2.3 QLIC . . . 429
8.381.2.4 Technology . . . 429
8.382 nas_dirNum Struct Reference . . . 429
8.382.1 Detailed Description . . . 429
8.382.2 Field Documentation . . . 429
8.382.2.1 dirNum . . . 429
8.382.2.2 dirNumLen . . . 429
8.383 nas_DRCParams Struct Reference . . . 429
8.383.1 Detailed Description . . . 430
8.383.2 Field Documentation . . . 430
8.383.2.1 DRCCover . . . 430
8.383.2.2 DRCValue . . . 430
8.384 nas_ecioListElement Struct Reference . . . 430
8.384.1 Detailed Description . . . 430
8.384.2 Field Documentation . . . 430
8.384.2.1 ecio . . . 430
8.384.2.2 radioIf . . . 431
8.385 nas_ECIOThresh Struct Reference . . . 431
8.385.1 Detailed Description . . . 431
8.385.2 Field Documentation . . . 431
8.385.2.1 ECIOThresListLen . . . 431
8.385.2.2 pECIOThresList . . . 431
8.386 nas_errorRateListElement Struct Reference . . . 431
8.386.1 Detailed Description . . . 431
8.386.2 Field Documentation . . . 432
8.386.2.1 errorRate . . . 432
8.386.2.2 radioIf . . . 432
8.387 nas_GERANInfo Struct Reference . . . 432
8.387.1 Detailed Description . . . 433
8.387.2 Field Documentation . . . 434
8.387.2.1 arfcn . . . 434
8.387.2.2 bsic . . . 434
8.387.2.3 cellID . . . 434
8.387.2.4 insNmrCellInfo . . . 434
8.387.2.5 lac . . . 434
8.387.2.6 nmrInst . . . 434
8.387.2.7 plmn . . . 434
8.387.2.8 rxLev . . . 434
8.387.2.9 timingAdvance . . . 434
8.388 nas_geranInstInfo Struct Reference . . . 434

8.388.1 Detailed Description . . . 434
8.388.2 Field Documentation . . . 434
8.388.2.1 geranArfcn . . . 434
8.388.2.2 geranBsicBcc . . . 435
8.388.2.3 geranBsicNcc . . . 435
8.388.2.4 geranRssi . . . 435
8.389 nas_gsmCellInfo Struct Reference . . . 435
8.389.1 Detailed Description . . . 435
8.389.2 Field Documentation . . . 435
8.389.2.1 arfcn . . . 436
8.389.2.2 band1900 . . . 436
8.389.2.3 bsicId . . . 436
8.389.2.4 cellIdValid . . . 436
8.389.2.5 rssi . . . 436
8.389.2.6 srxlev . . . 436
8.390 nas_GSMRSSIThresh Struct Reference . . . 436
8.390.1 Detailed Description . . . 436
8.390.2 Field Documentation . . . 436
8.390.2.1 GSMRSSIThreshListLen . . . 436
8.390.2.2 pGSMRSSIThreshList . . . 436
8.391 nas_GSMSrvStatusInfo Struct Reference . . . 436
8.391.1 Detailed Description . . . 436
8.391.2 Field Documentation . . . 437
8.391.2.1 isPrefDataPath . . . 437
8.391.2.2 srvStatus . . . 437
8.391.2.3 trueSrvStatus . . . 437
8.392 nas_GSMSysInfo Struct Reference . . . 437
8.392.1 Detailed Description . . . 438
8.392.2 Field Documentation . . . 439
8.392.2.1 cellId . . . 439
8.392.2.2 cellIdValid . . . 439
8.392.2.3 dtmSupp . . . 439
8.392.2.4 dtmSuppValid . . . 440
8.392.2.5 egprsSupp . . . 440
8.392.2.6 egprsSuppValid . . . 440
8.392.2.7 lac . . . 440
8.392.2.8 lacValid . . . 440
8.392.2.9 MCC . . . 440
8.392.2.10 MNC . . . 440
8.392.2.11 networkIdValid . . . 440

8.392.2.12 regRejectInfoValid . . . 440
8.392.2.13 rejCause . . . 440
8.392.2.14 rejectSrvDomain . . . 440
8.392.2.15 sysInfoGSM . . . 440
8.393 nas_HDRECIOThresh Struct Reference . . . 440
8.393.1 Detailed Description . . . 440
8.393.2 Field Documentation . . . 440
8.393.2.1 HDRECIOThreshListLen . . . 440
8.393.2.2 pHDRECIOThreshList . . . 440
8.394 nas_HDRIOThresh Struct Reference . . . 440
8.394.1 Detailed Description . . . 441
8.394.2 Field Documentation . . . 441
8.394.2.1 HDRIOThreshListLen . . . 441
8.394.2.2 pHDRIOThreshList . . . 441
8.395 nas_HDRRSSIThresh Struct Reference . . . 441
8.395.1 Detailed Description . . . 441
8.395.2 Field Documentation . . . 441
8.395.2.1 HDRRSSIThreshListLen . . . 441
8.395.2.2 pHDRRSSIThreshList . . . 442
8.396 nas_HDRSINRThresh Struct Reference . . . 442
8.396.1 Detailed Description . . . 442
8.396.2 Field Documentation . . . 442
8.396.2.1 HDRSINRThresListLen . . . 442
8.396.2.2 pHDRSINRThresList . . . 442
8.397 nas_HDRSINRThreshold Struct Reference . . . 442
8.397.1 Detailed Description . . . 442
8.397.2 Field Documentation . . . 443
8.397.2.1 HDRSINRThreshListLen . . . 443
8.397.2.2 pHDRSINRThreshList . . . 443
8.398 nas_HDRSysInfo Struct Reference . . . 443
8.398.1 Detailed Description . . . 443
8.398.2 Field Documentation . . . 445
8.398.2.1 hdrActiveProt . . . 445
8.398.2.2 hdrActiveProtValid . . . 445
8.398.2.3 hdrPersonality . . . 445
8.398.2.4 hdrPersonalityValid . . . 445
8.398.2.5 is856SysId . . . 445
8.398.2.6 is856SysIdValid . . . 445
8.398.2.7 isSysPrlMatch . . . 445
8.398.2.8 isSysPrlMatchValid . . . 445

8.398.2.9 sysInfoHDR . . . 445
8.399 nas_homeSIDNID Struct Reference . . . 445
8.399.1 Detailed Description . . . 445
8.399.2 Field Documentation . . . 445
8.399.2.1 numInstances . . . 445
8.399.2.2 SidNid . . . 445
8.400 nas_infoInterFreq Struct Reference . . . 446
8.400.1 Detailed Description . . . 446
8.400.2 Field Documentation . . . 446
8.400.2.1 cell_resel_priority . . . 446
8.400.2.2 cellInterFreqParams . . . 446
8.400.2.3 cells_len . . . 446
8.400.2.4 earfcn . . . 447
8.400.2.5 threshXHigh . . . 447
8.400.2.6 threshXLow . . . 447
8.401 nas_IOThresh Struct Reference . . . 447
8.401.1 Detailed Description . . . 447
8.401.2 Field Documentation . . . 447
8.401.2.1 IOThresListLen . . . 447
8.401.2.2 pIOThresList . . . 447
8.402 nas_lteEARFCN Struct Reference . . . 447
8.402.1 Detailed Description . . . 447
8.402.2 Field Documentation . . . 448
8.402.2.1 earfcn0 . . . 448
8.402.2.2 earfcn1 . . . 448
8.402.2.3 status . . . 448
8.403 nas_lteGsmCellInfo Struct Reference . . . 448
8.403.1 Detailed Description . . . 448
8.403.2 Field Documentation . . . 449
8.403.2.1 cellReselPriority . . . 449
8.403.2.2 cells_len . . . 449
8.403.2.3 GsmCellInfo . . . 449
8.403.2.4 nccPermitted . . . 449
8.403.2.5 threshGsmHigh . . . 449
8.403.2.6 threshGsmLow . . . 449
8.404 nas_LTEInfo Struct Reference . . . 449
8.404.1 Detailed Description . . . 449
8.404.2 Field Documentation . . . 451
8.404.2.1 band . . . 451
8.404.2.2 bandwidth . . . 451

8.404.2.3 emmConnState . . . 451
8.404.2.4 emmState . . . 451
8.404.2.5 emmSubState . . . 451
8.404.2.6 RXChan . . . 451
8.404.2.7 TXChan . . . 451
8.405 nas_LTEInfoInterfreq Struct Reference . . . 452
8.405.1 Detailed Description . . . 452
8.405.2 Field Documentation . . . 452
8.405.2.1 freqsLen . . . 452
8.405.2.2 InfoInterfreq . . . 452
8.405.2.3 ueInIdle . . . 452
8.406 nas_LTEInfoIntrafreq Struct Reference . . . 452
8.406.1 Detailed Description . . . 453
8.406.2 Field Documentation . . . 454
8.406.2.1 CellParams . . . 454
8.406.2.2 cellReselPriority . . . 454
8.406.2.3 cellsLen . . . 454
8.406.2.4 earfcn . . . 454
8.406.2.5 globalCellId . . . 454
8.406.2.6 plmn . . . 454
8.406.2.7 servingCellId . . . 454
8.406.2.8 sIntraSearch . . . 454
8.406.2.9 sNonIntraSearch . . . 454
8.406.2.10 tac . . . 454
8.406.2.11 threshServingLow . . . 454
8.406.2.12 ueInIdle . . . 454
8.407 nas_LTEInfoNeighboringGSM Struct Reference . . . 454
8.407.1 Detailed Description . . . 455
8.407.2 Field Documentation . . . 455
8.407.2.1 freqsLen . . . 455
8.407.2.2 LteGsmCellInfo . . . 455
8.407.2.3 ueInIdle . . . 455
8.408 nas_LTEInfoNeighboringWCDMA Struct Reference . . . 455
8.408.1 Detailed Description . . . 455
8.408.2 Field Documentation . . . 456
8.408.2.1 freqsLen . . . 456
8.408.2.2 LTEWCDMACellInfo . . . 456
8.408.2.3 ueInIdle . . . 456
8.409 nas_ltePCI Struct Reference . . . 456
8.409.1 Detailed Description . . . 456

8.409.2 Field Documentation . . . 456
8.409.2.1 earfcn . . . 456
8.409.2.2 pci . . . 456
8.409.2.3 status . . . 456
8.410 nas_lteRsrpinformation Struct Reference . . . 456
8.410.1 Detailed Description . . . 456
8.410.2 Field Documentation . . . 457
8.410.2.1 rsrplevel . . . 457
8.411 nas_LTERSRPThresh Struct Reference . . . 457
8.411.1 Detailed Description . . . 457
8.411.2 Field Documentation . . . 457
8.411.2.1 LTERSRPThreshListLen . . . 457
8.411.2.2 pLTERSRPThreshList . . . 457
8.412 nas_LTERSRQThresh Struct Reference . . . 457
8.412.1 Detailed Description . . . 457
8.412.2 Field Documentation . . . 458
8.412.2.1 LTERSRQThreshListLen . . . 458
8.412.2.2 pLTERSRQThreshList . . . 458
8.413 nas_LTERSSIThresh Struct Reference . . . 458
8.413.1 Detailed Description . . . 458
8.413.2 Field Documentation . . . 458
8.413.2.1 LTERSSIThreshListLen . . . 458
8.413.2.2 pLTERSSIThreshList . . . 458
8.414 nas_LTESigRptCfg Struct Reference . . . 458
8.414.1 Detailed Description . . . 459
8.414.2 Field Documentation . . . 459
8.414.2.1 avgPeriod . . . 459
8.414.2.2 rptRate . . . 459
8.415 nas_LTESigRptConfig Struct Reference . . . 459
8.415.1 Detailed Description . . . 459
8.415.2 Field Documentation . . . 460
8.415.2.1 avgPeriod . . . 460
8.415.2.2 rptRate . . . 460
8.416 nas_lteSnrinformation Struct Reference . . . 460
8.416.1 Detailed Description . . . 460
8.416.2 Field Documentation . . . 460
8.416.2.1 snrlevel . . . 461
8.417 nas_LTESNRThresh Struct Reference . . . 461
8.417.1 Detailed Description . . . 461
8.417.2 Field Documentation . . . 461

8.417.2.1 LTESNRThresListLen . . . 461
8.417.2.2 pLTESNRThresList . . . 461
8.418 nas_LTESNRThreshold Struct Reference . . . 461
8.418.1 Detailed Description . . . 461
8.418.2 Field Documentation . . . 462
8.418.2.1 LTESNRThreshListLen . . . 462
8.418.2.2 pLTESNRThreshList . . . 462
8.419 nas_LTESysInfo Struct Reference . . . 462
8.419.1 Detailed Description . . . 462
8.419.2 Field Documentation . . . 464
8.419.2.1 cellId . . . 464
8.419.2.2 cellIdValid . . . 464
8.419.2.3 lac . . . 464
8.419.2.4 lacValid . . . 464
8.419.2.5 MCC . . . 464
8.419.2.6 MNC . . . 464
8.419.2.7 networkIdValid . . . 464
8.419.2.8 regRejectInfoValid . . . 464
8.419.2.9 rejCause . . . 464
8.419.2.10 rejectSrvDomain . . . 464
8.419.2.11 sysInfoLTE . . . 464
8.419.2.12 tac . . . 464
8.419.2.13 tacValid . . . 464
8.420 nas_lteWcdmaCellInfo Struct Reference . . . 464
8.420.1 Detailed Description . . . 465
8.420.2 Field Documentation . . . 465
8.420.2.1 cellReselPriority . . . 465
8.420.2.2 cellsLen . . . 465
8.420.2.3 threshXhigh . . . 465
8.420.2.4 threshXlow . . . 465
8.420.2.5 uarfcn . . . 465
8.420.2.6 WCDMACellInfo . . . 465
8.421 nas_minBasedIMSI Struct Reference . . . 465
8.421.1 Detailed Description . . . 466
8.421.2 Field Documentation . . . 466
8.421.2.1 imsiM1112 . . . 466
8.421.2.2 imsiMS1 . . . 466
8.421.2.3 imsiMS2 . . . 466
8.421.2.4 mccM . . . 466
8.422 nas_MNRInfo Struct Reference . . . 466

8.422.1 Detailed Description . . . 466
8.422.2 Field Documentation . . . 467
8.422.2.1 mcc . . . 467
8.422.2.2 mnc . . . 467
8.422.2.3 rat . . . 467
8.423 nas_namName Struct Reference . . . 467
8.423.1 Detailed Description . . . 467
8.423.2 Field Documentation . . . 467
8.423.2.1 namName . . . 467
8.423.2.2 namNameLen . . . 467
8.424 nas_netSelectionPref Struct Reference . . . 467
8.424.1 Detailed Description . . . 468
8.424.2 Field Documentation . . . 468
8.424.2.1 mcc . . . 468
8.424.2.2 mnc . . . 468
8.424.2.3 netReg . . . 468
8.425 nas_NetworkStat1x Struct Reference . . . 468
8.425.1 Detailed Description . . . 468
8.425.2 Field Documentation . . . 470
8.425.2.1 ActSetCnt . . . 470
8.425.2.2 NeighborSetCnt . . . 470
8.425.2.3 pActPilotPNElements . . . 470
8.425.2.4 pNeighborSetPilotPN . . . 470
8.425.2.5 RX_EC_IO . . . 470
8.425.2.6 RX_PWR . . . 470
8.425.2.7 SO . . . 470
8.425.2.8 State . . . 470
8.425.2.9 TX_PWR . . . 470
8.426 nas_NetworkStatEVDO Struct Reference . . . 470
8.426.1 Detailed Description . . . 470
8.426.2 Field Documentation . . . 471
8.426.2.1 MACIndex . . . 472
8.426.2.2 PER . . . 472
8.426.2.3 PilotEnergy . . . 472
8.426.2.4 pSectorID . . . 472
8.426.2.5 RX_PWR . . . 472
8.426.2.6 SectorIDLen . . . 472
8.426.2.7 SNR . . . 472
8.426.2.8 State . . . 472
8.427 nas_nmrCellInfo Struct Reference . . . 472

8.427.1 Detailed Description . . . 472
8.427.2 Field Documentation . . . 473
8.427.2.1 nmrArfcn . . . 473
8.427.2.2 nmrBsic . . . 473
8.427.2.3 nmrCellID . . . 473
8.427.2.4 nmrLac . . . 473
8.427.2.5 nmrPlmn . . . 473
8.427.2.6 nmrRxLev . . . 473
8.428 nas_operatorNameString Struct Reference . . . 473
8.428.1 Detailed Description . . . 473
8.428.2 Field Documentation . . . 473
8.428.2.1 PLMNName . . . 473
8.429 nas_OperatorPLMNData Struct Reference . . . 473
8.429.1 Detailed Description . . . 474
8.429.2 Field Documentation . . . 474
8.429.2.1 lac1 . . . 474
8.429.2.2 lac2 . . . 474
8.429.2.3 mcc . . . 474
8.429.2.4 mnc . . . 474
8.429.2.5 PLMNRecID . . . 474
8.430 nas_operatorPLMNList Struct Reference . . . 474
8.430.1 Detailed Description . . . 474
8.430.2 Field Documentation . . . 475
8.430.2.1 numInstance . . . 475
8.430.2.2 PLMNData . . . 475
8.431 nas_PhyCaAggPcellInfo Struct Reference . . . 475
8.431.1 Detailed Description . . . 475
8.431.2 Field Documentation . . . 475
8.431.2.1 dl_bw_value . . . 475
8.431.2.2 freq . . . 475
8.431.2.3 iLTEbandValue . . . 476
8.431.2.4 pci . . . 476
8.431.2.5 TlvPresent . . . 476
8.432 nas_PhyCaAggScellDlBw Struct Reference . . . 476
8.432.1 Detailed Description . . . 476
8.432.2 Field Documentation . . . 476
8.432.2.1 dl_bw_value . . . 476
8.432.2.2 TlvPresent . . . 476
8.433 nas_PhyCaAggScellIndex Struct Reference . . . 476
8.433.1 Detailed Description . . . 476

8.433.2 Field Documentation . . . 477
8.433.2.1 scell_idx . . . 477
8.433.2.2 TlvPresent . . . 477
8.434 nas_PhyCaAggScellIndType Struct Reference . . . 477
8.434.1 Detailed Description . . . 477
8.434.2 Field Documentation . . . 477
8.434.2.1 freq . . . 477
8.434.2.2 pci . . . 477
8.434.2.3 scell_state . . . 477
8.434.2.4 TlvPresent . . . 477
8.435 nas_PhyCaAggScellInfo Struct Reference . . . 477
8.435.1 Detailed Description . . . 478
8.435.2 Field Documentation . . . 480
8.435.2.1 dl_bw_value . . . 480
8.435.2.2 freq . . . 480
8.435.2.3 iLTEbandValue . . . 480
8.435.2.4 pci . . . 480
8.435.2.5 scell_state . . . 480
8.435.2.6 TlvPresent . . . 480
8.436 nas_PilotSetData Struct Reference . . . 480
8.436.1 Detailed Description . . . 480
8.436.2 Field Documentation . . . 480
8.436.2.1 NumPilots . . . 480
8.436.2.2 pPilotSetInfo . . . 480
8.437 nas_PilotSetParams Struct Reference . . . 480
8.437.1 Detailed Description . . . 481
8.437.2 Field Documentation . . . 481
8.437.2.1 PilotPN . . . 481
8.437.2.2 PilotStrength . . . 481
8.437.2.3 PilotType . . . 481
8.438 nas_PLMNNetworkName Struct Reference . . . 481
8.438.1 Detailed Description . . . 481
8.438.2 Field Documentation . . . 481
8.438.2.1 numInstance . . . 481
8.438.2.2 PLMNNetName . . . 481
8.439 nas_PLMNNetworkNameData Struct Reference . . . 482
8.439.1 Detailed Description . . . 482
8.439.2 Field Documentation . . . 483
8.439.2.1 codingScheme . . . 483
8.439.2.2 countryInitials . . . 483

8.439.2.3 longName . . . 483
8.439.2.4 longNameLen . . . 483
8.439.2.5 longNameSpareBits . . . 483
8.439.2.6 shortName . . . 483
8.439.2.7 shortNameLen . . . 483
8.439.2.8 shortNameSpareBits . . . 483
8.440 nas_protocolSubtypeElement Struct Reference . . . 483
8.440.1 Detailed Description . . . 483
8.440.2 Field Documentation . . . 484
8.440.2.1 AccessMac . . . 484
8.440.2.2 AuthProt . . . 484
8.440.2.3 ControlMac . . . 484
8.440.2.4 EncryptProt . . . 484
8.440.2.5 ForwardMac . . . 484
8.440.2.6 IdleState . . . 484
8.440.2.7 KeyExchange . . . 484
8.440.2.8 MultDisc . . . 485
8.440.2.9 PhysicalLayer . . . 485
8.440.2.10 ReverseMac . . . 485
8.440.2.11 SecProt . . . 485
8.440.2.12 VirtStream . . . 485
8.441 nas_qaQmi3Gpp2TimeZone Struct Reference . . . 485
8.441.1 Detailed Description . . . 485
8.441.2 Field Documentation . . . 485
8.441.2.1 daylightSavings . . . 485
8.441.2.2 leapSeconds . . . 485
8.441.2.3 localTimeOffset . . . 485
8.442 nas_QmiNas3GppNetworkInfo Struct Reference . . . 485
8.442.1 Detailed Description . . . 486
8.442.2 Field Documentation . . . 486
8.442.2.1 Desription . . . 486
8.442.2.2 Forbidden . . . 486
8.442.2.3 InUse . . . 486
8.442.2.4 MCC . . . 486
8.442.2.5 MNC . . . 486
8.442.2.6 Preferred . . . 486
8.442.2.7 Roaming . . . 486
8.443 nas_QmiNas3GppNetworkRAT Struct Reference . . . 486
8.443.1 Detailed Description . . . 486
8.443.2 Field Documentation . . . 487

8.443.2.1 MCC . . . 487
8.443.2.2 MNC . . . 487
8.443.2.3 RAT . . . 487
8.444 nas_QmisNasPcsDigit Struct Reference . . . 487
8.444.1 Detailed Description . . . 487
8.444.2 Field Documentation . . . 487
8.444.2.1 includes_pcs_digit . . . 487
8.444.2.2 MCC . . . 487
8.444.2.3 MNC . . . 487
8.445 nas_RejectReasonTlv Struct Reference . . . 488
8.445.1 Detailed Description . . . 488
8.445.2 Field Documentation . . . 488
8.445.2.1 rejectCause . . . 488
8.445.2.2 serviceDomain . . . 488
8.445.2.3 TlvPresent . . . 488
8.446 nas_RFInfoTlv Struct Reference . . . 488
8.446.1 Detailed Description . . . 488
8.446.2 Field Documentation . . . 488
8.446.2.1 activeBandClass . . . 488
8.446.2.2 activeChannel . . . 488
8.446.2.3 radioInterface . . . 488
8.446.2.4 radioInterfaceSize . . . 488
8.446.2.5 TlvPresent . . . 489
8.447 nas_roamIndList Struct Reference . . . 489
8.447.1 Detailed Description . . . 489
8.447.2 Field Documentation . . . 489
8.447.2.1 numInstances . . . 489
8.447.2.2 radioInterface . . . 489
8.447.2.3 roamIndicator . . . 489
8.448 nas_RSRPThresh Struct Reference . . . 489
8.448.1 Detailed Description . . . 490
8.448.2 Field Documentation . . . 490
8.448.2.1 pRSRPThresList . . . 490
8.448.2.2 RSRPThresListLen . . . 490
8.449 nas_rsrqInformation Struct Reference . . . 490
8.449.1 Detailed Description . . . 490
8.449.2 Field Documentation . . . 490
8.449.2.1 radioIf . . . 491
8.449.2.2 rsrq . . . 491
8.450 nas_RSRQThresh Struct Reference . . . 491

8.450.1 Detailed Description . . . 491
8.450.2 Field Documentation . . . 491
8.450.2.1 pRSRQThresList . . . 491
8.450.2.2 RSRQThresListLen . . . 491
8.451 nas_RSSIThresh Struct Reference . . . 491
8.451.1 Detailed Description . . . 491
8.451.2 Field Documentation . . . 492
8.451.2.1 pRSSIThresList . . . 492
8.451.2.2 RSSIThresListLen . . . 492
8.452 nas_rxInfo Struct Reference . . . 492
8.452.1 Detailed Description . . . 492
8.452.2 Field Documentation . . . 493
8.452.2.1 ecio . . . 493
8.452.2.2 isRadioTuned . . . 493
8.452.2.3 phase . . . 493
8.452.2.4 rscp . . . 493
8.452.2.5 rsrp . . . 493
8.452.2.6 rxPower . . . 493
8.453 nas_RxSigInfo Struct Reference . . . 493
8.453.1 Detailed Description . . . 493
8.453.2 Field Documentation . . . 494
8.453.2.1 isRadioTuned . . . 494
8.453.2.2 rsrp . . . 494
8.453.2.3 rxChainIndex . . . 494
8.453.2.4 rxPower . . . 494
8.454 nas_rxSignalStrengthListElement Struct Reference . . . 494
8.454.1 Detailed Description . . . 494
8.454.2 Field Documentation . . . 495
8.454.2.1 radioIf . . . 495
8.454.2.2 rxSignalStrength . . . 495
8.455 nas_SccRxInfo Struct Reference . . . 495
8.455.1 Detailed Description . . . 495
8.455.2 Field Documentation . . . 495
8.455.2.1 numInstances . . . 495
8.455.2.2 rsrq . . . 496
8.455.2.3 sigInfo . . . 496
8.455.2.4 snr . . . 496
8.455.2.5 TlvPresent . . . 496
8.456 nas_serviceProviderName Struct Reference . . . 496
8.456.1 Detailed Description . . . 496

8.456.2 Field Documentation . . . 496
8.456.2.1 displayCondition . . . 496
8.456.2.2 spn . . . 496
8.456.2.3 spnLength . . . 496
8.457 nas_servSystem Struct Reference . . . 496
8.457.1 Detailed Description . . . 496
8.457.2 Field Documentation . . . 498
8.457.2.1 csAttachState . . . 498
8.457.2.2 numRadioInterfaces . . . 498
8.457.2.3 psAttachState . . . 498
8.457.2.4 radioInterface . . . 498
8.457.2.5 regState . . . 498
8.457.2.6 selNetwork . . . 498
8.458 nas_sidNid Struct Reference . . . 498
8.458.1 Detailed Description . . . 498
8.458.2 Field Documentation . . . 498
8.458.2.1 nid . . . 498
8.458.2.2 sid . . . 498
8.459 nas_SignalStrengthTlv Struct Reference . . . 498
8.459.1 Detailed Description . . . 498
8.459.2 Field Documentation . . . 499
8.459.2.1 radioInterface . . . 499
8.459.2.2 signalStrength . . . 499
8.459.2.3 TlvPresent . . . 499
8.460 nas_SLQSSignalStrengthsIndReq Struct Reference . . . 499
8.460.1 Detailed Description . . . 499
8.460.2 Field Documentation . . . 499
8.460.2.1 ecioDelta . . . 499
8.460.2.2 ecioThresholdList . . . 500
8.460.2.3 ecioThresholdListLen . . . 500
8.460.2.4 ioDelta . . . 500
8.460.2.5 lteRsrpDelta . . . 500
8.460.2.6 lteSnrDelta . . . 500
8.460.2.7 rsrqDelta . . . 500
8.460.2.8 rxSignalStrengthDelta . . . 500
8.460.2.9 sinrDelta . . . 500
8.460.2.10 sinrThresholdList . . . 500
8.460.2.11 sinrThresholdListLen . . . 500
8.461 nas_SLQSSignalStrengthsInformation Struct Reference . . . 500
8.461.1 Detailed Description . . . 500

8.461.2 Field Documentation . . . 500
8.461.2.1 ecioInfo . . . 500
8.461.2.2 errorRateInfo . . . 500
8.461.2.3 io . . . 500
8.461.2.4 lteRsrpinfo . . . 501
8.461.2.5 lteSnrinfo . . . 501
8.461.2.6 rsrqInfo . . . 501
8.461.2.7 rxSignalStrengthInfo . . . 501
8.461.2.8 sinr . . . 501
8.462 nas_SLQSSignalStrengthsTlv Struct Reference . . . 501
8.462.1 Detailed Description . . . 501
8.462.2 Field Documentation . . . 501
8.462.2.1 sSLQSSignalStrengthsInfo . . . 501
8.462.2.2 TlvPresent . . . 501
8.463 nas_SrvStatusInfo Struct Reference . . . 501
8.463.1 Detailed Description . . . 501
8.463.2 Field Documentation . . . 502
8.463.2.1 isPrefDataPath . . . 502
8.463.2.2 srvStatus . . . 502
8.464 nas_sysInfoCommon Struct Reference . . . 502
8.464.1 Detailed Description . . . 502
8.464.2 Field Documentation . . . 504
8.464.2.1 isSysForbidden . . . 504
8.464.2.2 isSysForbiddenValid . . . 504
8.464.2.3 roamStatus . . . 504
8.464.2.4 roamStatusValid . . . 504
8.464.2.5 srvCapability . . . 504
8.464.2.6 srvCapabilityValid . . . 504
8.464.2.7 srvDomain . . . 504
8.464.2.8 srvDomainValid . . . 504
8.465 nas_TDSCDMAECIOThresh Struct Reference . . . 504
8.465.1 Detailed Description . . . 504
8.465.2 Field Documentation . . . 504
8.465.2.1 pTDSCDMAECIOThreshList . . . 504
8.465.2.2 TDSCDMAECIOThreshListLen . . . 504
8.466 nas_TDSCDMARSCPThresh Struct Reference . . . 505
8.466.1 Detailed Description . . . 505
8.466.2 Field Documentation . . . 505
8.466.2.1 pTDSCDMARSCPThreshList . . . 505
8.466.2.2 TDSCDMARSCPThreshListLen . . . 505

8.467 nas_TDSCDMARSSIThresh Struct Reference . . . 505
8.467.1 Detailed Description . . . 505
8.467.2 Field Documentation . . . 505
8.467.2.1 pTDSCDMARSSIThreshList . . . 505
8.467.2.2 TDSCDMARSSIThreshListLen . . . 506
8.468 nas_TDSCDMASINRCONFThresh Struct Reference . . . 506
8.468.1 Detailed Description . . . 506
8.468.2 Field Documentation . . . 506
8.468.2.1 pTDSCDMASINRCONFThreshList . . . 506
8.468.2.2 TDSCDMASINRCONFThreshListLen . . . 506
8.469 nas_TDSCDMASINRThresh Struct Reference . . . 506
8.469.1 Detailed Description . . . 506
8.469.2 Field Documentation . . . 506
8.469.2.1 pTDSCDMASINRThreshList . . . 506
8.469.2.2 TDSCDMASINRThreshListLen . . . 507
8.470 nas_timeInfo Struct Reference . . . 507
8.470.1 Detailed Description . . . 507
8.470.2 Field Documentation . . . 508
8.470.2.1 day . . . 508
8.470.2.2 dayLtSavingAdj . . . 508
8.470.2.3 dayOfWeek . . . 508
8.470.2.4 hour . . . 508
8.470.2.5 minute . . . 508
8.470.2.6 month . . . 508
8.470.2.7 radioInterface . . . 508
8.470.2.8 second . . . 508
8.470.2.9 timeZone . . . 508
8.470.2.10 TlvPresent . . . 508
8.470.2.11 year . . . 508
8.471 nas_trueIMSI Struct Reference . . . 508
8.471.1 Detailed Description . . . 509
8.471.2 Field Documentation . . . 509
8.471.2.1 imsiT1112 . . . 509
8.471.2.2 imsiTaddrNum . . . 509
8.471.2.3 imsiTS1 . . . 509
8.471.2.4 imsiTS2 . . . 509
8.471.2.5 mccT . . . 509
8.472 nas_txInfo Struct Reference . . . 509
8.472.1 Detailed Description . . . 509
8.472.2 Field Documentation . . . 510

8.472.2.1 isInTraffic . . . 510
8.472.2.2 txPower . . . 510
8.473 nas_UMTSInfo Struct Reference . . . 510
8.473.1 Detailed Description . . . 510
8.473.2 Field Documentation . . . 511
8.473.2.1 cellID . . . 511
8.473.2.2 ecio . . . 511
8.473.2.3 geranInst . . . 511
8.473.2.4 GeranInstInfo . . . 511
8.473.2.5 lac . . . 511
8.473.2.6 plmn . . . 511
8.473.2.7 psc . . . 511
8.473.2.8 rscp . . . 511
8.473.2.9 uarfcn . . . 511
8.473.2.10 umtsInst . . . 512
8.473.2.11 UMTSInstInfo . . . 512
8.474 nas_UMTSinstInfo Struct Reference . . . 512
8.474.1 Detailed Description . . . 512
8.474.2 Field Documentation . . . 512
8.474.2.1 umtsEcio . . . 512
8.474.2.2 umtsPsc . . . 512
8.474.2.3 umtsRscp . . . 512
8.474.2.4 umtsUarfcn . . . 512
8.475 nas_umtsLTENbrCell Struct Reference . . . 512
8.475.1 Detailed Description . . . 513
8.475.2 Field Documentation . . . 513
8.475.2.1 cellIsTDD . . . 513
8.475.2.2 earfcn . . . 513
8.475.2.3 pci . . . 513
8.475.2.4 rsrp . . . 513
8.475.2.5 rsrq . . . 513
8.475.2.6 srxlev . . . 513
8.476 nas_UniversalTime Struct Reference . . . 513
8.476.1 Detailed Description . . . 514
8.476.2 Field Documentation . . . 514
8.476.2.1 day . . . 514
8.476.2.2 dayOfWeek . . . 514
8.476.2.3 hour . . . 514
8.476.2.4 minute . . . 514
8.476.2.5 month . . . 514

8.476.2.6 second . . . 514
8.476.2.7 year . . . 514
8.477 nas_wcdmaCellInfo Struct Reference . . . 514
8.477.1 Detailed Description . . . 515
8.477.2 Field Documentation . . . 515
8.477.2.1 cpich_ecno . . . 515
8.477.2.2 cpich_rscp . . . 515
8.477.2.3 psc . . . 515
8.477.2.4 srxlev . . . 515
8.478 nas_WCDMAECIOThresh Struct Reference . . . 515
8.478.1 Detailed Description . . . 515
8.478.2 Field Documentation . . . 516
8.478.2.1 pWCDMAECIOThreshList . . . 516
8.478.2.2 WCDMAECIOThreshListLen . . . 516
8.479 nas_WCDMAInfoLTENeighborCell Struct Reference . . . 516
8.479.1 Detailed Description . . . 516
8.479.2 Field Documentation . . . 517
8.479.2.1 UMTSLTENbrCell . . . 517
8.479.2.2 umtsLTENbrCellLen . . . 517
8.479.2.3 wcdmaRRCState . . . 517
8.480 nas_WCDMARSSIThresh Struct Reference . . . 517
8.480.1 Detailed Description . . . 517
8.480.2 Field Documentation . . . 517
8.480.2.1 pWCDMARSSIThreshList . . . 517
8.480.2.2 WCDMARSSIThreshListLen . . . 517
8.481 nas_WCDMASysInfo Struct Reference . . . 517
8.481.1 Detailed Description . . . 518
8.481.2 Field Documentation . . . 520
8.481.2.1 cellId . . . 520
8.481.2.2 cellIdValid . . . 520
8.481.2.3 hsCallStatus . . . 520
8.481.2.4 hsCallStatusValid . . . 520
8.481.2.5 hsInd . . . 520
8.481.2.6 hsIndValid . . . 520
8.481.2.7 lac . . . 520
8.481.2.8 lacValid . . . 520
8.481.2.9 MCC . . . 520
8.481.2.10 MNC . . . 520
8.481.2.11 networkIdValid . . . 520
8.481.2.12 psc . . . 520

8.481.2.13 pscValid . . . 520
8.481.2.14 regRejectInfoValid . . . 521
8.481.2.15 rejCause . . . 521
8.481.2.16 rejectSrvDomain . . . 521
8.481.2.17 sysInfoWCDMA . . . 521
8.482 nas_wcdmaUARFCN Struct Reference . . . 521
8.482.1 Detailed Description . . . 521
8.482.2 Field Documentation . . . 521
8.482.2.1 status . . . 521
8.482.2.2 uarfcn . . . 521
8.483 NASBandPreferenceTlv Struct Reference . . . 521
8.483.1 Field Documentation . . . 521
8.483.1.1 band_pref . . . 521
8.483.1.2 TlvPresent . . . 521
8.484 nasCellLocationInfoResp Struct Reference . . . 521
8.484.1 Detailed Description . . . 522
8.484.2 Field Documentation . . . 522
8.484.2.1 pCDMAInfo . . . 522
8.484.2.2 pGERANInfo . . . 522
8.484.2.3 pLTEInfoInterfreq . . . 522
8.484.2.4 pLTEInfoIntrafreq . . . 522
8.484.2.5 pLTEInfoNeighboringGSM . . . 523
8.484.2.6 pLTEInfoNeighboringWCDMA . . . 523
8.484.2.7 pUMTSCellID . . . 523
8.484.2.8 pUMTSInfo . . . 523
8.484.2.9 pWCDMAInfoLTENeighborCell . . . 523
8.485 NASEmergencyModeTlv Struct Reference . . . 523
8.485.1 Field Documentation . . . 523
8.485.1.1 EmerMode . . . 523
8.485.1.2 TlvPresent . . . 523
8.486 nasGet3GPP2SubscriptionInfoReq Struct Reference . . . 523
8.486.1 Detailed Description . . . 523
8.486.2 Field Documentation . . . 523
8.486.2.1 namID . . . 523
8.487 nasGet3GPP2SubscriptionInfoResp Struct Reference . . . 523
8.487.1 Detailed Description . . . 524
8.487.2 Field Documentation . . . 524
8.487.2.1 pCDMAChannel . . . 524
8.487.2.2 pDirNum . . . 524
8.487.2.3 pHomeSIDNID . . . 524

8.487.2.4 pMinBasedIMSI . . . 524
8.487.2.5 pNAMNameInfo . . . 524
8.487.2.6 pTrueIMSI . . . 524
8.488 nasGetHDRColorCodeResp Struct Reference . . . 524
8.488.1 Detailed Description . . . 524
8.488.2 Field Documentation . . . 525
8.488.2.1 pColorCode . . . 525
8.489 nasGetLTECphyCa Struct Reference . . . 525
8.489.1 Field Documentation . . . 525
8.489.1.1 sPhyCaAggPcellInfo . . . 525
8.489.1.2 sPhyCaAggScellDlBw . . . 525
8.489.1.3 sPhyCaAggScellIndex . . . 525
8.489.1.4 sPhyCaAggScellIndType . . . 525
8.489.1.5 sPhyCaAggScellInfo . . . 525
8.490 NasGetLTECphyCaInfo Struct Reference . . . 525
8.490.1 Field Documentation . . . 525
8.490.1.1 PhyCaAggPcellInfo . . . 525
8.490.1.2 PhyCaAggScellDlBw . . . 525
8.490.1.3 PhyCaAggScellIndex . . . 525
8.490.1.4 PhyCaAggScellIndType . . . 526
8.490.1.5 PhyCaAggScellInfo . . . 526
8.491 nasGetLTECphyCaResp Struct Reference . . . 526
8.491.1 Field Documentation . . . 526
8.491.1.1 pPhyCaAggPcellInfo . . . 526
8.491.1.2 pPhyCaAggScellDlBw . . . 526
8.491.1.3 pPhyCaAggScellIndex . . . 526
8.491.1.4 pPhyCaAggScellIndType . . . 526
8.491.1.5 pPhyCaAggScellInfo . . . 526
8.492 nasGetSigInfoResp Struct Reference . . . 526
8.492.1 Detailed Description . . . 526
8.492.2 Field Documentation . . . 527
8.492.2.1 pCDMASSInfo . . . 527
8.492.2.2 pGSMSSInfo . . . 527
8.492.2.3 pHDRSSInfo . . . 527
8.492.2.4 pLTESSInfo . . . 527
8.492.2.5 pTDSCDMASigInfoExt . . . 527
8.492.2.6 pTDSCDMASigInfoRscp . . . 527
8.492.2.7 pWCDMASSInfo . . . 527
8.493 nasGetSysInfoResp Struct Reference . . . 527
8.493.1 Detailed Description . . . 528

8.493.2 Field Documentation . . . 529
8.493.2.1 pAddCDMASysInfo . . . 529
8.493.2.2 pAddGSMSysInfo . . . 529
8.493.2.3 pAddHDRSysInfo . . . 529
8.493.2.4 pAddLTESysInfo . . . 529
8.493.2.5 pAddWCDMASysInfo . . . 529
8.493.2.6 pCDMASrvStatusInfo . . . 529
8.493.2.7 pCDMASysInfo . . . 529
8.493.2.8 pGSMCallBarringSysInfo . . . 529
8.493.2.9 pGSMCipherDomainSysInfo . . . 529
8.493.2.10 pGSMSrvStatusInfo . . . 529
8.493.2.11 pGSMSysInfo . . . 529
8.493.2.12 pHDRSrvStatusInfo . . . 529
8.493.2.13 pHDRSysInfo . . . 530
8.493.2.14 pLTESrvStatusInfo . . . 530
8.493.2.15 pLTESysInfo . . . 530
8.493.2.16 pLTEVoiceSupportSysInfo . . . 530
8.493.2.17 pWCDMACallBarringSysInfo . . . 530
8.493.2.18 pWCDMACipherDomainSysInfo . . . 530
8.493.2.19 pWCDMASrvStatusInfo . . . 530
8.493.2.20 pWCDMASysInfo . . . 530
8.494 nasGetTxRxInfoReq Struct Reference . . . 530
8.494.1 Detailed Description . . . 530
8.494.2 Field Documentation . . . 530
8.494.2.1 radio_if . . . 530
8.495 nasGetTxRxInfoResp Struct Reference . . . 530
8.495.1 Detailed Description . . . 531
8.495.2 Field Documentation . . . 531
8.495.2.1 pRXChain0Info . . . 531
8.495.2.2 pRXChain1Info . . . 531
8.495.2.3 pTXInfo . . . 531
8.496 NASGWAcqOrderPrefTlv Struct Reference . . . 531
8.496.1 Field Documentation . . . 531
8.496.1.1 GWAcqOrderPref . . . 531
8.496.1.2 TlvPresent . . . 531
8.497 nasIndicationRegisterReq Struct Reference . . . 531
8.497.1 Detailed Description . . . 532
8.497.2 Field Documentation . . . 534
8.497.2.1 pDDTMInd . . . 534
8.497.2.2 pDualStandByPrefInd . . . 534

8.497.2.3 pErrorRateInd . . . 534
8.497.2.4 pHDRNewUATIAssInd . . . 534
8.497.2.5 pHDRSessionCloseInd . . . 534
8.497.2.6 pLTECphyCa . . . 534
8.497.2.7 pManagedRoamingInd . . . 534
8.497.2.8 pNetworkRejectInd . . . 534
8.497.2.9 pNetworkTimeInd . . . 534
8.497.2.10 pServingSystemInd . . . 534
8.497.2.11 pSignalStrengthInd . . . 534
8.497.2.12 pSubscriptionInfoInd . . . 534
8.497.2.13 pSuppressSysInfoInd . . . 534
8.497.2.14 pSysInfoInd . . . 534
8.497.2.15 pSystemSelectionInd . . . 534
8.498 nasInitNetworkReg Struct Reference . . . 534
8.498.1 Detailed Description . . . 535
8.498.2 Field Documentation . . . 535
8.498.2.1 pChangeDuration . . . 535
8.498.2.2 pMncPcsDigitStatus . . . 535
8.498.2.3 pMNRInfo . . . 535
8.498.2.4 regAction . . . 535
8.499 NASLTEBandPreferenceTlv Struct Reference . . . 535
8.499.1 Field Documentation . . . 536
8.499.1.1 LTEBandPref . . . 536
8.499.1.2 TlvPresent . . . 536
8.500 NASLteNasReleaseInfoTlv Struct Reference . . . 536
8.500.1 Field Documentation . . . 536
8.500.1.1 nas_major . . . 536
8.500.1.2 nas_minor . . . 536
8.500.1.3 nas_release . . . 536
8.500.1.4 TlvPresent . . . 536
8.501 NASModePreferenceTlv Struct Reference . . . 536
8.501.1 Field Documentation . . . 536
8.501.1.1 ModePref . . . 536
8.501.1.2 TlvPresent . . . 536
8.502 NASNetSelPreferenceTlv Struct Reference . . . 536
8.502.1 Field Documentation . . . 536
8.502.1.1 NetSelPref . . . 536
8.502.1.2 TlvPresent . . . 536
8.503 nasNetworkReject Struct Reference . . . 537
8.503.1 Detailed Description . . . 537

8.503.2 Field Documentation . . . 539
8.503.2.1 pCsgId . . . 539
8.503.2.2 pPlmnId . . . 539
8.503.2.3 radioIf . . . 539
8.503.2.4 rejectCause . . . 539
8.503.2.5 serviceDomain . . . 539
8.504 nasNetworkTime Struct Reference . . . 539
8.504.1 Detailed Description . . . 539
8.504.2 Field Documentation . . . 540
8.504.2.1 pDayltSavAdj . . . 540
8.504.2.2 pRadioInterface . . . 540
8.504.2.3 pTimeZone . . . 540
8.504.2.4 universalTime . . . 540
8.505 nasOperatorNameResp Struct Reference . . . 540
8.505.1 Detailed Description . . . 540
8.505.2 Field Documentation . . . 541
8.505.2.1 pNITZInformation . . . 541
8.505.2.2 pOperatorNameString . . . 541
8.505.2.3 pOperatorPLMNList . . . 541
8.505.2.4 pPLMNNetworkName . . . 541
8.505.2.5 pSrvcProviderName . . . 541
8.506 NASOTAMessageTlv Struct Reference . . . 541
8.506.1 Field Documentation . . . 541
8.506.1.1 data_buf . . . 541
8.506.1.2 data_len . . . 541
8.506.1.3 message_type . . . 541
8.506.1.4 TlvPresent . . . 541
8.507 NASPhyCaAggPcellInfo Struct Reference . . . 541
8.507.1 Detailed Description . . . 541
8.507.2 Field Documentation . . . 542
8.507.2.1 dl_bw_value . . . 542
8.507.2.2 freq . . . 542
8.507.2.3 iLTEbandValue . . . 542
8.507.2.4 pci . . . 542
8.507.2.5 TlvPresent . . . 542
8.508 NASPhyCaAggScellDlBw Struct Reference . . . 542
8.508.1 Detailed Description . . . 542
8.508.2 Field Documentation . . . 542
8.508.2.1 dl_bw_value . . . 543
8.508.2.2 TlvPresent . . . 543

8.509 NASPhyCaAggScellIndex Struct Reference . . . 543
8.509.1 Detailed Description . . . 543
8.509.2 Field Documentation . . . 543
8.509.2.1 scell_idx . . . 543
8.509.2.2 TlvPresent . . . 543
8.510 NASPhyCaAggScellIndType Struct Reference . . . 543
8.510.1 Detailed Description . . . 543
8.510.2 Field Documentation . . . 544
8.510.2.1 freq . . . 544
8.510.2.2 pci . . . 544
8.510.2.3 scell_state . . . 544
8.510.2.4 TlvPresent . . . 544
8.511 NASPhyCaAggScellInfo Struct Reference . . . 544
8.511.1 Detailed Description . . . 544
8.511.2 Field Documentation . . . 545
8.511.2.1 dl_bw_value . . . 545
8.511.2.2 freq . . . 545
8.511.2.3 iLTEbandValue . . . 545
8.511.2.4 pci . . . 545
8.511.2.5 scell_state . . . 545
8.511.2.6 TlvPresent . . . 545
8.512 nasPLMNNameReq Struct Reference . . . 545
8.512.1 Detailed Description . . . 545
8.512.2 Field Documentation . . . 546
8.512.2.1 mcc . . . 546
8.512.2.2 mnc . . . 546
8.512.2.3 pMncPcsStatus . . . 546
8.513 nasPLMNNameResp Struct Reference . . . 546
8.513.1 Detailed Description . . . 546
8.513.2 Field Documentation . . . 548
8.513.2.1 longName . . . 548
8.513.2.2 longNameCI . . . 548
8.513.2.3 longNameEn . . . 548
8.513.2.4 longNameLen . . . 548
8.513.2.5 longNameSB . . . 548
8.513.2.6 shortName . . . 548
8.513.2.7 shortNameCI . . . 548
8.513.2.8 shortNameEn . . . 548
8.513.2.9 shortNameLen . . . 548
8.513.2.10 shortNameSB . . . 548

8.513.2.11 spn . . . 548
8.513.2.12 spnEncoding . . . 548
8.513.2.13 spnLength . . . 548
8.514 NASPRLPreferenceTlv Struct Reference . . . 548
8.514.1 Field Documentation . . . 548
8.514.1.1 PRLPref . . . 548
8.514.1.2 TlvPresent . . . 548
8.515 NASQmiCbkNasSwiOTAMessageInd Struct Reference . . . 548
8.515.1 Field Documentation . . . 549
8.515.1.1 nasRelInfoTlv . . . 549
8.515.1.2 otaMsgTlv . . . 549
8.515.1.3 timeTlv . . . 549
8.516 NASQmiCbkNasSystemSelPrefInd Struct Reference . . . 549
8.516.1 Field Documentation . . . 549
8.516.1.1 BPTlv . . . 549
8.516.1.2 EMTlv . . . 549
8.516.1.3 GWAOPTlv . . . 549
8.516.1.4 LBPTlv . . . 549
8.516.1.5 MPTlv . . . 549
8.516.1.6 NSPTlv . . . 549
8.516.1.7 PRLPTlv . . . 549
8.516.1.8 RPTlv . . . 549
8.516.1.9 SDPTlv . . . 549
8.517 NASRoamPreferenceTlv Struct Reference . . . 549
8.517.1 Field Documentation . . . 550
8.517.1.1 RoamPref . . . 550
8.517.1.2 TlvPresent . . . 550
8.518 NASServDomainPrefTlv Struct Reference . . . 550
8.518.1 Field Documentation . . . 550
8.518.1.1 SrvDomainPref . . . 550
8.518.1.2 TlvPresent . . . 550
8.519 NASServingSystemInfo Struct Reference . . . 550
8.519.1 Detailed Description . . . 550
8.519.2 Field Documentation . . . 551
8.519.2.1 csAttachState . . . 551
8.519.2.2 hdrPersonality . . . 551
8.519.2.3 psAttachState . . . 551
8.519.2.4 radioInterfaceList . . . 551
8.519.2.5 radioInterfaceNo . . . 551
8.519.2.6 registrationState . . . 551

8.519.2.7 selectedNetwork . . . 551
8.520 nasSigInfo Struct Reference . . . 551
8.520.1 Detailed Description . . . 552
8.520.2 Field Documentation . . . 552
8.520.2.1 pCDMASigInfo . . . 552
8.520.2.2 pGSMSigInfo . . . 552
8.520.2.3 pHDRSigInfo . . . 552
8.520.2.4 pLTESigInfo . . . 552
8.520.2.5 pRscp . . . 552
8.520.2.6 pTDSCDMASigInfoExt . . . 552
8.520.2.7 pWCDMASigInfo . . . 552
8.521 nasSwiGetChannelLockResp Struct Reference . . . 552
8.521.1 Detailed Description . . . 553
8.521.2 Field Documentation . . . 553
8.521.2.1 pLteEARFCN . . . 553
8.521.2.2 pLtePCI . . . 553
8.521.2.3 pWcdmaUARFCN . . . 553
8.522 NasSwiIndReg Struct Reference . . . 553
8.522.1 Detailed Description . . . 553
8.522.2 Field Documentation . . . 554
8.522.2.1 gsmUmtsDl . . . 554
8.522.2.2 gsmUmtsUl . . . 554
8.522.2.3 lteEmmDl . . . 554
8.522.2.4 lteEmmUl . . . 554
8.522.2.5 lteEsmDl . . . 554
8.522.2.6 lteEsmUl . . . 554
8.522.2.7 pRankIndicatorInd . . . 554
8.522.2.8 pTimer . . . 554
8.523 nasSwiSetChannelLockReq Struct Reference . . . 554
8.523.1 Detailed Description . . . 555
8.523.2 Field Documentation . . . 555
8.523.2.1 pLteEARFCN . . . 555
8.523.2.2 pLtePCI . . . 555
8.523.2.3 pWcdmaUARFCN . . . 555
8.524 nasSysInfo Struct Reference . . . 555
8.524.1 Detailed Description . . . 556
8.524.2 Field Documentation . . . 557
8.524.2.1 pAddCDMASysInfo . . . 557
8.524.2.2 pAddGSMSysInfo . . . 557
8.524.2.3 pAddHDRSysInfo . . . 557

8.524.2.4 pAddLTESysInfo . . . 557
8.524.2.5 pAddWCDMASysInfo . . . 557
8.524.2.6 pCDMASrvStatusInfo . . . 557
8.524.2.7 pCDMASysInfo . . . 557
8.524.2.8 pGSMCallBarringSysInfo . . . 557
8.524.2.9 pGSMCipherDomainSysInfo . . . 557
8.524.2.10 pGSMSrvStatusInfo . . . 557
8.524.2.11 pGSMSysInfo . . . 558
8.524.2.12 pHDRSrvStatusInfo . . . 558
8.524.2.13 pHDRSysInfo . . . 558
8.524.2.14 pLTESrvStatusInfo . . . 558
8.524.2.15 pLTESysInfo . . . 558
8.524.2.16 pLTEVoiceSupportSysInfo . . . 558
8.524.2.17 pSysInfoNoChange . . . 558
8.524.2.18 pWCDMACallBarringSysInfo . . . 558
8.524.2.19 pWCDMACipherDomainSysInfo . . . 558
8.524.2.20 pWCDMASrvStatusInfo . . . 558
8.524.2.21 pWCDMASysInfo . . . 558
8.525 NASTimeInfoTlv Struct Reference . . . 558
8.525.1 Field Documentation . . . 558
8.525.1.1 time . . . 558
8.525.1.2 TlvPresent . . . 558
8.526 nasTimers Struct Reference . . . 558
8.526.1 Detailed Description . . . 558
8.526.2 Field Documentation . . . 559
8.526.2.1 t3396_apn . . . 559
8.526.2.2 t3396_plmn_id . . . 559
8.526.2.3 t3396_val . . . 559
8.527 netSelectionPref Struct Reference . . . 559
8.527.1 Detailed Description . . . 559
8.527.2 Field Documentation . . . 559
8.527.2.1 mcc . . . 559
8.527.2.2 mnc . . . 559
8.527.2.3 netReg . . . 559
8.528 NetStats Struct Reference . . . 560
8.528.1 Detailed Description . . . 560
8.528.2 Field Documentation . . . 560
8.528.2.1 rx_bytes . . . 560
8.528.2.2 rx_errors . . . 560
8.528.2.3 rx_overflows . . . 560

8.528.2.4 rx_packets . . . 560
8.528.2.5 tx_bytes . . . 561
8.528.2.6 tx_errors . . . 561
8.528.2.7 tx_overflows . . . 561
8.528.2.8 tx_packets . . . 561
8.529 NetworkDebugResp Struct Reference . . . 561
8.529.1 Detailed Description . . . 561
8.529.2 Field Documentation . . . 561
8.529.2.1 pDataStatusDetail . . . 561
8.529.2.2 pDeviceConfigDetail . . . 561
8.529.2.3 pNetworkStat1x . . . 561
8.529.2.4 pNetworkStatEVDO . . . 561
8.529.2.5 pObjectVer . . . 561
8.530 NetworkStat1x Struct Reference . . . 562
8.530.1 Detailed Description . . . 562
8.530.2 Field Documentation . . . 563
8.530.2.1 ActSetCnt . . . 563
8.530.2.2 NeighborSetCnt . . . 563
8.530.2.3 pActPilotPNElements . . . 563
8.530.2.4 pNeighborSetPilotPN . . . 563
8.530.2.5 RX_EC_IO . . . 563
8.530.2.6 RX_PWR . . . 563
8.530.2.7 SO . . . 563
8.530.2.8 State . . . 563
8.530.2.9 TX_PWR . . . 564
8.531 NetworkStatEVDO Struct Reference . . . 564
8.531.1 Detailed Description . . . 564
8.531.2 Field Documentation . . . 565
8.531.2.1 MACIndex . . . 565
8.531.2.2 PER . . . 565
8.531.2.3 PilotEnergy . . . 565
8.531.2.4 pSectorID . . . 565
8.531.2.5 RX_PWR . . . 565
8.531.2.6 SectorIDLen . . . 565
8.531.2.7 SNR . . . 565
8.531.2.8 State . . . 565
8.532 newMTMessageTlv Struct Reference . . . 565
8.532.1 Detailed Description . . . 565
8.532.2 Field Documentation . . . 566
8.532.2.1 MTMessageInfo . . . 566

8.532.2.2 TlvPresent . . . 566
8.533 newPwdData Struct Reference . . . 566
8.533.1 Detailed Description . . . 566
8.533.2 Field Documentation . . . 566
8.533.2.1 newPwd . . . 566
8.533.2.2 newPwdAgain . . . 566
8.534 nmrCellInfo Struct Reference . . . 566
8.534.1 Detailed Description . . . 567
8.534.2 Field Documentation . . . 567
8.534.2.1 nmrArfcn . . . 567
8.534.2.2 nmrBsic . . . 567
8.534.2.3 nmrCellID . . . 567
8.534.2.4 nmrLac . . . 568
8.534.2.5 nmrPlmn . . . 568
8.534.2.6 nmrRxLev . . . 568
8.535 NSSAudioCtrl Struct Reference . . . 568
8.535.1 Detailed Description . . . 568
8.535.2 Field Documentation . . . 568
8.535.2.1 downLink . . . 568
8.535.2.2 upLink . . . 568
8.536 NWProfile Struct Reference . . . 568
8.536.1 Detailed Description . . . 568
8.536.2 Field Documentation . . . 568
8.536.2.1 pProfSz . . . 569
8.536.2.2 pProfValues . . . 569
8.536.2.3 tech . . . 569
8.537 omaDmConfigTlv Struct Reference . . . 569
8.537.1 Detailed Description . . . 569
8.537.2 Field Documentation . . . 569
8.537.2.1 alertmsg . . . 569
8.537.2.2 alertmsglength . . . 569
8.537.2.3 state . . . 569
8.537.2.4 userInputReq . . . 569
8.537.2.5 userInputTimeout . . . 569
8.538 omaDmConfigTlvExt Struct Reference . . . 570
8.538.1 Detailed Description . . . 570
8.538.2 Field Documentation . . . 571
8.538.2.1 alertmsg . . . 571
8.538.2.2 alertmsglength . . . 571
8.538.2.3 state . . . 571

8.538.2.4 userInputReq . . . 571
8.538.2.5 userInputTimeout . . . 571
8.539 omaDmFotaTlv Struct Reference . . . 571
8.539.1 Detailed Description . . . 571
8.539.2 Field Documentation . . . 573
8.539.2.1 description . . . 573
8.539.2.2 descriptionlength . . . 573
8.539.2.3 fwdloadsize . . . 573
8.539.2.4 fwloadComplete . . . 573
8.539.2.5 namelength . . . 573
8.539.2.6 package_name . . . 573
8.539.2.7 sessionType . . . 573
8.539.2.8 severity . . . 573
8.539.2.9 state . . . 573
8.539.2.10 updateCompleteStatus . . . 573
8.539.2.11 userInputReq . . . 573
8.539.2.12 userInputTimeout . . . 573
8.539.2.13 version . . . 573
8.539.2.14 versionlength . . . 573
8.540 omaDmFotaTlvExt Struct Reference . . . 573
8.540.1 Detailed Description . . . 573
8.540.2 Field Documentation . . . 575
8.540.2.1 description . . . 575
8.540.2.2 descriptionlength . . . 575
8.540.2.3 fumoResultCode . . . 575
8.540.2.4 namelength . . . 575
8.540.2.5 package_name . . . 575
8.540.2.6 packageSize . . . 575
8.540.2.7 receivedBytes . . . 575
8.540.2.8 reserved . . . 575
8.540.2.9 state . . . 575
8.540.2.10 userInputTimeout . . . 575
8.540.2.11 version . . . 575
8.540.2.12 versionlength . . . 575
8.541 omaDmNotificationsTlv Struct Reference . . . 575
8.541.1 Field Documentation . . . 575
8.541.1.1 notification . . . 575
8.541.1.2 sessionStatus . . . 575
8.542 operatorNameString Struct Reference . . . 575
8.542.1 Detailed Description . . . 576

8.542.2 Field Documentation . . . 576
8.542.2.1 PLMNName . . . 576
8.543 OperatorPLMNData Struct Reference . . . 576
8.543.1 Detailed Description . . . 576
8.543.2 Field Documentation . . . 576
8.543.2.1 lac1 . . . 576
8.543.2.2 lac2 . . . 577
8.543.2.3 mcc . . . 577
8.543.2.4 mnc . . . 577
8.543.2.5 PLMNRecID . . . 577
8.544 operatorPLMNList Struct Reference . . . 577
8.544.1 Detailed Description . . . 577
8.544.2 Field Documentation . . . 577
8.544.2.1 numInstance . . . 577
8.544.2.2 PLMNData . . . 577
8.545 pack_dms_ActivateAutomatic_t Struct Reference . . . 577
8.545.1 Detailed Description . . . 577
8.545.2 Field Documentation . . . 577
8.545.2.1 actCode . . . 578
8.546 pack_dms_GetCustFeaturesV2_t Struct Reference . . . 578
8.546.1 Detailed Description . . . 578
8.546.2 Field Documentation . . . 578
8.546.2.1 cust_id . . . 578
8.546.2.2 list_type . . . 578
8.546.2.3 Tlvresult . . . 578
8.547 pack_dms_ResetToFactoryDefaults_t Struct Reference . . . 578
8.547.1 Detailed Description . . . 578
8.547.2 Field Documentation . . . 578
8.547.2.1 spc . . . 579
8.548 pack_dms_SetActivationStatusCallback_t Struct Reference . . . 579
8.548.1 Detailed Description . . . 579
8.548.2 Field Documentation . . . 579
8.548.2.1 activationState . . . 579
8.549 pack_dms_SetCrashAction_t Struct Reference . . . 579
8.549.1 Detailed Description . . . 579
8.549.2 Field Documentation . . . 579
8.549.2.1 crashAction . . . 579
8.550 pack_dms_SetCustFeature_t Struct Reference . . . 579
8.550.1 Field Documentation . . . 580
8.550.1.1 DHCPRelayEnabled . . . 580

8.550.1.2 DisableIMSI . . . 580
8.550.1.3 GpsEnable . . . 580
8.550.1.4 GPSLPM . . . 580
8.550.1.5 GPSSel . . . 580
8.550.1.6 IPFamSupport . . . 580
8.550.1.7 IsVoiceEnabled . . . 580
8.550.1.8 RMAutoConnect . . . 580
8.550.1.9 SMSSupport . . . 580
8.551 pack_dms_SetCustFeaturesV2_t Struct Reference . . . 580
8.551.1 Detailed Description . . . 580
8.551.2 Field Documentation . . . 581
8.551.2.1 cust_id . . . 581
8.551.2.2 cust_value . . . 581
8.551.2.3 Tlvresult . . . 581
8.551.2.4 value_length . . . 581
8.552 pack_dms_SetEventReport_t Struct Reference . . . 581
8.552.1 Field Documentation . . . 581
8.552.1.1 mode . . . 581
8.553 pack_dms_SetPower_t Struct Reference . . . 581
8.553.1 Field Documentation . . . 581
8.553.1.1 mode . . . 581
8.553.1.2 Tlvresult . . . 581
8.554 pack_dms_SetUSBComp_t Struct Reference . . . 581
8.554.1 Field Documentation . . . 581
8.554.1.1 Tlvresult . . . 582
8.554.1.2 USBComp . . . 582
8.555 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 582
8.555.1 Detailed Description . . . 582
8.555.2 Field Documentation . . . 582
8.555.2.1 resetInfoInd . . . 582
8.556 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 582
8.556.1 Detailed Description . . . 582
8.556.2 Field Documentation . . . 582
8.556.2.1 clear . . . 582
8.557 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 582
8.557.1 Detailed Description . . . 583
8.557.2 Field Documentation . . . 583
8.557.2.1 pDestSMSContent . . . 583
8.557.2.2 pDestSMSNum . . . 583
8.558 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 583

8.558.1 Detailed Description . . . 583
8.558.2 Field Documentation . . . 583
8.558.2.1 manString . . . 583
8.558.2.2 modelString . . . 583
8.558.2.3 plasmaIDString . . . 584
8.558.2.4 swVerString . . . 584
8.559 pack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 584
8.559.1 Detailed Description . . . 584
8.559.2 Field Documentation . . . 584
8.559.2.1 nameString . . . 584
8.559.2.2 versionString . . . 584
8.560 pack_dms_UIMChangePIN_t Struct Reference . . . 584
8.560.1 Detailed Description . . . 584
8.560.2 Field Documentation . . . 585
8.560.2.1 id . . . 585
8.560.2.2 newValue . . . 585
8.560.2.3 oldValue . . . 585
8.561 pack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 585
8.561.1 Detailed Description . . . 585
8.561.2 Field Documentation . . . 585
8.561.2.1 facility . . . 585
8.562 pack_dms_UIMGetICCID_t Struct Reference . . . 585
8.562.1 Detailed Description . . . 585
8.562.2 Field Documentation . . . 586
8.562.2.1 Tlvresult . . . 586
8.563 pack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 586
8.563.1 Detailed Description . . . 586
8.563.2 Field Documentation . . . 586
8.563.2.1 facility . . . 586
8.563.2.2 facilityCk . . . 586
8.563.2.3 facilityState . . . 586
8.564 pack_dms_UIMSetPINProtection_t Struct Reference . . . 586
8.564.1 Detailed Description . . . 586
8.564.2 Field Documentation . . . 587
8.564.2.1 bEnable . . . 587
8.564.2.2 id . . . 587
8.564.2.3 value . . . 587
8.565 pack_dms_UIMUnblockControlKey_t Struct Reference . . . 587
8.565.1 Detailed Description . . . 587
8.565.2 Field Documentation . . . 587

8.565.2.1 facility . . . 588
8.565.2.2 facilityCk . . . 588
8.566 pack_dms_UIMUnblockPIN_t Struct Reference . . . 588
8.566.1 Detailed Description . . . 588
8.566.2 Field Documentation . . . 588
8.566.2.1 id . . . 588
8.566.2.2 newPin . . . 588
8.566.2.3 pukValue . . . 588
8.567 pack_dms_UIMVerifyPIN_t Struct Reference . . . 588
8.567.1 Detailed Description . . . 588
8.567.2 Field Documentation . . . 589
8.567.2.1 id . . . 589
8.567.2.2 value . . . 589
8.568 pack_fms_GetImagesPreference_t Struct Reference . . . 589
8.568.1 Detailed Description . . . 589
8.568.2 Field Documentation . . . 589
8.568.2.1 Tlvresult . . . 589
8.569 pack_fms_GetStoredImages_t Struct Reference . . . 589
8.569.1 Detailed Description . . . 589
8.569.2 Field Documentation . . . 590
8.569.2.1 Tlvresult . . . 590
8.570 pack_fms_SetImagesPreference_t Struct Reference . . . 590
8.570.1 Detailed Description . . . 590
8.570.2 Field Documentation . . . 590
8.570.2.1 bForceDownload . . . 590
8.570.2.2 imageListSize . . . 590
8.570.2.3 modemindex . . . 590
8.570.2.4 pImageList . . . 590
8.570.2.5 Tlvresult . . . 590
8.571 pack_loc_Delete_Assist_Data_t Struct Reference . . . 590
8.571.1 Detailed Description . . . 591
8.571.2 Field Documentation . . . 591
8.571.2.1 pBdsSVInfo . . . 591
8.571.2.2 pCellDb . . . 591
8.571.2.3 pClkInfo . . . 591
8.571.2.4 pGnssData . . . 591
8.571.2.5 pSVInfo . . . 591
8.571.2.6 Tlvresult . . . 591
8.572 pack_loc_EventRegister_t Struct Reference . . . 591
8.572.1 Detailed Description . . . 592

8.572.2 Field Documentation . . . 594
8.572.2.1 eventRegister . . . 594
8.572.2.2 TlvResult . . . 594
8.573 pack_loc_SetExtPowerState_t Struct Reference . . . 594
8.573.1 Detailed Description . . . 594
8.573.2 Field Documentation . . . 594
8.573.2.1 extPowerState . . . 594
8.573.2.2 TlvResult . . . 594
8.574 pack_loc_SetOperationMode_t Struct Reference . . . 594
8.574.1 Detailed Description . . . 595
8.574.2 Field Documentation . . . 595
8.574.2.1 mode . . . 595
8.574.2.2 TlvResult . . . 595
8.575 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 595
8.575.1 Detailed Description . . . 595
8.575.2 Field Documentation . . . 595
8.575.2.1 TlvResult . . . 595
8.575.2.2 xid . . . 595
8.576 pack_loc_SLQSLOCInjectPosition_t Struct Reference . . . 595
8.576.1 Detailed Description . . . 596
8.576.2 Field Documentation . . . 599
8.576.2.1 altitudeSrcInfo . . . 599
8.576.2.2 altitudeWrtEllipsoid . . . 599
8.576.2.3 altitudeWrtMeanSeaLevel . . . 599
8.576.2.4 has_altitudeSrcInfo . . . 599
8.576.2.5 has_altitudeWrtEllipsoid . . . 599
8.576.2.6 has_altitudeWrtMeanSeaLevel . . . 599
8.576.2.7 has_horConfidence . . . 599
8.576.2.8 has_horReliability . . . 599
8.576.2.9 has_horUncCircular . . . 600
8.576.2.10 has_latitude . . . 600
8.576.2.11 has_longitude . . . 600
8.576.2.12 has_positionSrc . . . 600
8.576.2.13 has_rawHorConfidence . . . 600
8.576.2.14 has_rawHorUncCircular . . . 600
8.576.2.15 has_timestampAge . . . 600
8.576.2.16 has_timestampUtc . . . 600
8.576.2.17 has_vertConfidence . . . 600
8.576.2.18 has_vertRelicability . . . 600
8.576.2.19 has_vertUnc . . . 600

8.576.2.20 horConfidence . . . 600
8.576.2.21 horReliability . . . 600
8.576.2.22 horUncCircular . . . 600
8.576.2.23 latitude . . . 600
8.576.2.24 longitude . . . 600
8.576.2.25 positionSrc . . . 600
8.576.2.26 rawHorConfidence . . . 600
8.576.2.27 rawHorUncCircular . . . 600
8.576.2.28 timestampAge . . . 600
8.576.2.29 timestampUtc . . . 600
8.576.2.30 vertConfidence . . . 600
8.576.2.31 vertReliability . . . 600
8.576.2.32 vertUnc . . . 600
8.577 pack_loc_SLQSLOCInjectSensorData_t Struct Reference . . . 600
8.577.1 Detailed Description . . . 601
8.577.2 Field Documentation . . . 602
8.577.2.1 acceleroData . . . 602
8.577.2.2 acceleroTimeSrc . . . 602
8.577.2.3 accelTemp . . . 602
8.577.2.4 gyroData . . . 602
8.577.2.5 gyroTemp . . . 602
8.577.2.6 gyroTimeSrc . . . 602
8.577.2.7 has_acceleroTimeSrc . . . 602
8.577.2.8 has_accelTemp . . . 602
8.577.2.9 has_accleroData . . . 602
8.577.2.10 has_gyroData . . . 602
8.577.2.11 has_gyroTemp . . . 602
8.577.2.12 has_gyroTimeSrc . . . 602
8.577.2.13 has_opaqueId . . . 602
8.577.2.14 opaqueId . . . 602
8.578 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference . . . 602
8.578.1 Detailed Description . . . 602
8.578.2 Field Documentation . . . 603
8.578.2.1 timeMsec . . . 603
8.578.2.2 timeUncMsec . . . 603
8.579 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference . . . 603
8.579.1 Detailed Description . . . 603
8.579.2 Field Documentation . . . 603
8.579.2.1 confidence . . . 603
8.579.2.2 has_confidence . . . 603

8.579.2.3 state . . . 603
8.580 pack_loc_Start_t Struct Reference . . . 603
8.580.1 Detailed Description . . . 604
8.580.2 Field Documentation . . . 604
8.580.2.1 pApplicationInfo . . . 605
8.580.2.2 pConfigAltitudeAssumed . . . 605
8.580.2.3 pHorizontalAccuracyLvl . . . 605
8.580.2.4 pIntermediateReportState . . . 605
8.580.2.5 pMinIntervalTime . . . 605
8.580.2.6 pRecurrenceType . . . 605
8.580.2.7 SessionId . . . 605
8.580.2.8 Tlvresult . . . 605
8.581 pack_loc_Stop_t Struct Reference . . . 605
8.581.1 Detailed Description . . . 605
8.581.2 Field Documentation . . . 605
8.581.2.1 SessionId . . . 605
8.581.2.2 Tlvresult . . . 605
8.582 pack_nas_InitiateDomainAttach_t Struct Reference . . . 605
8.582.1 Detailed Description . . . 605
8.582.2 Field Documentation . . . 606
8.582.2.1 action . . . 606
8.583 pack_nas_SetACCOLC_t Struct Reference . . . 606
8.583.1 Detailed Description . . . 606
8.583.2 Field Documentation . . . 606
8.583.2.1 accolc . . . 606
8.583.2.2 spc . . . 606
8.584 pack_nas_SetCDMANetworkParameters_t Struct Reference . . . 606
8.584.1 Detailed Description . . . 606
8.584.2 Field Documentation . . . 607
8.584.2.1 pApplication . . . 607
8.584.2.2 pBroadcast . . . 607
8.584.2.3 pCustomSCP . . . 607
8.584.2.4 pForceRev0 . . . 608
8.584.2.5 pProtocol . . . 608
8.584.2.6 pRoaming . . . 608
8.584.2.7 pSPC . . . 608
8.585 pack_nas_SetNetworkPreference_t Struct Reference . . . 608
8.585.1 Detailed Description . . . 608
8.585.2 Field Documentation . . . 608
8.585.2.1 Duration . . . 608

8.585.2.2 TechnologyPref . . . 608
8.585.2.3 Tlvresult . . . 608
8.586pack_nas_SLQSConfigSigInfo_t Struct Reference . . . 609
8.586.1 Detailed Description . . . 609
8.586.2 Field Documentation . . . 610
8.586.2.1 pECIOThresh . . . 610
8.586.2.2 pHDRSINRThresh . . . 610
8.586.2.3 pIOThresh . . . 610
8.586.2.4 pLTESigRptCfg . . . 610
8.586.2.5 pLTESNRThresh . . . 610
8.586.2.6 pRSRPThresh . . . 610
8.586.2.7 pRSRQThresh . . . 610
8.586.2.8 pRSSIThresh . . . 610
8.586.2.9 pTDSCDMASINRCONFThresh . . . 610
8.587pack_nas_SLQSGetPLMNName_t Struct Reference . . . 610
8.587.1 Detailed Description . . . 610
8.587.2 Field Documentation . . . 611
8.587.2.1 mcc . . . 611
8.587.2.2 mnc . . . 611
8.587.2.3 pMncPcsStatus . . . 611
8.588pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference . . . 611
8.588.1 Detailed Description . . . 611
8.588.2 Field Documentation . . . 611
8.588.2.1 pChangeDuration . . . 612
8.588.2.2 pMncPcsDigitStatus . . . 612
8.588.2.3 pMNRInfo . . . 612
8.588.2.4 regAction . . . 612
8.589pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . 612
8.589.1 Detailed Description . . . 613
8.589.2 Field Documentation . . . 615
8.589.2.1 pCDMAECIODelta . . . 615
8.589.2.2 pCDMAECIOThresh . . . 615
8.589.2.3 pCDMARSSIDelta . . . 615
8.589.2.4 pCDMARSSIThresh . . . 615
8.589.2.5 pGSMRSSIDelta . . . 615
8.589.2.6 pGSMRSSIThresh . . . 615
8.589.2.7 pHDRECIODelta . . . 615
8.589.2.8 pHDRECIOThresh . . . 615
8.589.2.9 pHDRIODelta . . . 615
8.589.2.10pHDRIOThresh . . . 615

8.589.2.11 pHDRRSSIDelta . . . 615
8.589.2.12 pHDRRSSIThresh . . . 615
8.589.2.13 pHDRSINRDelta . . . 615
8.589.2.14 pHDRSINRThresh . . . 615
8.589.2.15 pLTERSRPDelta . . . 615
8.589.2.16 pLTERSRPThresh . . . 615
8.589.2.17 pLTERSRQDelta . . . 616
8.589.2.18 pLTERSRQThresh . . . 616
8.589.2.19 pLTERSSIDelta . . . 616
8.589.2.20 pLTERSSIThresh . . . 616
8.589.2.21 pLTESigRptConfig . . . 616
8.589.2.22 pLTESNRDelta . . . 616
8.589.2.23 pLTESNRThresh . . . 616
8.589.2.24 pTDSCDMAECIODelta . . . 616
8.589.2.25 pTDSCDMAECIOThresh . . . 616
8.589.2.26 pTDSCDMARSCPDelta . . . 616
8.589.2.27 pTDSCDMARSCPThresh . . . 616
8.589.2.28 pTDSCDMARSSIDelta . . . 616
8.589.2.29 pTDSCDMARSSIThresh . . . 616
8.589.2.30 pTDSCDMASINRDelta . . . 616
8.589.2.31 pTDSCDMASINRThresh . . . 616
8.589.2.32 pWCDMAECIODelta . . . 616
8.589.2.33 pWCDMAECIOThresh . . . 616
8.589.2.34 pWCDMARSSIDelta . . . 616
8.589.2.35 pWCDMARSSIThresh . . . 616
8.590 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 616
8.590.1 Detailed Description . . . 616
8.590.2 Field Documentation . . . 617
8.590.2.1 namID . . . 617
8.591 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 617
8.591.1 Detailed Description . . . 617
8.591.2 Field Documentation . . . 617
8.591.2.1 radio_if . . . 617
8.592 pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference . . . 617
8.592.1 Detailed Description . . . 618
8.592.2 Field Documentation . . . 620
8.592.2.1 pDDTMInd . . . 620
8.592.2.2 pDualStandByPrefInd . . . 620
8.592.2.3 pErrorRateInd . . . 620
8.592.2.4 pHDRNewUATIAssInd . . . 620

8.592.2.5 pHDRSessionCloseInd . . . 620
8.592.2.6 pLTECphyCa . . . 620
8.592.2.7 pManagedRoamingInd . . . 620
8.592.2.8 pNetworkRejectInd . . . 620
8.592.2.9 pNetworkTimeInd . . . 620
8.592.2.10 pServingSystemInd . . . 620
8.592.2.11 pSignalStrengthInd . . . 620
8.592.2.12 pSubscriptionInfoInd . . . 620
8.592.2.13 pSuppressSysInfoInd . . . 620
8.592.2.14 pSysInfoInd . . . 620
8.592.2.15 pSystemSelectionInd . . . 620
8.593 pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference . . . 620
8.593.1 Detailed Description . . . 620
8.593.2 Field Documentation . . . 621
8.593.2.1 gsmUmtsDl . . . 621
8.593.2.2 gsmUmtsUl . . . 621
8.593.2.3 lteEmmDl . . . 621
8.593.2.4 lteEmmUl . . . 621
8.593.2.5 lteEsmDl . . . 621
8.593.2.6 lteEsmUl . . . 621
8.593.2.7 pRankIndicatorInd . . . 621
8.593.2.8 pTimer . . . 621
8.594 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference . . . 621
8.594.1 Detailed Description . . . 622
8.594.2 Field Documentation . . . 622
8.594.2.1 pLteEARFCN . . . 622
8.594.2.2 pLtePCI . . . 622
8.594.2.3 pWcdmaUARFCN . . . 622
8.595 pack_nas_SLQSSetBandPreference_t Struct Reference . . . 622
8.595.1 Detailed Description . . . 622
8.595.2 Field Documentation . . . 623
8.595.2.1 bandPref . . . 624
8.596 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference . . . 624
8.596.1 Detailed Description . . . 624
8.596.2 Field Documentation . . . 624
8.596.2.1 bEnable . . . 624
8.596.2.2 pSigIndReq . . . 624
8.597 pack_nas_SLQSSetSysSelectionPref_t Struct Reference . . . 624
8.597.1 Detailed Description . . . 624
8.597.2 Field Documentation . . . 628

8.597.2.1 pAcqOrderPref . . . 628
8.597.2.2 pBandPref . . . 628
8.597.2.3 pChgDuration . . . 628
8.597.2.4 pCSGID . . . 628
8.597.2.5 pEmerMode . . . 628
8.597.2.6 pGWAcqOrderPref . . . 628
8.597.2.7 pLTEBandPref . . . 629
8.597.2.8 pMNCInCPCSDigStat . . . 629
8.597.2.9 pModePref . . . 629
8.597.2.10 pNetSelPref . . . 629
8.597.2.11 pPRLPref . . . 629
8.597.2.12 pRAT . . . 629
8.597.2.13 pRoamPref . . . 629
8.597.2.14 pSrvDomainPref . . . 629
8.597.2.15 pSrvRegRestriction . . . 629
8.597.2.16 pTdscdmaBandPref . . . 629
8.598 pack_nas_SLQSSwiPSDetach_t Struct Reference . . . 629
8.598.1 Detailed Description . . . 629
8.598.2 Field Documentation . . . 629
8.598.2.1 pDetachAction . . . 629
8.599 pack_pds_SetPDSDefaults_t Struct Reference . . . 629
8.599.1 Detailed Description . . . 629
8.599.2 Field Documentation . . . 630
8.599.2.1 accuracy . . . 630
8.599.2.2 interval . . . 630
8.599.2.3 operation . . . 630
8.599.2.4 timeout . . . 630
8.600 pack_pds_SetPDSState_t Struct Reference . . . 630
8.600.1 Detailed Description . . . 630
8.600.2 Field Documentation . . . 630
8.600.2.1 enable . . . 630
8.601 pack_pds_SetPortAutomaticTracking_t Struct Reference . . . 630
8.601.1 Detailed Description . . . 631
8.601.2 Field Documentation . . . 631
8.601.2.1 bAuto . . . 631
8.602 pack_qmi_t Struct Reference . . . 631
8.602.1 Detailed Description . . . 631
8.602.2 Field Documentation . . . 631
8.602.2.1 msgid . . . 631
8.602.2.2 svc . . . 631

8.602.2.3 timeout . . . 631
8.602.2.4 xid . . . 631
8.603 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 631
8.603.1 Detailed Description . . . 632
8.603.2 Field Documentation . . . 632
8.603.2.1 apnId . . . 632
8.604 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 632
8.604.1 Detailed Description . . . 632
8.604.2 Field Documentation . . . 632
8.604.2.1 apnId . . . 632
8.605 pack_qos_SLQSSetQosEventCallback_t Struct Reference . . . 632
8.605.1 Detailed Description . . . 632
8.605.2 Field Documentation . . . 633
8.605.2.1 enable . . . 633
8.606 pack_rms_SetSMSWake_t Struct Reference . . . 633
8.606.1 Detailed Description . . . 633
8.606.2 Field Documentation . . . 633
8.606.2.1 enabled . . . 633
8.606.2.2 wake_mask . . . 633
8.607 pack_sar_SLQSSetRfSarState_t Struct Reference . . . 633
8.607.1 Detailed Description . . . 633
8.607.2 Field Documentation . . . 634
8.607.2.1 RfSarState . . . 634
8.608 pack_sms_SaveSMS_t Struct Reference . . . 634
8.608.1 Detailed Description . . . 634
8.608.2 Field Documentation . . . 635
8.608.2.1 messageFormat . . . 635
8.608.2.2 messageSize . . . 635
8.608.2.3 pMessage . . . 635
8.608.2.4 storageType . . . 635
8.609 pack_sms_SendSMS_t Struct Reference . . . 635
8.609.1 Detailed Description . . . 635
8.609.2 Field Documentation . . . 635
8.609.2.1 messageFormat . . . 635
8.609.2.2 messageSize . . . 635
8.609.2.3 pLinktimer . . . 635
8.609.2.4 pMessage . . . 635
8.610 pack_sms_SetNewSMSCallback_t Struct Reference . . . 635
8.610.1 Detailed Description . . . 636
8.610.2 Field Documentation . . . 636

8.610.2.1 status . . . 636
8.611 pack_sms_SetSMSCAddress_t Struct Reference . . . 636
8.611.1 Detailed Description . . . 636
8.611.2 Field Documentation . . . 636
8.611.2.1 pSMSCAddress . . . 636
8.611.2.2 pSMSCType . . . 636
8.612 pack_sms_SLQSDeleteSMS_t Struct Reference . . . 636
8.612.1 Detailed Description . . . 636
8.612.2 Field Documentation . . . 637
8.612.2.1 pMessageIndex . . . 637
8.612.2.2 pMessageMode . . . 637
8.612.2.3 pMessageTag . . . 637
8.612.2.4 storageType . . . 637
8.613 pack_sms_SLQSGetSMS_t Struct Reference . . . 637
8.613.1 Detailed Description . . . 637
8.613.2 Field Documentation . . . 638
8.613.2.1 messageIndex . . . 638
8.613.2.2 pMessageMode . . . 638
8.613.2.3 storageType . . . 638
8.614 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 638
8.614.1 Detailed Description . . . 638
8.614.2 Field Documentation . . . 638
8.614.2.1 mode . . . 638
8.615 pack_sms_SLQSGetSMSList_t Struct Reference . . . 638
8.615.1 Detailed Description . . . 638
8.615.2 Field Documentation . . . 639
8.615.2.1 pMessageMode . . . 639
8.615.2.2 pRequestedTag . . . 639
8.615.2.3 storageType . . . 639
8.616 pack_sms_SLQSModifySMSStatus_t Struct Reference . . . 639
8.616.1 Detailed Description . . . 639
8.616.2 Field Documentation . . . 640
8.616.2.1 messageIndex . . . 640
8.616.2.2 messageTag . . . 640
8.616.2.3 pMessageMode . . . 640
8.616.2.4 storageType . . . 640
8.617 pack_sms_SLQSSendAsyncSMS_t Struct Reference . . . 640
8.617.1 Detailed Description . . . 640
8.617.2 Field Documentation . . . 640
8.617.2.1 pSendSmsParams . . . 640

8.618 pack_sms_SLQSSetIndicationRegister_t Struct Reference . . . 640
8.618.1 Detailed Description . . . 640
8.618.2 Field Documentation . . . 641
8.618.2.1 pSetIndicationRegReq . . . 641
8.619 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . 641
8.619.1 Detailed Description . . . 641
8.619.2 Field Documentation . . . 641
8.619.2.1 broadcastActivate . . . 641
8.619.2.2 mode . . . 641
8.620 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference . . . 641
8.620.1 Detailed Description . . . 641
8.620.2 Field Documentation . . . 642
8.620.2.1 mode . . . 642
8.620.2.2 pBroadcastConfig . . . 642
8.620.2.3 pCDMABroadcastConfig . . . 642
8.621 pack_sms_SLQSSetSmsStorage_t Struct Reference . . . 642
8.621.1 Detailed Description . . . 642
8.621.2 Field Documentation . . . 642
8.621.2.1 smsStorage . . . 642
8.622 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 642
8.622.1 Detailed Description . . . 642
8.622.2 Field Documentation . . . 643
8.622.2.1 pMaxStorageSizeReq . . . 643
8.623 pack_sms_SLQSSmsSetRoutes_t Struct Reference . . . 643
8.623.1 Detailed Description . . . 643
8.623.2 Field Documentation . . . 643
8.623.2.1 pSetRoutesReq . . . 643
8.624 pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . 643
8.624.1 Detailed Description . . . 643
8.624.2 Field Documentation . . . 644
8.624.2.1 fix_rate . . . 644
8.624.2.2 fix_type . . . 644
8.624.2.3 function . . . 644
8.624.2.4 max_dist . . . 644
8.624.2.5 max_time . . . 644
8.624.2.6 set_fix_rate . . . 645
8.624.2.7 set_fix_type . . . 645
8.624.2.8 set_function . . . 645
8.624.2.9 set_max_dist . . . 645
8.624.2.10 set_max_time . . . 645

8.625 pack_swioma_SLQSOMADMCancelSession_t Struct Reference . . . 645
8.625.1 Detailed Description . . . 645
8.625.2 Field Documentation . . . 645
8.625.2.1 sessionType . . . 645
8.626 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 645
8.626.1 Detailed Description . . . 645
8.626.2 Field Documentation . . . 646
8.626.2.1 SessionType . . . 646
8.627 pack_swioma_SLQSOMADMSendSelection_t Struct Reference . . . 646
8.627.1 Detailed Description . . . 646
8.627.2 Field Documentation . . . 646
8.627.2.1 pDeferTime . . . 646
8.627.2.2 pRejectReason . . . 646
8.627.2.3 selection . . . 646
8.628 pack_swioma_SLQSOMADMSetSettings_t Struct Reference . . . 646
8.628.1 Detailed Description . . . 647
8.628.2 Field Documentation . . . 647
8.628.2.1 FOTAdownload . . . 647
8.628.2.2 FOTAUpdate . . . 647
8.628.2.3 pAutosdm . . . 647
8.628.2.4 pFwAutoCheck . . . 647
8.629 pack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 647
8.629.1 Detailed Description . . . 647
8.629.2 Field Documentation . . . 648
8.629.2.1 sessionType . . . 648
8.630 pack_uim_ChangePin_t Struct Reference . . . 648
8.630.1 Detailed Description . . . 648
8.630.2 Field Documentation . . . 649
8.630.2.1 changePIN . . . 649
8.630.2.2 EncryptedPIN1 . . . 649
8.630.2.3 pIndicationToken . . . 649
8.630.2.4 pKeyReferenceID . . . 649
8.630.2.5 sessionInfo . . . 649
8.630.2.6 Tlvresult . . . 649
8.631 pack_uim_ReadTransparent_t Struct Reference . . . 649
8.631.1 Detailed Description . . . 649
8.631.2 Field Documentation . . . 650
8.631.2.1 fileIndex . . . 650
8.631.2.2 pEncryptData . . . 650
8.631.2.3 pIndicationToken . . . 650

8.631.2.4 readTransparent . . . 650
8.631.2.5 sessionInfo . . . 650
8.631.2.6 TlvResult . . . 650
8.632 pack_uim_SetPinProtection_t Struct Reference . . . 650
8.632.1 Detailed Description . . . 650
8.632.2 Field Documentation . . . 651
8.632.2.1 EncryptedPIN1 . . . 651
8.632.2.2 pIndicationToken . . . 651
8.632.2.3 pinProtection . . . 651
8.632.2.4 pKeyReferenceID . . . 651
8.632.2.5 sessionInfo . . . 651
8.632.2.6 TlvResult . . . 651
8.633 pack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 651
8.633.1 Detailed Description . . . 651
8.633.2 Field Documentation . . . 652
8.633.2.1 authData . . . 652
8.633.2.2 pIndicationToken . . . 652
8.633.2.3 sessionInfo . . . 652
8.634 pack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 652
8.634.1 Detailed Description . . . 652
8.634.2 Field Documentation . . . 652
8.634.2.1 depersonilisationInfo . . . 652
8.635 pack_uim_SLQSUIMEventRegister_t Struct Reference . . . 652
8.635.1 Detailed Description . . . 652
8.635.2 Field Documentation . . . 652
8.635.2.1 eventMask . . . 652
8.636 pack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 652
8.636.1 Detailed Description . . . 653
8.636.2 Field Documentation . . . 653
8.636.2.1 pConfigurationMask . . . 653
8.637 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 653
8.637.1 Detailed Description . . . 653
8.637.2 Field Documentation . . . 654
8.637.2.1 fileIndex . . . 654
8.637.2.2 pIndicationToken . . . 654
8.637.2.3 sessionInfo . . . 654
8.638 pack_uim_SLQSUIMPowerDown_t Struct Reference . . . 654
8.638.1 Detailed Description . . . 654
8.638.2 Field Documentation . . . 654
8.638.2.1 slot . . . 654

8.639 pack_uim_SLQSUIMPowerUp_t Struct Reference . . . 654
8.639.1 Detailed Description . . . 654
8.639.2 Field Documentation . . . 655
8.639.2.1 pIgnoreHotSwapSwitch . . . 655
8.639.2.2 slot . . . 655
8.640 pack_uim_SLQSUIMRefreshComplete_t Struct Reference . . . 655
8.640.1 Detailed Description . . . 655
8.640.2 Field Documentation . . . 655
8.640.2.1 refreshComplete . . . 655
8.640.2.2 sessionInfo . . . 655
8.641 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 655
8.641.1 Detailed Description . . . 655
8.641.2 Field Documentation . . . 656
8.641.2.1 sessionInfo . . . 656
8.642 pack_uim_SLQSUIMRefreshOK_t Struct Reference . . . 656
8.642.1 Detailed Description . . . 656
8.642.2 Field Documentation . . . 656
8.642.2.1 OKtoRefresh . . . 656
8.642.2.2 sessionInfo . . . 656
8.643 pack_uim_SLQSUIMRefreshRegister_t Struct Reference . . . 656
8.643.1 Detailed Description . . . 656
8.643.2 Field Documentation . . . 657
8.643.2.1 regRefresh . . . 657
8.643.2.2 sessionInfo . . . 657
8.644 pack_uim_SLQSUIMSwitchSlot_t Struct Reference . . . 657
8.644.1 Detailed Description . . . 657
8.644.2 Field Documentation . . . 657
8.644.2.1 bLogicalSlot . . . 657
8.644.2.2 ulPhysicalSlot . . . 657
8.645 pack_uim_UnblockPin_t Struct Reference . . . 658
8.645.1 Detailed Description . . . 658
8.645.2 Field Documentation . . . 658
8.645.2.1 EncryptedPIN1 . . . 658
8.645.2.2 pIndicationToken . . . 658
8.645.2.3 pinProtection . . . 658
8.645.2.4 pKeyReferenceID . . . 658
8.645.2.5 sessionInfo . . . 658
8.645.2.6 Tlvresult . . . 659
8.646 pack_uim_VerifyPin_t Struct Reference . . . 659
8.646.1 Detailed Description . . . 659

8.646.2 Field Documentation . . . 659
8.646.2.1 pEncryptedPIN1 . . . 659
8.646.2.2 pIndicationToken . . . 659
8.646.2.3 pKeyReferenceID . . . 660
8.646.2.4 sessionInfo . . . 660
8.646.2.5 Tlvresult . . . 660
8.646.2.6 verifyPIN . . . 660
8.647 pack_voice_AnswerUSSD_t Struct Reference . . . 660
8.647.1 Detailed Description . . . 660
8.647.2 Field Documentation . . . 660
8.647.2.1 pInfo . . . 660
8.648 pack_voice_OriginateUSSD_t Struct Reference . . . 660
8.648.1 Detailed Description . . . 660
8.648.2 Field Documentation . . . 660
8.648.2.1 pInfo . . . 660
8.649 pack_voice_SLQSOriginateUSSD_t Struct Reference . . . 660
8.649.1 Detailed Description . . . 661
8.649.2 Field Documentation . . . 661
8.649.2.1 ussData . . . 661
8.649.2.2 ussDCS . . . 661
8.649.2.3 ussLen . . . 661
8.650 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference . . . 661
8.650.1 Detailed Description . . . 661
8.650.2 Field Documentation . . . 661
8.650.2.1 lineValue . . . 661
8.651 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference . . . 661
8.651.1 Detailed Description . . . 662
8.651.2 Field Documentation . . . 662
8.651.2.1 switchOption . . . 662
8.652 pack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 662
8.652.1 Detailed Description . . . 662
8.652.2 Field Documentation . . . 662
8.652.2.1 pCallId . . . 662
8.653 pack_voice_SLQSVoiceBindSubscription_t Struct Reference . . . 662
8.653.1 Detailed Description . . . 662
8.653.2 Field Documentation . . . 663
8.653.2.1 subsType . . . 663
8.654 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 663
8.654.1 Detailed Description . . . 663
8.654.2 Field Documentation . . . 663

8.654.2.1 BurstDTMFInfo . . . 663
8.654.2.2 pBurstDTMFLengths . . . 663
8.655 pack_voice_SLQSVoiceDialCall_t Struct Reference . . . 663
8.655.1 Detailed Description . . . 664
8.655.2 Field Documentation . . . 665
8.655.2.1 callNumber . . . 665
8.655.2.2 pCallPartySubAdd . . . 665
8.655.2.3 pCallType . . . 665
8.655.2.4 pCLIRType . . . 665
8.655.2.5 pCUGInfo . . . 665
8.655.2.6 pEmergencyCategory . . . 665
8.655.2.7 pSvcType . . . 665
8.655.2.8 pUUSInFo . . . 665
8.656 pack_voice_SLQSVoiceEndCall_t Struct Reference . . . 665
8.656.1 Detailed Description . . . 665
8.656.2 Field Documentation . . . 665
8.656.2.1 pCallId . . . 665
8.657 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 665
8.657.1 Detailed Description . . . 666
8.657.2 Field Documentation . . . 666
8.657.2.1 pSvcClass . . . 666
8.657.2.2 reason . . . 666
8.658 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 666
8.658.1 Detailed Description . . . 666
8.658.2 Field Documentation . . . 667
8.658.2.1 pSvcClass . . . 667
8.658.2.2 Reason . . . 667
8.659 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 667
8.659.1 Detailed Description . . . 667
8.659.2 Field Documentation . . . 667
8.659.2.1 callID . . . 667
8.660 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 667
8.660.1 Detailed Description . . . 668
8.660.2 Field Documentation . . . 668
8.660.2.1 pSvcClass . . . 668
8.661 pack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 668
8.661.1 Detailed Description . . . 668
8.661.2 Field Documentation . . . 669
8.661.2.1 pAirTimer . . . 669
8.661.2.2 pAMRStatus . . . 669

8.661.2.3 pAutoAnswer . . . 669
8.661.2.4 pNamID . . . 669
8.661.2.5 pPrefVoicePrivacy . . . 669
8.661.2.6 pPrefVoiceSO . . . 669
8.661.2.7 pRoamTimer . . . 669
8.661.2.8 pTTYMode . . . 669
8.661.2.9 pVoiceDomainPref . . . 669
8.662 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference . . . 670
8.662.1 Detailed Description . . . 670
8.662.2 Field Documentation . . . 670
8.662.2.1 pRegDTMFEvents . . . 670
8.662.2.2 pRegVoicePrivacyEvents . . . 670
8.662.2.3 pSuppsNotifEvents . . . 670
8.663 pack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 670
8.663.1 Detailed Description . . . 671
8.663.2 Field Documentation . . . 671
8.663.2.1 pCallID . . . 671
8.663.2.2 SUPSType . . . 671
8.664 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference . . . 671
8.664.1 Detailed Description . . . 671
8.664.2 Field Documentation . . . 672
8.664.2.1 USSInformation . . . 672
8.665 pack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 672
8.665.1 Detailed Description . . . 672
8.665.2 Field Documentation . . . 672
8.665.2.1 pCallID . . . 672
8.665.2.2 pFlashPayLd . . . 672
8.665.2.3 pFlashType . . . 672
8.666 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 672
8.666.1 Detailed Description . . . 673
8.666.2 Field Documentation . . . 673
8.666.2.1 newPasswd . . . 673
8.666.2.2 newPasswdAgain . . . 673
8.666.2.3 oldPasswd . . . 673
8.666.2.4 Reason . . . 673
8.667 pack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 673
8.667.1 Detailed Description . . . 674
8.667.2 Field Documentation . . . 675
8.667.2.1 pAirTimerConfig . . . 675
8.667.2.2 pAutoAnswer . . . 675

8.667.2.3 pPrefVoiceDomain . . . 675
8.667.2.4 pPrefVoiceSO . . . 675
8.667.2.5 pRoamTimerConfig . . . 675
8.667.2.6 pTTYMode . . . 675
8.668 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference . . . 675
8.668.1 Detailed Description . . . 675
8.668.2 Field Documentation . . . 675
8.668.2.1 privacyPref . . . 675
8.669 pack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 675
8.669.1 Detailed Description . . . 675
8.669.2 Field Documentation . . . 677
8.669.2.1 pCallBarringPasswd . . . 677
8.669.2.2 pCallForwardingNumber . . . 677
8.669.2.3 pCallFwdTypeAndPlan . . . 677
8.669.2.4 pServiceClass . . . 677
8.669.2.5 pTimerVal . . . 677
8.669.2.6 reason . . . 677
8.669.2.7 voiceSvc . . . 677
8.670 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 677
8.670.1 Detailed Description . . . 677
8.670.2 Field Documentation . . . 678
8.670.2.1 DTMFdigit . . . 678
8.670.2.2 pCallID . . . 678
8.671 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 678
8.671.1 Detailed Description . . . 678
8.671.2 Field Documentation . . . 678
8.671.2.1 callID . . . 678
8.672 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference . . . 678
8.672.1 Detailed Description . . . 678
8.672.2 Field Documentation . . . 679
8.672.2.1 pEnableNotification . . . 679
8.673 pack_wds_GetDefaultProfile_t Struct Reference . . . 679
8.673.1 Detailed Description . . . 679
8.673.2 Field Documentation . . . 679
8.673.2.1 profiletype . . . 679
8.674 pack_wds_GetDefaultProfileNum_t Struct Reference . . . 679
8.674.1 Detailed Description . . . 679
8.674.2 Field Documentation . . . 679
8.674.2.1 family . . . 679
8.674.2.2 type . . . 679

8.675 pack_wds_GetDormancyState_t Struct Reference . . . 679
8.676 pack_wds_GetLastMobileIPError_t Struct Reference . . . 679
8.677 pack_wds_GetMobileIP_t Struct Reference . . . 680
8.678 pack_wds_GetMobileIPProfile_t Struct Reference . . . 680
8.678.1 Detailed Description . . . 680
8.678.2 Field Documentation . . . 680
8.678.2.1 index . . . 680
8.679 pack_wds_GetPacketStatistics_t Struct Reference . . . 680
8.679.1 Detailed Description . . . 680
8.679.2 Field Documentation . . . 680
8.679.2.1 pStatMask . . . 680
8.680 pack_wds_GetPacketStatus_t Struct Reference . . . 680
8.680.1 Detailed Description . . . 680
8.680.2 Field Documentation . . . 681
8.680.2.1 statmask . . . 681
8.681 pack_wds_GetSessionDuration_t Struct Reference . . . 681
8.682 pack_wds_RMSetTransferStatistics_t Struct Reference . . . 681
8.682.1 Detailed Description . . . 681
8.682.2 Field Documentation . . . 681
8.682.2.1 RmTrasnferStaticsReq . . . 681
8.683 pack_wds_SetAutoconnect_t Struct Reference . . . 681
8.683.1 Detailed Description . . . 681
8.683.2 Field Documentation . . . 681
8.683.2.1 acroamsetting . . . 681
8.683.2.2 acsetting . . . 681
8.684 pack_wds_SetDefaultProfile_t Struct Reference . . . 681
8.684.1 Detailed Description . . . 682
8.684.2 Field Documentation . . . 682
8.684.2.1 authentication . . . 682
8.684.2.2 ipAddress . . . 682
8.684.2.3 pApnname . . . 682
8.684.2.4 pdpType . . . 682
8.684.2.5 pName . . . 682
8.684.2.6 pPassword . . . 682
8.684.2.7 primaryDNS . . . 682
8.684.2.8 profileType . . . 682
8.684.2.9 pUsername . . . 682
8.684.2.10 secondaryDNS . . . 682
8.685 pack_wds_SetDefaultProfileNum_t Struct Reference . . . 682
8.685.1 Field Documentation . . . 683

8.685.1.1 family . . . 683
8.685.1.2 index . . . 683
8.685.1.3 type . . . 683
8.686 pack_wds_SetMobileIP_t Struct Reference . . . 683
8.686.1 Detailed Description . . . 683
8.686.2 Field Documentation . . . 683
8.686.2.1 mode . . . 683
8.687 pack_wds_SetMobileIPParameters_t Struct Reference . . . 683
8.687.1 Detailed Description . . . 683
8.687.2 Field Documentation . . . 684
8.687.2.1 pHA2002bis . . . 684
8.687.2.2 pHAAuthenticator . . . 684
8.687.2.3 pMode . . . 684
8.687.2.4 pReRegPeriod . . . 684
8.687.2.5 pReRegTraffic . . . 684
8.687.2.6 pRetryInterval . . . 684
8.687.2.7 pRetryLimit . . . 684
8.687.2.8 pSPC . . . 684
8.688 pack_wds_SetMobileIPProfile_t Struct Reference . . . 685
8.688.1 Detailed Description . . . 685
8.688.2 Field Documentation . . . 685
8.688.2.1 index . . . 685
8.688.2.2 pAAASPI . . . 685
8.688.2.3 pAddress . . . 685
8.688.2.4 pEnabled . . . 685
8.688.2.5 pHASPI . . . 685
8.688.2.6 pMNAAA . . . 685
8.688.2.7 pMNHA . . . 685
8.688.2.8 pNAI . . . 685
8.688.2.9 pPrimaryHA . . . 685
8.688.2.10 pRevTunneling . . . 686
8.688.2.11 pSecondaryHA . . . 686
8.688.2.12 spc . . . 686
8.689 pack_wds_SLQSCreateProfile_t Struct Reference . . . 686
8.689.1 Detailed Description . . . 686
8.689.2 Field Documentation . . . 686
8.689.2.1 pCurProfile . . . 686
8.689.2.2 pProfileId . . . 686
8.689.2.3 pProfileType . . . 686
8.690 pack_wds_SLQSDeleteProfile_t Struct Reference . . . 686

8.690.1 Detailed Description . . . 687
8.690.2 Field Documentation . . . 687
8.690.2.1 profileIndex . . . 687
8.690.2.2 profileType . . . 687
8.691 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 687
8.692 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 687
8.693 pack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 687
8.693.1 Detailed Description . . . 687
8.693.2 Field Documentation . . . 687
8.693.2.1 Mask . . . 687
8.693.2.2 pReportChannelRate . . . 687
8.693.2.3 pReportConnStatus . . . 687
8.693.2.4 pReportDataBearerTech . . . 687
8.693.2.5 pReportDormStatus . . . 688
8.693.2.6 pTransferStatInd . . . 688
8.694 pack_wds_SLQSGetProfileSettings_t Struct Reference . . . 688
8.694.1 Detailed Description . . . 688
8.694.2 Field Documentation . . . 688
8.694.2.1 ProfileId . . . 688
8.694.2.2 ProfileType . . . 688
8.695 pack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 688
8.695.1 Detailed Description . . . 689
8.695.2 Field Documentation . . . 689
8.695.2.1 pReqSettings . . . 689
8.696 pack_wds_SLQSModifyProfile_t Struct Reference . . . 689
8.696.1 Detailed Description . . . 689
8.696.2 Field Documentation . . . 690
8.696.2.1 curProfile . . . 690
8.696.2.2 pProfileId . . . 690
8.696.2.3 pProfileType . . . 690
8.697 pack_wds_SLQSSet3GPPConfigItem_t Struct Reference . . . 690
8.697.1 Detailed Description . . . 690
8.697.2 Field Documentation . . . 691
8.697.2.1 LTEAttachProfileListLen . . . 691
8.697.2.2 p3gppRelease . . . 691
8.697.2.3 pDefaultPDNEnabled . . . 691
8.697.2.4 pLTEAttachProfile . . . 691
8.697.2.5 pLTEAttachProfileList . . . 691
8.697.2.6 pProfileList . . . 692
8.698 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 692

8.698.1 Detailed Description . . . 692
8.698.2 Field Documentation . . . 692
8.698.2.1 IPFamilyPreference . . . 692
8.699 pack_wds_SLQSSetWdsEventCallback_t Struct Reference . . . 692
8.699.1 Detailed Description . . . 692
8.699.2 Field Documentation . . . 692
8.699.2.1 currentDataBearer . . . 692
8.699.2.2 dataBearer . . . 693
8.699.2.3 dataBearerTechExt . . . 693
8.699.2.4 dataSystemStatus . . . 693
8.699.2.5 dormancyStatus . . . 693
8.699.2.6 interval . . . 693
8.699.2.7 mobileIP . . . 693
8.699.2.8 transferStats . . . 693
8.700 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 693
8.700.1 Detailed Description . . . 693
8.700.2 Field Documentation . . . 693
8.700.2.1 pProfileId . . . 693
8.701 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference . . . 693
8.701.1 Detailed Description . . . 693
8.701.2 Field Documentation . . . 694
8.701.2.1 pHwConfig . . . 694
8.701.2.2 pProfileId . . . 694
8.701.2.3 pRequestOptionList . . . 694
8.702 pack_wds_SLQSSSetLoopback_t Struct Reference . . . 694
8.702.1 Detailed Description . . . 694
8.702.2 Field Documentation . . . 694
8.702.2.1 loopbackMode . . . 694
8.702.2.2 loopbackMultiplier . . . 694
8.703 pack_wds_SLQSStartDataSession_t Struct Reference . . . 694
8.703.1 Detailed Description . . . 694
8.703.2 Field Documentation . . . 695
8.703.2.1 pAuth . . . 695
8.703.2.2 pPass . . . 695
8.703.2.3 pprofileid3gpp . . . 695
8.703.2.4 pprofileid3gpp2 . . . 695
8.703.2.5 pTech . . . 695
8.703.2.6 pUser . . . 695
8.704 pack_wds_SLQSStopDataSession_t Struct Reference . . . 695
8.704.1 Detailed Description . . . 696

8.704.2 Field Documentation . . . 696
8.704.2.1 psid . . . 696
8.705 pack_wds_SLQSWdsSetEventReport_t Struct Reference . . . 696
8.705.1 Detailed Description . . . 696
8.705.2 Field Documentation . . . 697
8.705.2.1 pCurrChannelRateInd . . . 697
8.705.2.2 pCurrDataBearerTechInd . . . 697
8.705.2.3 pCurrPrefDataSysInd . . . 697
8.705.2.4 pDataBearerTechInd . . . 697
8.705.2.5 pDataCallStatusChangeInd . . . 697
8.705.2.6 pDataSystemStatusChangeInd . . . 697
8.705.2.7 pDormancyStatusInd . . . 697
8.705.2.8 pEVDOPageMonPerChangeInd . . . 697
8.705.2.9 pMIPStatusInd . . . 697
8.705.2.10 pTransferStatInd . . . 697
8.706 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 698
8.706.1 Detailed Description . . . 698
8.706.2 Field Documentation . . . 698
8.706.2.1 contextId . . . 698
8.706.2.2 contextType . . . 698
8.707 PackCreateProfileOut Struct Reference . . . 698
8.707.1 Field Documentation . . . 698
8.707.1.1 ExtErrorCode . . . 698
8.707.1.2 ProfileIndex . . . 698
8.707.1.3 ProfileType . . . 698
8.708 packgetDyingGaspCfg Struct Reference . . . 698
8.708.1 Detailed Description . . . 698
8.708.2 Field Documentation . . . 699
8.708.2.1 pDestSMSContent . . . 699
8.708.2.2 pDestSMSNum . . . 699
8.709 packgetDyingGaspStatistics Struct Reference . . . 699
8.709.1 Detailed Description . . . 699
8.709.2 Field Documentation . . . 699
8.709.2.1 pSMSAttemptedFlag . . . 699
8.709.2.2 pTimeStamp . . . 699
8.710 PCMparams Struct Reference . . . 699
8.710.1 Detailed Description . . . 699
8.710.2 Field Documentation . . . 700
8.710.2.1 iFaceTab . . . 700
8.710.2.2 iFaceTabLen . . . 700

8.711 PCSCFFQDNAddress Struct Reference . . . 700
8.711.1 Detailed Description . . . 700
8.711.2 Field Documentation . . . 700
8.711.2.1 fqdnAddr . . . 700
8.711.2.2 fqdnLen . . . 700
8.712 PCSCFFQDNAddressList Struct Reference . . . 700
8.712.1 Detailed Description . . . 700
8.712.2 Field Documentation . . . 701
8.712.2.1 numInstances . . . 701
8.712.2.2 pcsfFQDNAddress . . . 701
8.713 PCSCFIPv4ServerAddressList Struct Reference . . . 701
8.713.1 Detailed Description . . . 701
8.713.2 Field Documentation . . . 701
8.713.2.1 numInstances . . . 701
8.713.2.2 pscsfIPv4Addr . . . 701
8.714 PDSPositionData Struct Reference . . . 701
8.714.1 Detailed Description . . . 702
8.714.2 Field Documentation . . . 703
8.714.2.1 pAltitudeWrtEllipsoid . . . 703
8.714.2.2 pAltitudeWrtSealevel . . . 703
8.714.2.3 pHorizontalConfidence . . . 703
8.714.2.4 pHorizontalUncCircular . . . 703
8.714.2.5 pLatitude . . . 703
8.714.2.6 pLongitude . . . 703
8.714.2.7 pPositionSource . . . 703
8.714.2.8 pTimeStamp . . . 703
8.714.2.9 pTimeType . . . 703
8.714.2.10 pVerticalConfidence . . . 703
8.714.2.11 pVerticalUnc . . . 703
8.715 PDSPosMethodStateReq Struct Reference . . . 703
8.715.1 Detailed Description . . . 704
8.715.2 Field Documentation . . . 704
8.715.2.1 pWifiState . . . 704
8.715.2.2 pXtraDataState . . . 704
8.715.2.3 pXtraTimeState . . . 704
8.716 peerNumberInfo Struct Reference . . . 704
8.716.1 Detailed Description . . . 704
8.716.2 Field Documentation . . . 705
8.716.2.1 callID . . . 706
8.716.2.2 number . . . 706

8.716.2.3 numLen . . . 706
8.716.2.4 numPI . . . 706
8.716.2.5 numPlan . . . 706
8.716.2.6 numSI . . . 706
8.716.2.7 numType . . . 706
8.717 personalizationStatus Struct Reference . . . 706
8.717.1 Detailed Description . . . 706
8.717.2 Field Documentation . . . 707
8.717.2.1 feature . . . 707
8.717.2.2 numFeatures . . . 707
8.717.2.3 unblockLeft . . . 707
8.717.2.4 verifyLeft . . . 707
8.718 PhyCaAggPcellInfo Struct Reference . . . 707
8.718.1 Detailed Description . . . 707
8.718.2 Field Documentation . . . 707
8.718.2.1 dl_bw_value . . . 707
8.718.2.2 freq . . . 707
8.718.2.3 iLTEbandValue . . . 707
8.718.2.4 pci . . . 708
8.718.2.5 TlvPresent . . . 708
8.719 PhyCaAggScellDlBw Struct Reference . . . 708
8.719.1 Detailed Description . . . 708
8.719.2 Field Documentation . . . 708
8.719.2.1 dl_bw_value . . . 708
8.719.2.2 TlvPresent . . . 708
8.720 PhyCaAggScellIndex Struct Reference . . . 708
8.720.1 Detailed Description . . . 708
8.720.2 Field Documentation . . . 708
8.720.2.1 scell_idx . . . 709
8.720.2.2 TlvPresent . . . 709
8.721 PhyCaAggScellIndType Struct Reference . . . 709
8.721.1 Detailed Description . . . 709
8.721.2 Field Documentation . . . 709
8.721.2.1 freq . . . 709
8.721.2.2 pci . . . 709
8.721.2.3 scell_state . . . 709
8.721.2.4 TlvPresent . . . 709
8.722 PhyCaAggScellInfo Struct Reference . . . 709
8.722.1 Detailed Description . . . 710
8.722.2 Field Documentation . . . 712

8.722.2.1 dl_bw_value . . . 712
8.722.2.2 freq . . . 712
8.722.2.3 iLTEbandValue . . . 712
8.722.2.4 pci . . . 712
8.722.2.5 scell_state . . . 712
8.722.2.6 TlvPresent . . . 712
8.723 PilotSetData Struct Reference . . . 712
8.723.1 Detailed Description . . . 712
8.723.2 Field Documentation . . . 712
8.723.2.1 NumPilots . . . 712
8.723.2.2 pPilotSetInfo . . . 712
8.724 PilotSetParams Struct Reference . . . 713
8.724.1 Detailed Description . . . 713
8.724.2 Field Documentation . . . 713
8.724.2.1 PilotPN . . . 713
8.724.2.2 PilotStrength . . . 713
8.724.2.3 PilotType . . . 713
8.725 pktErrRate Struct Reference . . . 713
8.725.1 Detailed Description . . . 713
8.725.2 Field Documentation . . . 713
8.725.2.1 exponent . . . 713
8.725.2.2 multiplier . . . 714
8.726 PlmnID Struct Reference . . . 714
8.726.1 Detailed Description . . . 714
8.726.2 Field Documentation . . . 714
8.726.2.1 mcc . . . 714
8.726.2.2 mnc . . . 714
8.726.2.3 pcsDigit . . . 714
8.726.2.4 TlvPresent . . . 714
8.727 PLMNNetworkName Struct Reference . . . 714
8.727.1 Detailed Description . . . 715
8.727.2 Field Documentation . . . 715
8.727.2.1 numInstance . . . 715
8.727.2.2 PLMNNetName . . . 715
8.728 PLMNNetworkNameData Struct Reference . . . 715
8.728.1 Detailed Description . . . 715
8.728.2 Field Documentation . . . 716
8.728.2.1 codingScheme . . . 716
8.728.2.2 countryInitials . . . 716
8.728.2.3 longName . . . 716

8.728.2.4 longNameLen . . . 716
8.728.2.5 longNameSpareBits . . . 716
8.728.2.6 shortName . . . 716
8.728.2.7 shortNameLen . . . 717
8.728.2.8 shortNameSpareBits . . . 717
8.729 Port Struct Reference . . . 717
8.729.1 Detailed Description . . . 717
8.729.2 Field Documentation . . . 717
8.729.2.1 port . . . 717
8.729.2.2 range . . . 717
8.730 precisionDilution_s Struct Reference . . . 717
8.730.1 Detailed Description . . . 717
8.730.2 Field Documentation . . . 718
8.730.2.1 HDOP . . . 718
8.730.2.2 PDOP . . . 718
8.730.2.3 VDOP . . . 718
8.731 PrefImageList Struct Reference . . . 718
8.731.1 Detailed Description . . . 718
8.731.2 Field Documentation . . . 718
8.731.2.1 listEntries . . . 718
8.731.2.2 listSize . . . 718
8.732 prefVoiceSO Struct Reference . . . 718
8.732.1 Detailed Description . . . 719
8.732.2 Field Documentation . . . 720
8.732.2.1 evrcCapability . . . 720
8.732.2.2 homeOrigVoiceSO . . . 720
8.732.2.3 homePageVoiceSO . . . 720
8.732.2.4 namID . . . 720
8.732.2.5 roamOrigVoiceSO . . . 720
8.733 Profile3GPP Struct Reference . . . 720
8.733.1 Detailed Description . . . 721
8.733.2 Field Documentation . . . 725
8.733.2.1 pAddrAllocPref . . . 725
8.733.2.2 pAPNClass . . . 725
8.733.2.3 pAPNDisabledFlag . . . 725
8.733.2.4 pAPNName . . . 725
8.733.2.5 pAPNnameSize . . . 725
8.733.2.6 pAuthenticationPref . . . 725
8.733.2.7 pGPRSMinimumQoS . . . 725
8.733.2.8 pGPRSRequestedQos . . . 725

8.733.2.9 pImCnFlag . . . 725
8.733.2.10 pIPv4AddrPref . . . 725
8.733.2.11 pIPv6AddPref . . . 725
8.733.2.12 pPassword . . . 725
8.733.2.13 pPasswordSize . . . 725
8.733.2.14 pPcscfAddrUsingDhcp . . . 725
8.733.2.15 pPcscfAddrUsingPCO . . . 725
8.733.2.16 pPDNInactivTimeout . . . 725
8.733.2.17 pPdpAccessConFlag . . . 725
8.733.2.18 pPdpContext . . . 725
8.733.2.19 pPdpDataCompType . . . 725
8.733.2.20 pPdpHdrCompType . . . 725
8.733.2.21 pPDPtype . . . 725
8.733.2.22 pPriDNSIPv4AddPref . . . 725
8.733.2.23 pPriDNSIPv6addpref . . . 725
8.733.2.24 pPrimaryID . . . 726
8.733.2.25 pProfilename . . . 726
8.733.2.26 pProfilenameSize . . . 726
8.733.2.27 pQosClassID . . . 726
8.733.2.28 pSecDNSIPv4AddPref . . . 726
8.733.2.29 pSecDNSIPv6addpref . . . 726
8.733.2.30 pSecondaryFlag . . . 726
8.733.2.31 pTFTID1Params . . . 726
8.733.2.32 pTFTID2Params . . . 726
8.733.2.33 pUMTSMinQoS . . . 726
8.733.2.34 pUMTSMinQosSigInd . . . 726
8.733.2.35 pUMTSReqQoS . . . 726
8.733.2.36 pUMTSReqQoSSigInd . . . 726
8.733.2.37 pUsername . . . 726
8.733.2.38 pUsernameSize . . . 726
8.734 Profile3GPP2 Struct Reference . . . 726
8.734.1 Detailed Description . . . 727
8.734.2 Field Documentation . . . 730
8.734.2.1 pAllowLinger . . . 730
8.734.2.2 pAPNClass3GPP2 . . . 730
8.734.2.3 pAPNEnabled3GPP2 . . . 730
8.734.2.4 pApnString . . . 730
8.734.2.5 pApnStringSize . . . 730
8.734.2.6 pAppPriority . . . 730
8.734.2.7 pAppType . . . 730

8.734.2.8 pAuthPassword . . . 730
8.734.2.9 pAuthPasswordSize . . . 730
8.734.2.10 pAuthProtocol . . . 731
8.734.2.11 pAuthRetryCount . . . 731
8.734.2.12 pAuthTimeout . . . 731
8.734.2.13 pDataMode . . . 731
8.734.2.14 pDataRate . . . 731
8.734.2.15 pIpcpAckTimeout . . . 731
8.734.2.16 pIpcpCreqRetryCount . . . 731
8.734.2.17 pIsPcscfAddressNedded . . . 731
8.734.2.18 pLcpAckTimeout . . . 731
8.734.2.19 pLcpCreqRetryCount . . . 731
8.734.2.20 pNegoDnsSrvrPref . . . 731
8.734.2.21 pPDNInactivTimeout3GPP2 . . . 731
8.734.2.22 pPdnType . . . 731
8.734.2.23 pPppSessCloseTimer1x . . . 731
8.734.2.24 pPppSessCloseTimerDO . . . 731
8.734.2.25 pPrimaryV4DnsAddress . . . 731
8.734.2.26 pPriV6DnsAddress . . . 731
8.734.2.27 pRATType . . . 731
8.734.2.28 pSecondaryV4DnsAddress . . . 731
8.734.2.29 pSecV6DnsAddress . . . 731
8.734.2.30 pUserId . . . 731
8.734.2.31 pUserIdSize . . . 731
8.735 ProfileIdentifier Struct Reference . . . 731
8.735.1 Detailed Description . . . 731
8.735.2 Field Documentation . . . 732
8.735.2.1 profileIndex . . . 732
8.735.2.2 profileType . . . 732
8.736 protocolSubtypeElement Struct Reference . . . 732
8.736.1 Detailed Description . . . 732
8.736.2 Field Documentation . . . 733
8.736.2.1 AccessMac . . . 733
8.736.2.2 AuthProt . . . 733
8.736.2.3 ControlMac . . . 733
8.736.2.4 EncryptProt . . . 733
8.736.2.5 ForwardMac . . . 733
8.736.2.6 IdleState . . . 733
8.736.2.7 KeyExchange . . . 733
8.736.2.8 MultDisc . . . 733

8.736.2.9 PhysicalLayer . . . 733
8.736.2.10 ReverseMac . . . 733
8.736.2.11 SecProt . . . 733
8.736.2.12 VirtStream . . . 733
8.737 PSDetachReq Struct Reference . . . 733
8.737.1 Detailed Description . . . 734
8.737.2 Field Documentation . . . 734
8.737.2.1 pDetachAction . . . 734
8.738 qaQmi3Gpp2TimeZone Struct Reference . . . 734
8.738.1 Detailed Description . . . 734
8.738.2 Field Documentation . . . 734
8.738.2.1 daylightSavings . . . 734
8.738.2.2 leapSeconds . . . 734
8.738.2.3 localTimeOffset . . . 735
8.739 qaQmiInterfaceInfo Struct Reference . . . 735
8.739.1 Detailed Description . . . 735
8.739.2 Field Documentation . . . 735
8.739.2.1 qaQmiinstanceid . . . 735
8.739.2.2 qaQmisvctype . . . 735
8.739.2.3 v4sessionId . . . 735
8.739.2.4 v6sessionId . . . 735
8.740 qaQmiServingSystemParam Struct Reference . . . 735
8.740.1 Detailed Description . . . 736
8.740.2 Field Documentation . . . 738
8.740.2.1 BasestationID . . . 738
8.740.2.2 BasestationLatitude . . . 738
8.740.2.3 BasestationLongitude . . . 738
8.740.2.4 CallBarStatus . . . 738
8.740.2.5 CDMA_P_Rev . . . 738
8.740.2.6 CDMASystemInfoExt . . . 738
8.740.2.7 CellID . . . 738
8.740.2.8 concSvcInfo . . . 738
8.740.2.9 CurrentPLMN . . . 739
8.740.2.10 DataSrvCapabilities . . . 739
8.740.2.11 defaultRoamInd . . . 739
8.740.2.12 DetailedSvcInfo . . . 739
8.740.2.13 DTMInd . . . 739
8.740.2.14 Gpp2TimeZone . . . 739
8.740.2.15 GppNetworkDSTAdjustment . . . 739
8.740.2.16 GppTimeZone . . . 739

8.740.2.17 hdrPersonality . . . 739
8.740.2.18 Lac . . . 739
8.740.2.19 NetworkID . . . 739
8.740.2.20 PRLInd . . . 739
8.740.2.21 roamIndicatorVal . . . 739
8.740.2.22 RoamingIndicatorList . . . 739
8.740.2.23 ServingSystem . . . 739
8.740.2.24 SystemID . . . 739
8.740.2.25 trackAreaCode . . . 739
8.741 QmiCbkCatEventStatusReportInd Struct Reference . . . 739
8.741.1 Field Documentation . . . 739
8.741.1.1 CCETlv . . . 739
8.741.1.2 event_Index . . . 739
8.742 QmiCbkLocBestAvailPosInd Struct Reference . . . 739
8.742.1 Detailed Description . . . 740
8.742.2 Field Documentation . . . 744
8.742.2.1 pAltitudeWrtEllipsoid . . . 744
8.742.2.2 pAltitudeWrtMeanSeaLevel . . . 744
8.742.2.3 pGpsTime . . . 744
8.742.2.4 pHeading . . . 744
8.742.2.5 pHeadingUnc . . . 744
8.742.2.6 pHorCirConf . . . 744
8.742.2.7 pHorEllpConf . . . 744
8.742.2.8 pHorReliability . . . 744
8.742.2.9 pHorUncCircular . . . 744
8.742.2.10 pHorUncEllipseOrientAzimuth . . . 744
8.742.2.11 pHorUncEllipseSemiMajor . . . 744
8.742.2.12 pHorUncEllipseSemiMinor . . . 745
8.742.2.13 pLatitude . . . 745
8.742.2.14 pLongitude . . . 745
8.742.2.15 pMagneticDeviation . . . 745
8.742.2.16 pPrecisionDilution . . . 745
8.742.2.17 pSensorDataUsage . . . 745
8.742.2.18 pSpeedHorizontal . . . 745
8.742.2.19 pSpeedUnc . . . 745
8.742.2.20 pSpeedVertical . . . 745
8.742.2.21 pSpeedVerticalUnc . . . 745
8.742.2.22 pSvUsedforFix . . . 745
8.742.2.23 pTechnologyMask . . . 745
8.742.2.24 pTimeSrc . . . 745

8.742.2.25 pTimestampUtc . . . 745
8.742.2.26 pTimeUnc . . . 745
8.742.2.27 pVertConfidence . . . 745
8.742.2.28 pVertReliability . . . 745
8.742.2.29 pVertUnc . . . 745
8.742.2.30 pXid . . . 745
8.742.2.31 status . . . 745
8.743 QmiCbkLocCradleMountInd Struct Reference . . . 745
8.743.1 Detailed Description . . . 745
8.743.2 Field Documentation . . . 746
8.743.2.1 cradleMountConfigStatus . . . 746
8.744 QmiCbkLocEngineStateInd Struct Reference . . . 746
8.744.1 Detailed Description . . . 746
8.744.2 Field Documentation . . . 746
8.744.2.1 engineState . . . 746
8.745 QmiCbkLocEventTimeSyncInd Struct Reference . . . 746
8.745.1 Detailed Description . . . 747
8.745.2 Field Documentation . . . 747
8.745.2.1 timeSyncRefCounter . . . 747
8.746 QmiCbkLocInjectPositionInd Struct Reference . . . 747
8.746.1 Detailed Description . . . 747
8.746.2 Field Documentation . . . 747
8.746.2.1 status . . . 748
8.747 QmiCbkLocInjectSensorDataInd Struct Reference . . . 748
8.747.1 Detailed Description . . . 748
8.747.2 Field Documentation . . . 749
8.747.2.1 injectSensorDataStatus . . . 749
8.747.2.2 pAccelSamplesAccepted . . . 749
8.747.2.3 pAccelTempSamplesAccepted . . . 749
8.747.2.4 pGyroSamplesAccepted . . . 749
8.747.2.5 pGyroTempSamplesAccepted . . . 749
8.747.2.6 pOpaqueIdentifier . . . 749
8.748 QmiCbkLocInjectTimeInd Struct Reference . . . 749
8.748.1 Detailed Description . . . 749
8.748.2 Field Documentation . . . 749
8.748.2.1 injectTimeSyncStatus . . . 749
8.749 QmiCbkLocInjectUTCTimeInd Struct Reference . . . 749
8.749.1 Detailed Description . . . 750
8.749.2 Field Documentation . . . 750
8.749.2.1 status . . . 750

8.750 QmiCbkLocPositionReportInd Struct Reference . . . 750
8.750.1 Detailed Description . . . 751
8.750.2 Field Documentation . . . 755
8.750.2.1 pAltitudeAssumed . . . 755
8.750.2.2 pAltitudeWrtEllipsoid . . . 755
8.750.2.3 pAltitudeWrtMeanSeaLevel . . . 755
8.750.2.4 pFixId . . . 755
8.750.2.5 pGpsTime . . . 755
8.750.2.6 pHeading . . . 755
8.750.2.7 pHeadingUnc . . . 755
8.750.2.8 pHorConfidence . . . 755
8.750.2.9 pHorReliability . . . 755
8.750.2.10 pHorUncCircular . . . 755
8.750.2.11 pHorUncEllipseOrientAzimuth . . . 755
8.750.2.12 pHorUncEllipseSemiMajor . . . 755
8.750.2.13 pHorUncEllipseSemiMinor . . . 755
8.750.2.14 pLatitude . . . 755
8.750.2.15 pLeapSeconds . . . 755
8.750.2.16 pLongitude . . . 755
8.750.2.17 pMagneticDeviation . . . 755
8.750.2.18 pPrecisionDilution . . . 755
8.750.2.19 pSensorDataUsage . . . 755
8.750.2.20 pSpeedHorizontal . . . 755
8.750.2.21 pSpeedUnc . . . 755
8.750.2.22 pSpeedVertical . . . 755
8.750.2.23 pSvUsedforFix . . . 755
8.750.2.24 pTechnologyMask . . . 755
8.750.2.25 pTimeSrc . . . 755
8.750.2.26 pTimestampUtc . . . 755
8.750.2.27 pTimeUnc . . . 755
8.750.2.28 pVertConfidence . . . 756
8.750.2.29 pVertReliability . . . 756
8.750.2.30 pVertUnc . . . 756
8.750.2.31 sessionId . . . 756
8.750.2.32 sessionStatus . . . 756
8.751 QmiCbkLocSensorStreamingInd Struct Reference . . . 756
8.751.1 Detailed Description . . . 756
8.751.2 Field Documentation . . . 756
8.751.2.1 pAccelAcceptReady . . . 756
8.751.2.2 pAccelTempAcceptReady . . . 756

8.751.2.3 pGyroAcceptReady . . . 756
8.751.2.4 pGyroTempAcceptReady . . . 756
8.752 QmiCbkLocSetExtPowerConfigInd Struct Reference . . . 756
8.752.1 Detailed Description . . . 757
8.752.2 Field Documentation . . . 757
8.752.2.1 status . . . 757
8.753 QmiCbkNasLTECphyCaInfo Struct Reference . . . 757
8.753.1 Detailed Description . . . 757
8.753.2 Field Documentation . . . 758
8.753.2.1 sPhyCaAggPcellInfo . . . 758
8.753.2.2 sPhyCaAggScellDlBw . . . 758
8.753.2.3 sPhyCaAggScellIndex . . . 758
8.753.2.4 sPhyCaAggScellIndType . . . 758
8.753.2.5 sPhyCaAggScellInfo . . . 758
8.754 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference . . . 758
8.754.1 Field Documentation . . . 758
8.754.1.1 SITlv . . . 758
8.755 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference . . . 758
8.755.1 Field Documentation . . . 758
8.755.1.1 SITlv . . . 758
8.756 QmiCbkTmdMitiLvlRptInd Struct Reference . . . 758
8.756.1 Detailed Description . . . 758
8.756.2 Field Documentation . . . 759
8.756.2.1 currentMitigationLvl . . . 759
8.756.2.2 MitigationDevInfo . . . 759
8.757 QmiCbkWdsStatisticsIndState Struct Reference . . . 759
8.757.1 Detailed Description . . . 759
8.757.2 Field Documentation . . . 759
8.757.2.1 RxDropConutTlv . . . 760
8.757.2.2 RxOkByteCountTlv . . . 760
8.757.2.3 RxOkConutTlv . . . 760
8.757.2.4 TxDropConutTlv . . . 760
8.757.2.5 TxOkByteCountTlv . . . 760
8.757.2.6 TxOkConutTlv . . . 760
8.758 qmifwinfo_s Struct Reference . . . 760
8.758.1 Detailed Description . . . 760
8.758.2 Field Documentation . . . 760
8.758.2.1 dev . . . 761
8.758.2.2 g . . . 761
8.758.2.3 s . . . 761

8.759 QmiNas3GppNetworkInfo Struct Reference . . . 761
8.759.1 Detailed Description . . . 761
8.759.2 Field Documentation . . . 762
8.759.2.1 pDesription . . . 762
8.759.2.2 pForbidden . . . 762
8.759.2.3 pInUse . . . 762
8.759.2.4 pMCC . . . 762
8.759.2.5 pMNC . . . 762
8.759.2.6 pPreferred . . . 762
8.759.2.7 pRoaming . . . 762
8.760 QmiNasGetRFBandInfoResp Struct Reference . . . 762
8.760.1 Field Documentation . . . 762
8.760.1.1 pInstancesSize . . . 762
8.760.1.2 pRFBandInfoElements . . . 762
8.760.1.3 results . . . 762
8.761 QmiNasPerformNetworkScanResp Struct Reference . . . 762
8.761.1 Field Documentation . . . 763
8.761.1.1 pInstances . . . 763
8.761.1.2 pInstanceSize . . . 763
8.761.1.3 results . . . 763
8.762 qmiSmsMessageList Struct Reference . . . 763
8.762.1 Detailed Description . . . 763
8.762.2 Field Documentation . . . 763
8.762.2.1 messageIndex . . . 763
8.762.2.2 messageTag . . . 763
8.763 qmiWDSDataBearerTechnology Struct Reference . . . 763
8.763.1 Detailed Description . . . 763
8.763.2 Field Documentation . . . 763
8.763.2.1 currentNetwork . . . 763
8.763.2.2 ratMask . . . 764
8.763.2.3 soMask . . . 764
8.764 QmiWdsIpAddressInfo Struct Reference . . . 764
8.764.1 Detailed Description . . . 764
8.764.2 Field Documentation . . . 764
8.764.2.1 pIPAddressV4 . . . 764
8.764.2.2 pIPAddressV6 . . . 764
8.764.2.3 pIPv6prefixlen . . . 764
8.765 qmiWdsRunTimeSettings Struct Reference . . . 764
8.765.1 Detailed Description . . . 765
8.765.2 Field Documentation . . . 767

8.765.2.1 pAPNName . . . 767
8.765.2.2 pAuthentication . . . 767
8.765.2.3 pDomainList . . . 767
8.765.2.4 pGPRSGrantedQoS . . . 767
8.765.2.5 pGWAddressV4 . . . 767
8.765.2.6 pIMCNflag . . . 767
8.765.2.7 pIPAddressV4 . . . 767
8.765.2.8 pIPFamilyPreference . . . 767
8.765.2.9 pIPV6AddrInfo . . . 767
8.765.2.10 pIPV6GWAddrInfo . . . 767
8.765.2.11 pMtu . . . 767
8.765.2.12 pPCSCFAddrPCO . . . 767
8.765.2.13 pPCSCFFQDNAddrList . . . 767
8.765.2.14 pPDPType . . . 768
8.765.2.15 pPrimaryDNSV4 . . . 768
8.765.2.16 pPrimaryDNSV6 . . . 768
8.765.2.17 pProfileID . . . 768
8.765.2.18 pProfileName . . . 768
8.765.2.19 pSecondaryDNSV4 . . . 768
8.765.2.20 pSecondaryDNSV6 . . . 768
8.765.2.21 pServerAddrList . . . 768
8.765.2.22 pSubnetMaskV4 . . . 768
8.765.2.23 pTechnology . . . 768
8.765.2.24 pUMTSGrantedQoS . . . 768
8.765.2.25 pUsername . . . 768
8.766 QosClassID Struct Reference . . . 768
8.766.1 Detailed Description . . . 768
8.766.2 Field Documentation . . . 769
8.766.2.1 gDlBitRate . . . 769
8.766.2.2 gUlBitRate . . . 769
8.766.2.3 maxDlBitRate . . . 769
8.766.2.4 maxUlBitRate . . . 769
8.766.2.5 QCI . . . 769
8.767 QosEventInfo Struct Reference . . . 769
8.767.1 Detailed Description . . . 769
8.767.2 Field Documentation . . . 770
8.767.2.1 pDataBearer . . . 770
8.767.2.2 pPacketsCountRX . . . 770
8.767.2.3 pPacketsCountTX . . . 770
8.767.2.4 pTotalBytesRX . . . 770

8.767.2.5 pTotalBytesTX . . . 771
8.768 QosFlowInfo Struct Reference . . . 771
8.768.1 Detailed Description . . . 771
8.768.2 Field Documentation . . . 771
8.768.2.1 pBearerID . . . 771
8.768.2.2 pQFlowState . . . 771
8.768.2.3 pRxQFilter . . . 771
8.768.2.4 pRxQFlowGranted . . . 771
8.768.2.5 pTxQFilter . . . 771
8.768.2.6 pTxQFlowGranted . . . 772
8.769 QosFlowInfoState Struct Reference . . . 772
8.769.1 Detailed Description . . . 772
8.769.2 Field Documentation . . . 772
8.769.2.1 id . . . 772
8.769.2.2 isNewFlow . . . 772
8.769.2.3 state . . . 772
8.770 QosMap Struct Reference . . . 772
8.770.1 Detailed Description . . . 772
8.770.2 Field Documentation . . . 773
8.770.2.1 dscp . . . 773
8.770.2.2 qos_id . . . 773
8.770.2.3 state . . . 773
8.771 RankIndicatorInd Struct Reference . . . 773
8.771.1 Field Documentation . . . 773
8.771.1.1 Count1 . . . 773
8.771.1.2 Count2 . . . 773
8.772 readResult Struct Reference . . . 773
8.772.1 Detailed Description . . . 773
8.772.2 Field Documentation . . . 774
8.772.2.1 content . . . 774
8.772.2.2 contentLen . . . 774
8.773 readTransparentInfo Struct Reference . . . 774
8.773.1 Detailed Description . . . 774
8.773.2 Field Documentation . . . 774
8.773.2.1 length . . . 774
8.773.2.2 offset . . . 774
8.774 redirNumInfo Struct Reference . . . 774
8.774.1 Detailed Description . . . 774
8.774.2 Field Documentation . . . 776
8.774.2.1 number . . . 776

8.774.2.2 numLen . . . 776
8.774.2.3 numPlan . . . 776
8.774.2.4 numType . . . 776
8.774.2.5 PI . . . 776
8.774.2.6 reason . . . 776
8.774.2.7 SI . . . 776
8.775 registerRefresh Struct Reference . . . 776
8.775.1 Detailed Description . . . 776
8.775.2 Field Documentation . . . 777
8.775.2.1 arrfileInfo . . . 777
8.775.2.2 numFiles . . . 777
8.775.2.3 registerFlag . . . 777
8.775.2.4 voteForInit . . . 777
8.776 remainingRetries Struct Reference . . . 777
8.776.1 Detailed Description . . . 777
8.776.2 Field Documentation . . . 777
8.776.2.1 unblockLeft . . . 777
8.776.2.2 verifyLeft . . . 777
8.777 remotePartyName Struct Reference . . . 777
8.777.1 Detailed Description . . . 778
8.777.2 Field Documentation . . . 778
8.777.2.1 callerName . . . 778
8.777.2.2 codingScheme . . . 778
8.777.2.3 nameLen . . . 778
8.777.2.4 namePI . . . 778
8.778 remotePartyNum Struct Reference . . . 778
8.778.1 Detailed Description . . . 778
8.778.2 Field Documentation . . . 779
8.778.2.1 numLen . . . 779
8.778.2.2 presentationInd . . . 779
8.778.2.3 remPartyNumber . . . 779
8.779 ReqFieldsList Struct Reference . . . 779
8.779.1 Detailed Description . . . 779
8.779.2 Field Documentation . . . 779
8.779.2.1 requestFields . . . 779
8.779.2.2 requestFieldsLen . . . 780
8.780 RespFieldsList Struct Reference . . . 780
8.780.1 Detailed Description . . . 780
8.780.2 Field Documentation . . . 780
8.780.2.1 responseFields . . . 780

8.780.2.2 responseFieldsLen . . . 780
8.781 RFBandInfoElements Struct Reference . . . 780
8.781.1 Detailed Description . . . 780
8.781.2 Field Documentation . . . 781
8.781.2.1 activeBandClass . . . 781
8.781.2.2 activeBandClass . . . 781
8.781.2.3 activeChannel . . . 781
8.781.2.4 activeChannel . . . 781
8.781.2.5 radioInterface . . . 781
8.781.2.6 radioInterface . . . 781
8.782 rmTrasnferStaticsReq Struct Reference . . . 781
8.782.1 Detailed Description . . . 781
8.782.2 Field Documentation . . . 781
8.782.2.1 bResetStatistics . . . 781
8.782.2.2 ulMask . . . 781
8.783 roamIndList Struct Reference . . . 781
8.783.1 Detailed Description . . . 782
8.783.2 Field Documentation . . . 782
8.783.2.1 numInstances . . . 782
8.783.2.2 radioInterface . . . 782
8.783.2.3 roamIndicator . . . 782
8.784 RoamingInfo Struct Reference . . . 782
8.784.1 Field Documentation . . . 782
8.784.1.1 roaming_ind . . . 782
8.784.1.2 TlvPresent . . . 782
8.785 roamTimer Struct Reference . . . 782
8.785.1 Detailed Description . . . 783
8.785.2 Field Documentation . . . 783
8.785.2.1 namID . . . 783
8.785.2.2 roamTimerValue . . . 783
8.786 RSRPThresh Struct Reference . . . 783
8.786.1 Detailed Description . . . 783
8.786.2 Field Documentation . . . 784
8.786.2.1 pRSRPThresList . . . 784
8.786.2.2 RSRPThresListLen . . . 784
8.787 rsrqInformation Struct Reference . . . 784
8.787.1 Detailed Description . . . 784
8.787.2 Field Documentation . . . 784
8.787.2.1 radioIf . . . 784
8.787.2.2 rsrq . . . 784

8.788 RSRQThresh Struct Reference . . . 784
8.788.1 Detailed Description . . . 785
8.788.2 Field Documentation . . . 785
8.788.2.1 pRSRQThresList . . . 785
8.788.2.2 RSRQThresListLen . . . 785
8.789 RSSIThresh Struct Reference . . . 785
8.789.1 Detailed Description . . . 785
8.789.2 Field Documentation . . . 786
8.789.2.1 pRSSIThresList . . . 786
8.789.2.2 RSSIThresListLen . . . 786
8.790 RXAGCList Struct Reference . . . 786
8.790.1 Detailed Description . . . 786
8.790.2 Field Documentation . . . 787
8.790.2.1 pRXAIG . . . 787
8.790.2.2 pRXComprSlope . . . 787
8.790.2.3 pRXComprThres . . . 787
8.790.2.4 pRXExpSlope . . . 787
8.790.2.5 pRXExpThres . . . 787
8.790.2.6 pRXStaticGain . . . 787
8.791 RXAVCList Struct Reference . . . 787
8.791.1 Detailed Description . . . 787
8.791.2 Field Documentation . . . 787
8.791.2.1 pAVRXAVCHeadroom . . . 787
8.791.2.2 pAVRXAVCSens . . . 787
8.792 rxInfo Struct Reference . . . 787
8.792.1 Detailed Description . . . 788
8.792.2 Field Documentation . . . 788
8.792.2.1 ecio . . . 788
8.792.2.2 isRadioTuned . . . 788
8.792.2.3 phase . . . 788
8.792.2.4 rscp . . . 788
8.792.2.5 rsrp . . . 788
8.792.2.6 rxPower . . . 788
8.793 RXPCMIIRFltr Struct Reference . . . 789
8.793.1 Detailed Description . . . 789
8.793.2 Field Documentation . . . 790
8.793.2.1 pFlag . . . 790
8.793.2.2 pStage0Val . . . 790
8.793.2.3 pStage1Val . . . 790
8.793.2.4 pStage2Val . . . 790

8.793.2.5 pStage3Val . . . 790
8.793.2.6 pStage4Val . . . 790
8.793.2.7 pStageCnt . . . 790
8.794 RxSigInfo Struct Reference . . . 790
8.794.1 Detailed Description . . . 790
8.794.2 Field Documentation . . . 791
8.794.2.1 isRadioTuned . . . 791
8.794.2.2 rsrp . . . 791
8.794.2.3 rxChainIndex . . . 791
8.794.2.4 rxPower . . . 791
8.795 rxSignalStrengthListElement Struct Reference . . . 791
8.795.1 Detailed Description . . . 791
8.795.2 Field Documentation . . . 792
8.795.2.1 radioIf . . . 792
8.795.2.2 rxSignalStrength . . . 792
8.796 sApnExtraParams Struct Reference . . . 792
8.796.1 Detailed Description . . . 792
8.796.2 Field Documentation . . . 793
8.796.2.1 ambr_dl . . . 793
8.796.2.2 ambr_dl_ext . . . 793
8.796.2.3 ambr_dl_ext2 . . . 793
8.796.2.4 ambr_ul . . . 793
8.796.2.5 ambr_ul_ext . . . 793
8.796.2.6 ambr_ul_ext2 . . . 793
8.796.2.7 apnId . . . 793
8.797 satelliteInfo Struct Reference . . . 793
8.797.1 Detailed Description . . . 794
8.797.2 Field Documentation . . . 795
8.797.2.1 azimuth . . . 795
8.797.2.2 elevation . . . 795
8.797.2.3 gnssSvId . . . 795
8.797.2.4 healthStatus . . . 795
8.797.2.5 snr . . . 795
8.797.2.6 svInfoMask . . . 795
8.797.2.7 svListLen . . . 795
8.797.2.8 svStatus . . . 795
8.797.2.9 system . . . 795
8.797.2.10 validMask . . . 795
8.798 SccRxInfo Struct Reference . . . 796
8.798.1 Detailed Description . . . 796

8.798.2 Field Documentation . . . 796
8.798.2.1 numInstances . . . 796
8.798.2.2 rsrq . . . 796
8.798.2.3 sigInfo . . . 796
8.798.2.4 snr . . . 796
8.798.2.5 TlvPresent . . . 796
8.799 sensorData Struct Reference . . . 796
8.799.1 Detailed Description . . . 797
8.799.2 Field Documentation . . . 798
8.799.2.1 flags . . . 798
8.799.2.2 sensorDataLen . . . 798
8.799.2.3 timeOfFirstSample . . . 798
8.799.2.4 timeOffset . . . 798
8.799.2.5 xAxis . . . 798
8.799.2.6 yAxis . . . 798
8.799.2.7 zAxis . . . 798
8.800 sensorData_t Struct Reference . . . 798
8.800.1 Detailed Description . . . 798
8.800.2 Field Documentation . . . 799
8.800.2.1 flags . . . 799
8.800.2.2 sensorDataLen . . . 799
8.800.2.3 timeOfFirstSample . . . 799
8.800.2.4 timeOffset . . . 799
8.800.2.5 xAxis . . . 799
8.800.2.6 yAxis . . . 799
8.800.2.7 zAxis . . . 799
8.801 sensorDataUsage_s Struct Reference . . . 800
8.801.1 Detailed Description . . . 800
8.801.2 Field Documentation . . . 800
8.801.2.1 aidingIndicatorMask . . . 800
8.801.2.2 usageMask . . . 800
8.802 serialNumbersInfo Struct Reference . . . 800
8.802.1 Detailed Description . . . 801
8.802.2 Field Documentation . . . 801
8.802.2.1 esnSize . . . 801
8.802.2.2 imeiSize . . . 801
8.802.2.3 imeiSvnSize . . . 801
8.802.2.4 meidSize . . . 801
8.802.2.5 pESNString . . . 801
8.802.2.6 pIMEIString . . . 801

8.802.2.7 pImeiSvnString . . . 801
8.802.2.8 pMEIDString . . . 801
8.803 serviceProviderName Struct Reference . . . 802
8.803.1 Detailed Description . . . 802
8.803.2 Field Documentation . . . 802
8.803.2.1 displayCondition . . . 802
8.803.2.2 spn . . . 802
8.803.2.3 spnLength . . . 802
8.804 ServingSystemInfo Struct Reference . . . 802
8.804.1 Detailed Description . . . 802
8.804.2 Field Documentation . . . 803
8.804.2.1 csAttachState . . . 803
8.804.2.2 hdrPersonality . . . 803
8.804.2.3 psAttachState . . . 803
8.804.2.4 radioInterfaceList . . . 804
8.804.2.5 radioInterfaceNo . . . 804
8.804.2.6 registrationState . . . 804
8.804.2.7 selectedNetwork . . . 804
8.805 servSystem Struct Reference . . . 804
8.805.1 Detailed Description . . . 804
8.805.2 Field Documentation . . . 805
8.805.2.1 csAttachState . . . 805
8.805.2.2 numRadioInterfaces . . . 805
8.805.2.3 psAttachState . . . 805
8.805.2.4 radioInterface . . . 805
8.805.2.5 regState . . . 805
8.805.2.6 selNetwork . . . 805
8.806 sessionInfo Union Reference . . . 805
8.806.1 Detailed Description . . . 805
8.806.2 Field Documentation . . . 806
8.806.2.1 omaDmConfig . . . 806
8.806.2.2 omaDmFota . . . 806
8.806.2.3 omaDmNotifications . . . 806
8.807 sessionInfoExt Union Reference . . . 806
8.807.1 Detailed Description . . . 806
8.807.2 Field Documentation . . . 806
8.807.2.1 omaDmConfig . . . 806
8.807.2.2 omaDmFota . . . 806
8.808 sessionInfoTlv Struct Reference . . . 806
8.808.1 Detailed Description . . . 806

8.808.2 Field Documentation . . . 806
8.808.2.1 sessionInfo . . . 806
8.808.2.2 sessionType . . . 806
8.808.2.3 TlvPresent . . . 806
8.809 sessionInfoTlvExt Struct Reference . . . 807
8.809.1 Detailed Description . . . 807
8.809.2 Field Documentation . . . 807
8.809.2.1 sessionInfo . . . 807
8.809.2.2 sessionType . . . 807
8.809.2.3 TlvPresent . . . 807
8.810 SetAudioPathConfigReq Struct Reference . . . 807
8.810.1 Detailed Description . . . 807
8.810.2 Field Documentation . . . 808
8.810.2.1 pCodecSTGain . . . 808
8.810.2.2 pDTMFTXGain . . . 809
8.810.2.3 pECMode . . . 809
8.810.2.4 pNSEnable . . . 809
8.810.2.5 Profile . . . 809
8.810.2.6 pRXAGCList . . . 809
8.810.2.7 pRXAVCAGCSwitch . . . 809
8.810.2.8 pRXAVCList . . . 809
8.810.2.9 pRXPCMIIRFltr . . . 809
8.810.2.10 pTXAGCList . . . 809
8.810.2.11 pTXAVCSwitch . . . 809
8.810.2.12 pTXGain . . . 809
8.810.2.13 pTXPCMIIRFltr . . . 809
8.811 SetAudioProfileReq Struct Reference . . . 809
8.811.1 Detailed Description . . . 809
8.811.2 Field Documentation . . . 810
8.811.2.1 EarMute . . . 810
8.811.2.2 Generator . . . 810
8.811.2.3 MicMute . . . 810
8.811.2.4 Profile . . . 810
8.811.2.5 Volume . . . 810
8.812 SetAudioVolTLBConfigReq Struct Reference . . . 810
8.812.1 Detailed Description . . . 810
8.812.2 Field Documentation . . . 811
8.812.2.1 Generator . . . 811
8.812.2.2 Item . . . 811
8.812.2.3 Profile . . . 811

8.812.2.4 Volume . . . 811
8.812.2.5 VolValue . . . 811
8.813 SetAudioVolTLBConfigResp Struct Reference . . . 811
8.813.1 Detailed Description . . . 811
8.813.2 Field Documentation . . . 811
8.813.2.1 ResCode . . . 811
8.814 setCustomSettingV2 Struct Reference . . . 812
8.814.1 Detailed Description . . . 812
8.814.2 Field Documentation . . . 812
8.814.2.1 cust_id . . . 812
8.814.2.2 cust_value . . . 812
8.814.2.3 value_length . . . 812
8.815 setDyingGaspCfg Struct Reference . . . 812
8.815.1 Detailed Description . . . 812
8.815.2 Field Documentation . . . 813
8.815.2.1 pDestSMSContent . . . 813
8.815.2.2 pDestSMSNum . . . 813
8.816 SetIMSSMSConfigReq Struct Reference . . . 813
8.816.1 Detailed Description . . . 813
8.816.2 Field Documentation . . . 813
8.816.2.1 pPhoneCtxtURI . . . 813
8.816.2.2 pPhoneCtxtURILen . . . 813
8.816.2.3 pSMSFormat . . . 813
8.816.2.4 pSMSOverIPNwInd . . . 813
8.817 SetIMSSMSConfigResp Struct Reference . . . 813
8.817.1 Detailed Description . . . 814
8.817.2 Field Documentation . . . 814
8.817.2.1 pSettingResp . . . 814
8.818 SetIMSUserConfigReq Struct Reference . . . 814
8.818.1 Detailed Description . . . 814
8.818.2 Field Documentation . . . 814
8.818.2.1 pIMSDomain . . . 814
8.818.2.2 pIMSDomainLen . . . 814
8.819 SetIMSUserConfigResp Struct Reference . . . 814
8.819.1 Detailed Description . . . 815
8.819.2 Field Documentation . . . 815
8.819.2.1 pSettingResp . . . 815
8.820 SetIMSVoIPConfigReq Struct Reference . . . 815
8.820.1 Detailed Description . . . 815
8.820.2 Field Documentation . . . 817

8.820.2.1 pAmrMode . . . 817
8.820.2.2 pAmrOctetAligned . . . 817
8.820.2.3 pAmrWbEnable . . . 817
8.820.2.4 pAmrWBMode . . . 817
8.820.2.5 pAmrWBOctetAligned . . . 817
8.820.2.6 pMinSessionExpiryTimer . . . 817
8.820.2.7 pRingBackTimer . . . 817
8.820.2.8 pRingingTimer . . . 817
8.820.2.9 pRTPRTCPInactTimer . . . 817
8.820.2.10 pScrAmrEnable . . . 817
8.820.2.11 pScrAmrWbEnable . . . 817
8.820.2.12 pSessionExpiryTimer . . . 817
8.821 SetIMSVoIPConfigResp Struct Reference . . . 817
8.821.1 Detailed Description . . . 817
8.821.2 Field Documentation . . . 818
8.821.2.1 pSettingResp . . . 818
8.822 SetM2MAudioAVCFGReq Struct Reference . . . 818
8.822.1 Detailed Description . . . 818
8.822.2 Field Documentation . . . 818
8.822.2.1 Device . . . 818
8.822.2.2 PIFACEId . . . 818
8.822.2.3 pPCMParams . . . 818
8.822.2.4 Profile . . . 818
8.823 SetM2MAudioLPBKReq Struct Reference . . . 819
8.823.1 Detailed Description . . . 819
8.823.2 Field Documentation . . . 819
8.823.2.1 Enable . . . 819
8.824 SetM2MAudioProfileReq Struct Reference . . . 819
8.824.1 Detailed Description . . . 819
8.824.2 Field Documentation . . . 820
8.824.2.1 pCwtMute . . . 820
8.824.2.2 pEarMute . . . 820
8.824.2.3 pGenerator . . . 820
8.824.2.4 pMicMute . . . 820
8.824.2.5 Profile . . . 820
8.824.2.6 pVolume . . . 820
8.825 SetM2MAudioVolumeReq Struct Reference . . . 820
8.825.1 Detailed Description . . . 820
8.825.2 Field Documentation . . . 821
8.825.2.1 Generator . . . 821

8.825.2.2 Level . . . 821
8.825.2.3 Profile . . . 821
8.826 SetM2MAVMuteReq Struct Reference . . . 821
8.826.1 Detailed Description . . . 821
8.826.2 Field Documentation . . . 821
8.826.2.1 EarMute . . . 821
8.826.2.2 MicMute . . . 821
8.826.2.3 pCwtMute . . . 822
8.826.2.4 Profile . . . 822
8.827 SetM2MSpkrGainReq Struct Reference . . . 822
8.827.1 Detailed Description . . . 822
8.827.2 Field Documentation . . . 822
8.827.2.1 Profile . . . 822
8.827.2.2 Value . . . 822
8.828 setPINProtection Struct Reference . . . 822
8.828.1 Detailed Description . . . 822
8.828.2 Field Documentation . . . 823
8.828.2.1 pinID . . . 823
8.828.2.2 pinLength . . . 823
8.828.2.3 pinOperation . . . 823
8.828.2.4 pinValue . . . 823
8.829 SetRegMgrConfigReq Struct Reference . . . 823
8.829.1 Detailed Description . . . 823
8.829.2 Field Documentation . . . 824
8.829.2.1 pCSCFPortName . . . 824
8.829.2.2 pCSCFPortNameLen . . . 824
8.829.2.3 pIMSTestMode . . . 824
8.829.2.4 pPriCSCFPort . . . 824
8.830 SetRegMgrConfigResp Struct Reference . . . 824
8.830.1 Detailed Description . . . 824
8.830.2 Field Documentation . . . 824
8.830.2.1 pSettingResp . . . 824
8.831 setSignalStrengthInfo Struct Reference . . . 824
8.831.1 Detailed Description . . . 825
8.831.2 Field Documentation . . . 828
8.831.2.1 pCDMAECIODelta . . . 828
8.831.2.2 pCDMAECIOThresh . . . 828
8.831.2.3 pCDMARSSIDelta . . . 828
8.831.2.4 pCDMARSSIThresh . . . 828
8.831.2.5 pGSMRSSIDelta . . . 828

8.831.2.6 pGSMRSSIThresh . . . 828
8.831.2.7 pHDRECIODelta . . . 828
8.831.2.8 pHDRECIOThresh . . . 828
8.831.2.9 pHDRIODelta . . . 828
8.831.2.10 pHDRIOThresh . . . 828
8.831.2.11 pHDRRSSIDelta . . . 828
8.831.2.12 pHDRRSSIThresh . . . 828
8.831.2.13 pHDRSINRDelta . . . 828
8.831.2.14 pHDRSINRThresh . . . 828
8.831.2.15 pLTERSRPDelta . . . 828
8.831.2.16 pLTERSRPThresh . . . 828
8.831.2.17 pLTERSRQDelta . . . 828
8.831.2.18 pLTERSRQThresh . . . 829
8.831.2.19 pLTERSSIDelta . . . 829
8.831.2.20 pLTERSSIThresh . . . 829
8.831.2.21 pLTESigRptConfig . . . 829
8.831.2.22 pLTESNRDelta . . . 829
8.831.2.23 pLTESNRThresh . . . 829
8.831.2.24 pTDSCDMAECIODelta . . . 829
8.831.2.25 pTDSCDMAECIOThresh . . . 829
8.831.2.26 pTDSCDMARSCPDelta . . . 829
8.831.2.27 pTDSCDMARSCPThresh . . . 829
8.831.2.28 pTDSCDMARSSIDelta . . . 829
8.831.2.29 pTDSCDMARSSIThresh . . . 829
8.831.2.30 pTDSCDMASINRDelta . . . 829
8.831.2.31 pTDSCDMASINRThresh . . . 829
8.831.2.32 pWCDMAECIODelta . . . 829
8.831.2.33 pWCDMAECIOThresh . . . 829
8.831.2.34 pWCDMARSSIDelta . . . 829
8.831.2.35 pWCDMARSSIThresh . . . 829
8.832 SetSIPConfigReq Struct Reference . . . 829
8.832.1 Detailed Description . . . 829
8.832.2 Field Documentation . . . 830
8.832.2.1 pSigCompEnabled . . . 830
8.832.2.2 pSIPLocalPort . . . 830
8.832.2.3 pSubscribeTimer . . . 830
8.832.2.4 pTimerSIPReg . . . 830
8.832.2.5 pTimerT1 . . . 830
8.832.2.6 pTimerT2 . . . 830
8.832.2.7 pTimerTf . . . 830

8.833 SetSIPConfigResp Struct Reference . . . 830
8.833.1 Detailed Description . . . 830
8.833.2 Field Documentation . . . 831
8.833.2.1 pSettingResp . . . 831
8.834 sGetDeviceSeriesResult Struct Reference . . . 831
8.834.1 Detailed Description . . . 831
8.834.2 Field Documentation . . . 831
8.834.2.1 eDevice . . . 831
8.834.2.2 uResult . . . 831
8.835 sidNid Struct Reference . . . 831
8.835.1 Detailed Description . . . 831
8.835.2 Field Documentation . . . 832
8.835.2.1 nid . . . 832
8.835.2.2 sid . . . 832
8.836 sigInfo Struct Reference . . . 832
8.836.1 Detailed Description . . . 832
8.836.2 Field Documentation . . . 833
8.836.2.1 pECIOThresh . . . 833
8.836.2.2 pHDRSINRThresh . . . 833
8.836.2.3 pIOThresh . . . 833
8.836.2.4 pLTESigRptCfg . . . 833
8.836.2.5 pLTESNRThresh . . . 833
8.836.2.6 pRSRPThresh . . . 833
8.836.2.7 pRSRQThresh . . . 833
8.836.2.8 pRSSIThresh . . . 833
8.836.2.9 pTDSCDMASINRCONFThresh . . . 833
8.837 signalInfo Struct Reference . . . 833
8.837.1 Detailed Description . . . 833
8.837.2 Field Documentation . . . 834
8.837.2.1 alertPitch . . . 834
8.837.2.2 signal . . . 834
8.837.2.3 signalType . . . 834
8.838 SignalStrengthDataType Struct Reference . . . 834
8.838.1 Field Documentation . . . 834
8.838.1.1 thresholds . . . 834
8.838.1.2 thresholdsSize . . . 834
8.839 slot_t Struct Reference . . . 834
8.839.1 Detailed Description . . . 834
8.839.2 Field Documentation . . . 835
8.839.2.1 bICCID . . . 835

8.839.2.2 bICCIDLength . . . 835
8.839.2.3 bLogicalSlot . . . 835
8.839.2.4 uPhyCardStatus . . . 835
8.839.2.5 uPhySlotStatus . . . 835
8.840 slotInf Struct Reference . . . 835
8.840.1 Detailed Description . . . 835
8.840.2 Field Documentation . . . 836
8.840.2.1 AppStatus . . . 836
8.840.2.2 cardState . . . 836
8.840.2.3 errorState . . . 836
8.840.2.4 numApp . . . 836
8.840.2.5 upinRetries . . . 837
8.840.2.6 upinState . . . 837
8.840.2.7 upukRetries . . . 837
8.841 slotInfo Struct Reference . . . 837
8.841.1 Detailed Description . . . 837
8.841.2 Field Documentation . . . 838
8.841.2.1 AppStatus . . . 838
8.841.2.2 cardState . . . 838
8.841.2.3 errorState . . . 838
8.841.2.4 numApp . . . 838
8.841.2.5 upinRetries . . . 838
8.841.2.6 upinState . . . 838
8.841.2.7 upukRetries . . . 838
8.842 slots_t Struct Reference . . . 838
8.842.1 Field Documentation . . . 838
8.842.1.1 uimSlotStatus . . . 838
8.843 slqsautoconnect Struct Reference . . . 838
8.843.1 Detailed Description . . . 839
8.843.2 Field Documentation . . . 839
8.843.2.1 acroamsetting . . . 839
8.843.2.2 acsetting . . . 839
8.843.2.3 action . . . 839
8.844 SLQSDeleteProfileParams Struct Reference . . . 839
8.844.1 Detailed Description . . . 839
8.844.2 Field Documentation . . . 840
8.844.2.1 profileIndex . . . 840
8.844.2.2 profileType . . . 840
8.845 slqsfwinfo_s Struct Reference . . . 840
8.845.1 Detailed Description . . . 840

8.845.2 Field Documentation . . . 841
8.845.2.1 appversion_str . . . 841
8.845.2.2 bootversion_str . . . 841
8.845.2.3 carrier_str . . . 841
8.845.2.4 cur_carr_name . . . 841
8.845.2.5 cur_carr_rev . . . 841
8.845.2.6 modelid_str . . . 841
8.845.2.7 packageid_str . . . 841
8.845.2.8 priversion_str . . . 841
8.845.2.9 sku_str . . . 841
8.846 SlqsNas3GppNetworkInfo Struct Reference . . . 841
8.846.1 Detailed Description . . . 841
8.846.2 Field Documentation . . . 842
8.846.2.1 Description . . . 842
8.846.2.2 Forbidden . . . 842
8.846.2.3 InUse . . . 842
8.846.2.4 MCC . . . 842
8.846.2.5 MNC . . . 842
8.846.2.6 Preferred . . . 842
8.846.2.7 Roaming . . . 842
8.847 SlqsNasPcsDigit Struct Reference . . . 842
8.847.1 Detailed Description . . . 843
8.847.2 Field Documentation . . . 843
8.847.2.1 includes_pcs_digit . . . 843
8.847.2.2 MCC . . . 843
8.847.2.3 MNC . . . 843
8.848 slqssendasyncsmsparams_s Struct Reference . . . 843
8.848.1 Detailed Description . . . 843
8.848.2 Field Documentation . . . 845
8.848.2.1 messageFormat . . . 845
8.848.2.2 messageSize . . . 845
8.848.2.3 pFollowOnDC . . . 845
8.848.2.4 pForceOnDC . . . 845
8.848.2.5 pLinktimer . . . 845
8.848.2.6 pMessage . . . 845
8.848.2.7 pRetryMessage . . . 845
8.848.2.8 pRetryMessageId . . . 845
8.848.2.9 pServiceOption . . . 845
8.848.2.10 pSmsOnIms . . . 845
8.848.2.11 pUserData . . . 845

8.849 slqssendsmsparams_s Struct Reference . . . 845
8.849.1 Detailed Description . . . 845
8.849.2 Field Documentation . . . 846
8.849.2.1 messageFailureCode . . . 846
8.849.2.2 messageFormat . . . 846
8.849.2.3 messageID . . . 846
8.849.2.4 messageSize . . . 846
8.849.2.5 pLinktimer . . . 846
8.849.2.6 pMessage . . . 846
8.849.2.7 pSmsOnIms . . . 846
8.850 slqsSessionStateInfo Struct Reference . . . 846
8.850.1 Detailed Description . . . 847
8.850.2 Field Documentation . . . 847
8.850.2.1 pQmiInterfaceInfo . . . 847
8.850.2.2 reconfiguration_required . . . 847
8.850.2.3 sessionEndReason . . . 847
8.850.2.4 state . . . 847
8.851 slqsSignalStrengthInfo Struct Reference . . . 847
8.851.1 Detailed Description . . . 848
8.851.2 Field Documentation . . . 849
8.851.2.1 ecioList . . . 849
8.851.2.2 ecioListLen . . . 849
8.851.2.3 errorRateList . . . 849
8.851.2.4 errorRateListLen . . . 849
8.851.2.5 Io . . . 849
8.851.2.6 ltersrp . . . 849
8.851.2.7 ltesnr . . . 849
8.851.2.8 rsrqInfo . . . 849
8.851.2.9 rxSignalStrengthList . . . 849
8.851.2.10 rxSignalStrengthListLen . . . 849
8.851.2.11 signalStrengthReqMask . . . 850
8.851.2.12 sinr . . . 850
8.852 SLQSSignalStrengthsIndReq Struct Reference . . . 850
8.852.1 Detailed Description . . . 850
8.852.2 Field Documentation . . . 851
8.852.2.1 ecioDelta . . . 851
8.852.2.2 ecioThresholdList . . . 851
8.852.2.3 ecioThresholdListLen . . . 851
8.852.2.4 ioDelta . . . 851
8.852.2.5 lteRsrpDelta . . . 851

8.852.2.6 lteSnrDelta . . . 851
8.852.2.7 rsrqDelta . . . 851
8.852.2.8 rxSignalStrengthDelta . . . 851
8.852.2.9 sinrDelta . . . 851
8.852.2.10 sinrThresholdList . . . 851
8.852.2.11 sinrThresholdListLen . . . 851
8.853 SLQSSignalStrengthsInformation Struct Reference . . . 852
8.853.1 Detailed Description . . . 852
8.853.2 Field Documentation . . . 852
8.853.2.1 ecioInfo . . . 853
8.853.2.2 errorRateInfo . . . 853
8.853.2.3 io . . . 853
8.853.2.4 lteRsrpinfo . . . 853
8.853.2.5 lteSnrinfo . . . 853
8.853.2.6 rsrqInfo . . . 853
8.853.2.7 rxSignalStrengthInfo . . . 853
8.853.2.8 sinr . . . 853
8.854 slqsWdsEventInfo Struct Reference . . . 853
8.854.1 Detailed Description . . . 853
8.854.2 Field Documentation . . . 854
8.854.2.1 pDataBearer . . . 854
8.854.2.2 pDormancyStatus . . . 854
8.854.2.3 pPacketsCountRX . . . 854
8.854.2.4 pPacketsCountTX . . . 854
8.854.2.5 pQmiInterfaceInfo . . . 855
8.854.2.6 pTotalBytesRX . . . 855
8.854.2.7 pTotalBytesTX . . . 855
8.855 sms_BroadcastConfig Struct Reference . . . 855
8.855.1 Detailed Description . . . 855
8.855.2 Field Documentation . . . 855
8.855.2.1 fromServiceId . . . 855
8.855.2.2 selected . . . 855
8.855.2.3 toServiceId . . . 855
8.856 sms_CDMABroadcastConfig Struct Reference . . . 855
8.856.1 Detailed Description . . . 855
8.856.2 Field Documentation . . . 856
8.856.2.1 language . . . 856
8.856.2.2 selected . . . 856
8.856.2.3 serviceCategory . . . 856
8.857 sms_getIndicationReg Struct Reference . . . 856

8.857.1 Detailed Description . . . 856
8.857.2 Field Documentation . . . 857
8.857.2.1 pRegCallStatInfoEvt . . . 857
8.857.2.2 pRegTransLayerInfoEvt . . . 857
8.857.2.3 pRegTransNWRegInfoEvt . . . 857
8.858 sms_getMsgWaitingInfo Struct Reference . . . 857
8.858.1 Detailed Description . . . 857
8.858.2 Field Documentation . . . 857
8.858.2.1 msgWaitInfo . . . 857
8.858.2.2 numInstances . . . 857
8.859 sms_getTransLayerInfo Struct Reference . . . 857
8.859.1 Detailed Description . . . 858
8.859.2 Field Documentation . . . 858
8.859.2.1 pRegInd . . . 858
8.859.2.2 pTransLayerInfo . . . 858
8.860 sms_getTransNWRegInfo Struct Reference . . . 858
8.860.1 Detailed Description . . . 858
8.860.2 Field Documentation . . . 858
8.860.2.1 pRegStatus . . . 859
8.861 sms_maxStorageSizeReq Struct Reference . . . 859
8.861.1 Detailed Description . . . 859
8.861.2 Field Documentation . . . 859
8.861.2.1 pMessageMode . . . 859
8.861.2.2 storageType . . . 859
8.862 sms_maxStorageSizeResp Struct Reference . . . 859
8.862.1 Detailed Description . . . 859
8.862.2 Field Documentation . . . 860
8.862.2.1 freeSlots . . . 860
8.862.2.2 maxStorageSize . . . 860
8.863 sms_messageWaitingInfoContent Struct Reference . . . 860
8.863.1 Detailed Description . . . 860
8.863.2 Field Documentation . . . 860
8.863.2.1 activeInd . . . 860
8.863.2.2 msgCount . . . 860
8.863.2.3 msgType . . . 860
8.864 sms_msgProtocolResp Struct Reference . . . 861
8.864.1 Detailed Description . . . 861
8.864.2 Field Documentation . . . 861
8.864.2.1 msgProtocol . . . 861
8.865 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . 861

8.865.1 Detailed Description . . . 861
8.865.2 Field Documentation . . . 862
8.865.2.1 activated_ind . . . 862
8.865.2.2 CDMABroadcastConfig . . . 862
8.865.2.3 num_instances . . . 862
8.866 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference . . . 862
8.866.1 Detailed Description . . . 862
8.866.2 Field Documentation . . . 862
8.866.2.1 activated_ind . . . 862
8.866.2.2 broadcastConfig . . . 862
8.866.2.3 num_instances . . . 862
8.867 sms_routeEntry Struct Reference . . . 862
8.867.1 Detailed Description . . . 863
8.867.2 Field Documentation . . . 863
8.867.2.1 messageClass . . . 863
8.867.2.2 messageType . . . 863
8.867.2.3 receiptAction . . . 863
8.867.2.4 routeStorage . . . 863
8.868 sms_sendAsyncsmsParams Struct Reference . . . 863
8.868.1 Detailed Description . . . 864
8.868.2 Field Documentation . . . 865
8.868.2.1 messageFormat . . . 865
8.868.2.2 messageSize . . . 865
8.868.2.3 pFollowOnDC . . . 865
8.868.2.4 pForceOnDC . . . 865
8.868.2.5 pLinktimer . . . 865
8.868.2.6 pMessage . . . 865
8.868.2.7 pRetryMessage . . . 865
8.868.2.8 pRetryMessageId . . . 865
8.868.2.9 pServiceOption . . . 865
8.868.2.10 pSmsOnIms . . . 865
8.868.2.11 pUserData . . . 865
8.869 sms_setIndicationReg Struct Reference . . . 866
8.869.1 Detailed Description . . . 866
8.869.2 Field Documentation . . . 866
8.869.2.1 pRegCallStatInfoEvt . . . 866
8.869.2.2 pRegTransLayerInfoEvt . . . 866
8.869.2.3 pRegTransNWRegInfoEvt . . . 866
8.870 sms_setRoutesReq Struct Reference . . . 866
8.870.1 Detailed Description . . . 867

8.870.2 Field Documentation . . . 867
8.870.2.1 numOfRoutes . . . 867
8.870.2.2 pTransferStatusReport . . . 867
8.870.2.3 routeList . . . 867
8.871 sms_transLayerInfo Struct Reference . . . 867
8.871.1 Detailed Description . . . 867
8.871.2 Field Documentation . . . 867
8.871.2.1 TransCap . . . 868
8.871.2.2 TransType . . . 868
8.872 SMSAsyncRawSend_s Struct Reference . . . 868
8.872.1 Detailed Description . . . 868
8.872.2 Field Documentation . . . 869
8.872.2.1 alphaIDLen . . . 869
8.872.2.2 causeCode . . . 869
8.872.2.3 errorClass . . . 869
8.872.2.4 messageID . . . 869
8.872.2.5 msgDelFailureCause . . . 869
8.872.2.6 msgDelFailureType . . . 869
8.872.2.7 pAlphaID . . . 869
8.872.2.8 RPCause . . . 869
8.872.2.9 sendStatus . . . 869
8.872.2.10 TPCause . . . 869
8.872.2.11 userData . . . 869
8.873 sMSCAddress Struct Reference . . . 869
8.873.1 Detailed Description . . . 870
8.873.2 Field Documentation . . . 870
8.873.2.1 data . . . 870
8.873.2.2 length . . . 870
8.874 SMSCAddress Struct Reference . . . 870
8.874.1 Detailed Description . . . 870
8.874.2 Field Documentation . . . 870
8.874.2.1 data . . . 870
8.874.2.2 length . . . 870
8.875 sMSCAddressTlv Struct Reference . . . 870
8.875.1 Detailed Description . . . 870
8.875.2 Field Documentation . . . 871
8.875.2.1 SMSCInfo . . . 871
8.875.2.2 TlvPresent . . . 871
8.876 sMSEtwsMessage Struct Reference . . . 871
8.876.1 Detailed Description . . . 871

8.876.2 Field Documentation . . . 871
8.876.2.1 data . . . 871
8.876.2.2 length . . . 871
8.876.2.3 notificationType . . . 871
8.877 SMSEtwsMessage Struct Reference . . . 871
8.877.1 Detailed Description . . . 872
8.877.2 Field Documentation . . . 872
8.877.2.1 data . . . 872
8.877.2.2 length . . . 872
8.877.2.3 notificationType . . . 872
8.878 sMSEtwsMessageTlv Struct Reference . . . 872
8.878.1 Detailed Description . . . 872
8.878.2 Field Documentation . . . 872
8.878.2.1 EtwsMessageInfo . . . 872
8.878.2.2 TlvPresent . . . 872
8.879 SMSEtwsPlmn Struct Reference . . . 872
8.879.1 Detailed Description . . . 873
8.879.2 Field Documentation . . . 873
8.879.2.1 mobileCountryCode . . . 873
8.879.2.2 mobileNetworkCode . . . 873
8.880 sMSEtwsPlmn Struct Reference . . . 873
8.880.1 Detailed Description . . . 873
8.880.2 Field Documentation . . . 873
8.880.2.1 mobileCountryCode . . . 873
8.880.2.2 mobileNetworkCode . . . 873
8.881 SMSEventInfo_s Struct Reference . . . 873
8.881.1 Detailed Description . . . 874
8.881.2 Field Documentation . . . 874
8.881.2.1 pEtwsMessageInfo . . . 874
8.881.2.2 pEtwsPlmnInfo . . . 874
8.881.2.3 pMessageModeInfo . . . 875
8.881.2.4 pMTMessageInfo . . . 875
8.881.2.5 pSMSCAddressInfo . . . 875
8.881.2.6 pSMSOnIMSInfo . . . 875
8.881.2.7 pTransferRouteMTMessageInfo . . . 875
8.881.2.8 smsEventType . . . 875
8.882 smsMaxStorageSizeReq Struct Reference . . . 875
8.882.1 Detailed Description . . . 875
8.882.2 Field Documentation . . . 875
8.882.2.1 pMessageMode . . . 875

8.882.2.2 storageType . . . 875
8.883 smsMaxStorageSizeResp Struct Reference . . . 875
8.883.1 Detailed Description . . . 876
8.883.2 Field Documentation . . . 876
8.883.2.1 freeSlots . . . 876
8.883.2.2 maxStorageSize . . . 876
8.884 SMSMemoryInfo Struct Reference . . . 876
8.884.1 Detailed Description . . . 876
8.884.2 Field Documentation . . . 876
8.884.2.1 messageMode . . . 876
8.884.2.2 storageType . . . 876
8.885 SMSMessageMode Struct Reference . . . 876
8.885.1 Detailed Description . . . 877
8.885.2 Field Documentation . . . 877
8.885.2.1 messageMode . . . 877
8.886 sMSMessageMode Struct Reference . . . 877
8.886.1 Detailed Description . . . 877
8.886.2 Field Documentation . . . 877
8.886.2.1 messageMode . . . 877
8.887 smsMsgprotocolResp Struct Reference . . . 877
8.887.1 Detailed Description . . . 877
8.887.2 Field Documentation . . . 878
8.887.2.1 msgProtocol . . . 878
8.888 SMSMTMessage Struct Reference . . . 878
8.888.1 Detailed Description . . . 878
8.888.2 Field Documentation . . . 878
8.888.2.1 messageIndex . . . 878
8.888.2.2 storageType . . . 878
8.889 sMSMTMessage Struct Reference . . . 878
8.889.1 Detailed Description . . . 878
8.889.2 Field Documentation . . . 878
8.889.2.1 messageIndex . . . 878
8.889.2.2 storageType . . . 878
8.890 SMSOnIMS Struct Reference . . . 879
8.890.1 Detailed Description . . . 879
8.890.2 Field Documentation . . . 879
8.890.2.1 smsOnIMS . . . 879
8.891 sMSOnIMS Struct Reference . . . 879
8.891.1 Detailed Description . . . 879
8.891.2 Field Documentation . . . 879

8.891.2.1 smsOnIMS . . . 879
8.892 sMSOnIMSTlv Struct Reference . . . 879
8.892.1 Detailed Description . . . 879
8.892.2 Field Documentation . . . 880
8.892.2.1 IMSInfo . . . 880
8.892.2.2 TlvPresent . . . 880
8.893 smsRouteEntry Struct Reference . . . 880
8.893.1 Detailed Description . . . 880
8.893.2 Field Documentation . . . 881
8.893.2.1 messageClass . . . 881
8.893.2.2 messageType . . . 881
8.893.2.3 receiptAction . . . 881
8.893.2.4 routeStorage . . . 881
8.894 smsSetRoutesReq Struct Reference . . . 881
8.894.1 Detailed Description . . . 881
8.894.2 Field Documentation . . . 881
8.894.2.1 numOfRoutes . . . 881
8.894.2.2 pTransferStatusReport . . . 881
8.894.2.3 routeList . . . 881
8.895 SMSTransferRouteMTMessage Struct Reference . . . 882
8.895.1 Detailed Description . . . 882
8.895.2 Field Documentation . . . 882
8.895.2.1 ackIndicator . . . 882
8.895.2.2 data . . . 882
8.895.2.3 format . . . 882
8.895.2.4 length . . . 882
8.895.2.5 transactionID . . . 882
8.896 sMSTransferRouteMTMessage Struct Reference . . . 882
8.896.1 Detailed Description . . . 883
8.896.2 Field Documentation . . . 883
8.896.2.1 ackIndicator . . . 883
8.896.2.2 data . . . 883
8.896.2.3 format . . . 883
8.896.2.4 length . . . 883
8.896.2.5 transactionID . . . 883
8.897 sQosFlowStat Struct Reference . . . 883
8.897.1 Detailed Description . . . 883
8.897.2 Field Documentation . . . 884
8.897.2.1 bearerId . . . 884
8.897.2.2 tx_bytes . . . 884

8.897.2.3 tx_bytes_drp . . . 884
8.897.2.4 tx_pkt . . . 884
8.897.2.5 tx_pkt_drp . . . 884
8.898 sQosStat Struct Reference . . . 884
8.898.1 Detailed Description . . . 884
8.898.2 Field Documentation . . . 885
8.898.2.1 apnId . . . 885
8.898.2.2 numQosFlow . . . 885
8.898.2.3 qosFlow . . . 885
8.898.2.4 total_rx_bytes . . . 885
8.898.2.5 total_rx_pkt . . . 885
8.898.2.6 total_tx_bytes . . . 885
8.898.2.7 total_tx_bytes_drp . . . 885
8.898.2.8 total_tx_pkt . . . 885
8.898.2.9 total_tx_pkt_drp . . . 885
8.899 SrvStatusInfo Struct Reference . . . 885
8.899.1 Detailed Description . . . 886
8.899.2 Field Documentation . . . 886
8.899.2.1 isPrefDataPath . . . 886
8.899.2.2 srvStatus . . . 886
8.900 ssdatasession_params Struct Reference . . . 886
8.900.1 Detailed Description . . . 887
8.900.2 Field Documentation . . . 888
8.900.2.1 action . . . 888
8.900.2.2 failureReason . . . 888
8.900.2.3 failureReasonv4 . . . 888
8.900.2.4 failureReasonv6 . . . 888
8.900.2.5 instanceId . . . 888
8.900.2.6 ipfamily . . . 888
8.900.2.7 pAuthentication . . . 888
8.900.2.8 pPassword . . . 888
8.900.2.9 pProfileId3GPP . . . 888
8.900.2.10 pProfileId3GPP2 . . . 888
8.900.2.11 pTechnology . . . 889
8.900.2.12 pUsername . . . 889
8.900.2.13 rcv4 . . . 889
8.900.2.14 rcv6 . . . 889
8.900.2.15 sessionId . . . 889
8.900.2.16 v4sessionId . . . 889
8.900.2.17 v6sessionId . . . 889

8.900.2.18 verbFailReason . . . 889
8.900.2.19 verbFailReasonType . . . 889
8.901 SupportedMsgList Struct Reference . . . 889
8.901.1 Detailed Description . . . 889
8.901.2 Field Documentation . . . 889
8.901.2.1 supportedMsgLen . . . 889
8.901.2.2 supportedMsgs . . . 889
8.902 SUPSInfo Struct Reference . . . 889
8.902.1 Detailed Description . . . 890
8.902.2 Field Documentation . . . 890
8.902.2.1 isModByCC . . . 890
8.902.2.2 svcType . . . 890
8.903 SV Struct Reference . . . 890
8.903.1 Detailed Description . . . 890
8.903.2 Field Documentation . . . 891
8.903.2.1 id . . . 891
8.903.2.2 mask . . . 891
8.903.2.3 system . . . 891
8.904 SVInfo Struct Reference . . . 891
8.904.1 Detailed Description . . . 891
8.904.2 Field Documentation . . . 891
8.904.2.1 len . . . 892
8.904.2.2 pSV . . . 892
8.905 svUsedforFix_s Struct Reference . . . 892
8.905.1 Detailed Description . . . 892
8.905.2 Field Documentation . . . 892
8.905.2.1 gnssSvUsedList . . . 892
8.905.2.2 gnssSvUsedList_len . . . 892
8.906 SWI_STRUCT_CarrierImage Struct Reference . . . 892
8.906.1 Detailed Description . . . 892
8.906.2 Field Documentation . . . 893
8.906.2.1 m_FwBuildId . . . 893
8.906.2.2 m_FwImageId . . . 893
8.906.2.3 m_nCarrierId . . . 893
8.906.2.4 m_nFolderId . . . 893
8.906.2.5 m_nStorage . . . 893
8.906.2.6 m_PriBuildId . . . 893
8.906.2.7 m_PriImageId . . . 893
8.907 SwiLocGetAutoStartResp Struct Reference . . . 893
8.907.1 Detailed Description . . . 894

8.907.2 Field Documentation . . . 895
8.907.2.1 fix_rate . . . 895
8.907.2.2 fix_rate_reported . . . 895
8.907.2.3 fix_type . . . 895
8.907.2.4 fix_type_reported . . . 895
8.907.2.5 function . . . 895
8.907.2.6 function_reported . . . 895
8.907.2.7 max_dist . . . 895
8.907.2.8 max_dist_reported . . . 895
8.907.2.9 max_time . . . 895
8.907.2.10 max_time_reported . . . 895
8.908 SwiLocSetAutoStartReq Struct Reference . . . 895
8.908.1 Detailed Description . . . 895
8.908.2 Field Documentation . . . 896
8.908.2.1 fix_rate . . . 896
8.908.2.2 fix_type . . . 896
8.908.2.3 function . . . 896
8.908.2.4 max_dist . . . 896
8.908.2.5 max_time . . . 896
8.908.2.6 set_fix_rate . . . 897
8.908.2.7 set_fix_type . . . 897
8.908.2.8 set_function . . . 897
8.908.2.9 set_max_dist . . . 897
8.908.2.10 set_max_time . . . 897
8.909 swiModemStatusResp Struct Reference . . . 897
8.909.1 Detailed Description . . . 897
8.909.2 Field Documentation . . . 897
8.909.2.1 commonInfo . . . 897
8.909.2.2 pLTEInfo . . . 897
8.910 SwiOTAMsg_s Struct Reference . . . 897
8.910.1 Detailed Description . . . 897
8.910.2 Field Documentation . . . 898
8.910.2.1 data . . . 898
8.910.2.2 data_len . . . 898
8.910.2.3 pLteNasRelInfo . . . 898
8.910.2.4 pTime . . . 898
8.910.2.5 type . . . 898
8.911 swiPDPRuntimeSettingsReq Struct Reference . . . 898
8.911.1 Detailed Description . . . 898
8.911.2 Field Documentation . . . 899

8.911.2.1 contextId . . . 899
8.911.2.2 contextType . . . 899
8.912 swiPDPRuntimeSettingsResp Struct Reference . . . 899
8.912.1 Detailed Description . . . 899
8.912.2 Field Documentation . . . 901
8.912.2.1 pAPNName . . . 901
8.912.2.2 pBearerId . . . 901
8.912.2.3 pContextId . . . 901
8.912.2.4 pIPv4Address . . . 901
8.912.2.5 pIPv4GWAddress . . . 901
8.912.2.6 pIPv6Address . . . 901
8.912.2.7 pIPv6GWAddress . . . 901
8.912.2.8 pPrDNSIPv4Address . . . 901
8.912.2.9 pPrDNSIPv6Address . . . 901
8.912.2.10 pPrPCSCFIPv4Address . . . 901
8.912.2.11 pPrPCSCFIPv6Address . . . 901
8.912.2.12 pSeDNSIPv4Address . . . 901
8.912.2.13 pSeDNSIPv6Address . . . 901
8.912.2.14 pSePCSCFIPv4Address . . . 901
8.912.2.15 pSePCSCFIPv6Address . . . 901
8.913 swiQosFilter Struct Reference . . . 901
8.913.1 Detailed Description . . . 902
8.913.2 Field Documentation . . . 903
8.913.2.1 index . . . 903
8.913.2.2 pEspSpi . . . 903
8.913.2.3 pId . . . 903
8.913.2.4 pIPv4DstAddr . . . 903
8.913.2.5 pIPv4SrcAddr . . . 903
8.913.2.6 pIPv6DstAddr . . . 903
8.913.2.7 pIPv6Label . . . 903
8.913.2.8 pIPv6SrcAddr . . . 903
8.913.2.9 pIPv6TrafCls . . . 903
8.913.2.10 pNxtHdrProto . . . 903
8.913.2.11 pPrecedence . . . 903
8.913.2.12 pTCPDstPort . . . 903
8.913.2.13 pTCPSrcPort . . . 903
8.913.2.14 pTos . . . 903
8.913.2.15 pTranDstPort . . . 903
8.913.2.16 pTranSrcPort . . . 903
8.913.2.17 pUDPDstPort . . . 903

8.913.2.18 pUDPSrcPort . . . 904
8.913.2.19 version . . . 904
8.914 swiQosFlow Struct Reference . . . 904
8.914.1 Detailed Description . . . 904
8.914.2 Field Documentation . . . 906
8.914.2.1 index . . . 906
8.914.2.2 p3GPP2Pri . . . 906
8.914.2.3 p3GPPImCn . . . 906
8.914.2.4 p3GPPResResidualBER . . . 906
8.914.2.5 p3GPPSigInd . . . 906
8.914.2.6 p3GPPTraHdlPri . . . 906
8.914.2.7 pDataRate . . . 906
8.914.2.8 pJitter . . . 906
8.914.2.9 pLatency . . . 906
8.914.2.10 pLteQci . . . 906
8.914.2.11 pMaxAllowedPktSz . . . 907
8.914.2.12 pMinPolicedPktSz . . . 907
8.914.2.13 pPktErrRate . . . 907
8.914.2.14 pProfileId3GPP2 . . . 907
8.914.2.15 pTokenBucket . . . 907
8.914.2.16 pTrafficClass . . . 907
8.915 swiQosGranted Struct Reference . . . 907
8.915.1 Detailed Description . . . 907
8.915.2 Field Documentation . . . 907
8.915.2.1 pRxFlow . . . 907
8.915.2.2 pTxFlow . . . 907
8.916 swiQosIds Struct Reference . . . 907
8.916.1 Detailed Description . . . 907
8.916.2 Field Documentation . . . 907
8.916.2.1 pIds . . . 908
8.916.2.2 sz . . . 908
8.917 swiQosModifyReq Struct Reference . . . 908
8.917.1 Detailed Description . . . 908
8.917.2 Field Documentation . . . 908
8.917.2.1 id . . . 908
8.917.2.2 pRxFilter . . . 908
8.917.2.3 pRxFlow . . . 908
8.917.2.4 pTxFilter . . . 908
8.917.2.5 pTxFlow . . . 908
8.918 swiQosReq Struct Reference . . . 908

8.918.1 Detailed Description . . . 908
8.918.2 Field Documentation . . . 909
8.918.2.1 index . . . 909
8.918.2.2 pRxFilter . . . 909
8.918.2.3 pRxFlow . . . 909
8.918.2.4 pTxFilter . . . 909
8.918.2.5 pTxFlow . . . 909
8.919 swiRMTrasnferStaticsReq Struct Reference . . . 909
8.919.1 Detailed Description . . . 909
8.919.2 Field Documentation . . . 910
8.919.2.1 bResetStatistics . . . 910
8.919.2.2 ulMask . . . 910
8.920 sysInfoCommon Struct Reference . . . 910
8.920.1 Detailed Description . . . 910
8.920.2 Field Documentation . . . 912
8.920.2.1 isSysForbidden . . . 912
8.920.2.2 isSysForbiddenValid . . . 912
8.920.2.3 roamStatus . . . 912
8.920.2.4 roamStatusValid . . . 912
8.920.2.5 srvCapability . . . 912
8.920.2.6 srvCapabilityValid . . . 912
8.920.2.7 srvDomain . . . 912
8.920.2.8 srvDomainValid . . . 912
8.921 t_gpsTime Struct Reference . . . 912
8.921.1 Field Documentation . . . 912
8.921.1.1 gpsTimeOfWeekMs . . . 912
8.921.1.2 gpsWeek . . . 912
8.922 t_sensor Struct Reference . . . 912
8.922.1 Field Documentation . . . 912
8.922.1.1 aidingIndicatorMask . . . 912
8.922.1.2 usageMask . . . 912
8.923 t_Sv Struct Reference . . . 913
8.923.1 Field Documentation . . . 913
8.923.1.1 entries . . . 913
8.923.1.2 len . . . 913
8.924 TDSCDMAECIOThresh Struct Reference . . . 913
8.924.1 Detailed Description . . . 913
8.924.2 Field Documentation . . . 913
8.924.2.1 pTDSCDMAECIOThreshList . . . 913
8.924.2.2 TDSCDMAECIOThreshListLen . . . 913

8.925 TDSCDMARSCPThresh Struct Reference . . . 913
8.925.1 Detailed Description . . . 913
8.925.2 Field Documentation . . . 914
8.925.2.1 pTDSCDMARSCPThreshList . . . 914
8.925.2.2 TDSCDMARSCPThreshListLen . . . 914
8.926 TDSCDMARSSIThresh Struct Reference . . . 914
8.926.1 Detailed Description . . . 914
8.926.2 Field Documentation . . . 914
8.926.2.1 pTDSCDMARSSIThreshList . . . 914
8.926.2.2 TDSCDMARSSIThreshListLen . . . 914
8.927 tdscdmaSigInfoExt Struct Reference . . . 914
8.927.1 Detailed Description . . . 915
8.927.2 Field Documentation . . . 915
8.927.2.1 ecio . . . 915
8.927.2.2 rscp . . . 915
8.927.2.3 rssi . . . 915
8.927.2.4 sinr . . . 915
8.928 TDSCDMASigInfoExt Struct Reference . . . 915
8.928.1 Detailed Description . . . 915
8.928.2 Field Documentation . . . 915
8.928.2.1 ecio . . . 915
8.928.2.2 rscp . . . 915
8.928.2.3 rssi . . . 916
8.928.2.4 sinr . . . 916
8.929 TDSCDMASINRCONFThresh Struct Reference . . . 916
8.929.1 Detailed Description . . . 916
8.929.2 Field Documentation . . . 916
8.929.2.1 pTDSCDMASINRCONFThreshList . . . 916
8.929.2.2 TDSCDMASINRCONFThreshListLen . . . 916
8.930 TDSCDMASINRThresh Struct Reference . . . 916
8.930.1 Detailed Description . . . 916
8.930.2 Field Documentation . . . 917
8.930.2.1 pTDSCDMASINRThreshList . . . 917
8.930.2.2 TDSCDMASINRThreshListLen . . . 917
8.931 tempData_t Struct Reference . . . 917
8.931.1 Detailed Description . . . 917
8.931.2 Field Documentation . . . 917
8.931.2.1 temperature . . . 918
8.931.2.2 temperatureDataLen . . . 918
8.931.2.3 timeOfFirstSample . . . 918

8.931.2.4 timeOffset . . . 918
8.931.2.5 timeSource . . . 918
8.932 tempratureData Struct Reference . . . 918
8.932.1 Detailed Description . . . 918
8.932.2 Field Documentation . . . 919
8.932.2.1 temperature . . . 919
8.932.2.2 temperatureDataLen . . . 919
8.932.2.3 timeOfFirstSample . . . 919
8.932.2.4 timeOffset . . . 919
8.932.2.5 timeSource . . . 919
8.933 TFTIDParams Struct Reference . . . 919
8.933.1 Detailed Description . . . 919
8.933.2 Field Documentation . . . 920
8.933.2.1 destPortRangeEnd . . . 920
8.933.2.2 destPortRangeStart . . . 920
8.933.2.3 eValid . . . 920
8.933.2.4 filterId . . . 920
8.933.2.5 flowLabel . . . 920
8.933.2.6 IPSECSPI . . . 920
8.933.2.7 ipVersion . . . 920
8.933.2.8 nextHeader . . . 920
8.933.2.9 pSourceIP . . . 920
8.933.2.10 sourceIPMask . . . 920
8.933.2.11 srcPortRangeEnd . . . 920
8.933.2.12 srcPortRangeStart . . . 920
8.933.2.13 tosMask . . . 921
8.934 timeInfo Struct Reference . . . 921
8.934.1 Detailed Description . . . 921
8.934.2 Field Documentation . . . 922
8.934.2.1 day . . . 922
8.934.2.2 dayLtSavingAdj . . . 922
8.934.2.3 dayOfWeek . . . 922
8.934.2.4 hour . . . 922
8.934.2.5 minute . . . 922
8.934.2.6 month . . . 922
8.934.2.7 radioInterface . . . 922
8.934.2.8 second . . . 922
8.934.2.9 timeZone . . . 922
8.934.2.10 TlvPresent . . . 922
8.934.2.11 year . . . 922

8.935 TmdDeRegNotMitigationLvlReq Struct Reference . . . 922
8.935.1 Detailed Description . . . 923
8.935.2 Field Documentation . . . 923
8.935.2.1 mitigationDevID . . . 923
8.935.2.2 mitigationDevIDLen . . . 923
8.936 TmdGetMitigationDevListResp Struct Reference . . . 923
8.936.1 Detailed Description . . . 923
8.936.2 Field Documentation . . . 923
8.936.2.1 pMitigationDevList . . . 923
8.936.2.2 pMitigationDevListLen . . . 924
8.937 TmdGetMitigationLvlReq Struct Reference . . . 924
8.937.1 Detailed Description . . . 924
8.937.2 Field Documentation . . . 924
8.937.2.1 mitigationDevID . . . 924
8.937.2.2 mitigationDevIDLen . . . 924
8.938 TmdGetMitigationLvlResp Struct Reference . . . 924
8.938.1 Detailed Description . . . 924
8.938.2 Field Documentation . . . 925
8.938.2.1 pCurrentmitigationLvl . . . 925
8.938.2.2 pReqMitigationLvl . . . 925
8.939 TmdMitigationLvlIndReq Struct Reference . . . 925
8.939.1 Detailed Description . . . 925
8.939.2 Field Documentation . . . 925
8.939.2.1 mitigationDevID . . . 925
8.939.2.2 mitigationDevIDLen . . . 925
8.940 TmdRegNotMitigationLvlReq Struct Reference . . . 925
8.940.1 Detailed Description . . . 925
8.940.2 Field Documentation . . . 926
8.940.2.1 mitigationDevID . . . 926
8.940.2.2 mitigationDevIDLen . . . 926
8.941 tokenBucket Struct Reference . . . 926
8.941.1 Detailed Description . . . 926
8.941.2 Field Documentation . . . 926
8.941.2.1 bucketSz . . . 926
8.941.2.2 peakRate . . . 926
8.941.2.3 tokenRate . . . 926
8.942 Tos Struct Reference . . . 926
8.942.1 Detailed Description . . . 926
8.942.2 Field Documentation . . . 927
8.942.2.1 mask . . . 927

8.942.2.2 val . . . 927
8.943transferRouteMessageTlv Struct Reference . . . 927
8.943.1 Detailed Description . . . 927
8.943.2 Field Documentation . . . 927
8.943.2.1 TlvPresent . . . 927
8.943.2.2 TransferRouteMTMessageInfo . . . 927
8.944TransferStatInd Struct Reference . . . 927
8.944.1 Detailed Description . . . 927
8.944.2 Field Documentation . . . 928
8.944.2.1 StatsMask . . . 928
8.944.2.2 StatsPeriod . . . 928
8.945transferStatInd Struct Reference . . . 928
8.945.1 Detailed Description . . . 928
8.945.2 Field Documentation . . . 928
8.945.2.1 StatsMask . . . 928
8.945.2.2 StatsPeriod . . . 928
8.946TransferStatsDataType Struct Reference . . . 928
8.946.1 Field Documentation . . . 928
8.946.1.1 interval . . . 928
8.947TrStatInd Struct Reference . . . 928
8.947.1 Detailed Description . . . 929
8.947.2 Field Documentation . . . 929
8.947.2.1 statsMask . . . 929
8.947.2.2 statsPeriod . . . 929
8.948trueIMSI Struct Reference . . . 929
8.948.1 Detailed Description . . . 929
8.948.2 Field Documentation . . . 930
8.948.2.1 imsiT1112 . . . 930
8.948.2.2 imsiTaddrNum . . . 930
8.948.2.3 imsiTS1 . . . 930
8.948.2.4 imsiTS2 . . . 930
8.948.2.5 mccT . . . 930
8.949TXAGCList Struct Reference . . . 930
8.949.1 Detailed Description . . . 930
8.949.2 Field Documentation . . . 931
8.949.2.1 pTXAIG . . . 931
8.949.2.2 pTXComprSlope . . . 931
8.949.2.3 pTXComprThres . . . 931
8.949.2.4 pTXExpSlope . . . 931
8.949.2.5 pTXExpThres . . . 931

8.949.2.6 pTXStaticGain . . . 931
8.950 txInfo Struct Reference . . . 931
8.950.1 Detailed Description . . . 931
8.950.2 Field Documentation . . . 931
8.950.2.1 isInTraffic . . . 932
8.950.2.2 txPower . . . 932
8.951 TXPCMIIRFltr Struct Reference . . . 932
8.951.1 Detailed Description . . . 932
8.951.2 Field Documentation . . . 933
8.951.2.1 pFlag . . . 933
8.951.2.2 pStage0Val . . . 933
8.951.2.3 pStage1Val . . . 933
8.951.2.4 pStage2Val . . . 933
8.951.2.5 pStage3Val . . . 933
8.951.2.6 pStage4Val . . . 933
8.951.2.7 pStageCnt . . . 933
8.952 uim_appStatus Struct Reference . . . 933
8.952.1 Detailed Description . . . 934
8.952.2 Field Documentation . . . 936
8.952.2.1 aidLength . . . 936
8.952.2.2 aidVal . . . 936
8.952.2.3 appState . . . 936
8.952.2.4 appType . . . 936
8.952.2.5 persoFeature . . . 936
8.952.2.6 persoRetries . . . 936
8.952.2.7 persoState . . . 936
8.952.2.8 persoUnblockRetries . . . 936
8.952.2.9 pin1Retries . . . 936
8.952.2.10 pin1State . . . 936
8.952.2.11 pin2Retries . . . 936
8.952.2.12 pin2State . . . 936
8.952.2.13 puk1Retries . . . 936
8.952.2.14 puk2Retries . . . 936
8.952.2.15 univPin . . . 936
8.953 uim_authenticateResult Struct Reference . . . 936
8.953.1 Detailed Description . . . 937
8.953.2 Field Documentation . . . 937
8.953.2.1 content . . . 937
8.953.2.2 contentLen . . . 937
8.954 uim_authenticationData Struct Reference . . . 937

8.954.1 Detailed Description . . . 937
8.954.2 Field Documentation . . . 938
8.954.2.1 context . . . 938
8.954.2.2 data . . . 938
8.954.2.3 dataLen . . . 938
8.955 uim_cardResult Struct Reference . . . 938
8.955.1 Detailed Description . . . 938
8.955.2 Field Documentation . . . 939
8.955.2.1 sw1 . . . 939
8.955.2.2 sw2 . . . 939
8.956 uim_cardStatus Struct Reference . . . 939
8.956.1 Detailed Description . . . 939
8.956.2 Field Documentation . . . 940
8.956.2.1 index1xPri . . . 940
8.956.2.2 index1xSec . . . 940
8.956.2.3 indexGwPri . . . 940
8.956.2.4 indexGwSec . . . 940
8.956.2.5 numSlot . . . 940
8.956.2.6 SlotInfo . . . 940
8.957 uim_changeUIMPIN Struct Reference . . . 940
8.957.1 Detailed Description . . . 940
8.957.2 Field Documentation . . . 941
8.957.2.1 oldPINLen . . . 941
8.957.2.2 oldPINVal . . . 941
8.957.2.3 pinID . . . 941
8.957.2.4 pinLen . . . 941
8.957.2.5 pinVal . . . 941
8.958 uim_depersonalizationInformation Struct Reference . . . 941
8.958.1 Detailed Description . . . 941
8.958.2 Field Documentation . . . 942
8.958.2.1 ckLen . . . 942
8.958.2.2 ckVal . . . 942
8.958.2.3 feature . . . 942
8.958.2.4 operation . . . 942
8.959 uim_encryptedPIN1 Struct Reference . . . 942
8.959.1 Detailed Description . . . 942
8.959.2 Field Documentation . . . 943
8.959.2.1 pin1Len . . . 943
8.959.2.2 pin1Val . . . 943
8.960 uim_fileAttributes Struct Reference . . . 943

8.960.1 Detailed Description . . . 943
8.960.2 Field Documentation . . . 946
8.960.2.1 fileID . . . 946
8.960.2.2 fileSize . . . 946
8.960.2.3 fileType . . . 946
8.960.2.4 rawLen . . . 946
8.960.2.5 rawValue . . . 946
8.960.2.6 recordCount . . . 946
8.960.2.7 recordSize . . . 946
8.960.2.8 secActivate . . . 946
8.960.2.9 secActivateMask . . . 946
8.960.2.10 secDeactivate . . . 946
8.960.2.11 secDeactivateMask . . . 946
8.960.2.12 secIncrease . . . 946
8.960.2.13 secIncreaseMask . . . 946
8.960.2.14 secRead . . . 946
8.960.2.15 secReadMask . . . 946
8.960.2.16 secWrite . . . 946
8.960.2.17 secWriteMask . . . 946
8.961 uim_fileInfo Struct Reference . . . 946
8.961.1 Detailed Description . . . 946
8.961.2 Field Documentation . . . 947
8.961.2.1 fileID . . . 947
8.961.2.2 path . . . 947
8.961.2.3 pathLen . . . 947
8.962 uim_hotSwapStatus Struct Reference . . . 947
8.962.1 Detailed Description . . . 947
8.962.2 Field Documentation . . . 947
8.962.2.1 hotSwap . . . 947
8.962.2.2 hotSwapLength . . . 947
8.963 uim_personalizationStatus Struct Reference . . . 947
8.963.1 Detailed Description . . . 948
8.963.2 Field Documentation . . . 948
8.963.2.1 feature . . . 948
8.963.2.2 numFeatures . . . 948
8.963.2.3 unblockLeft . . . 948
8.963.2.4 verifyLeft . . . 948
8.964 uim_readResult Struct Reference . . . 948
8.964.1 Detailed Description . . . 949
8.964.2 Field Documentation . . . 949

8.964.2.1 content . . . 949
8.964.2.2 contentLen . . . 949
8.965 uim_readTransparentInfo Struct Reference . . . 949
8.965.1 Detailed Description . . . 949
8.965.2 Field Documentation . . . 949
8.965.2.1 length . . . 949
8.965.2.2 offset . . . 949
8.966 uim_refreshevent Struct Reference . . . 949
8.966.1 Detailed Description . . . 950
8.966.2 Field Documentation . . . 951
8.966.2.1 aid . . . 951
8.966.2.2 aidLength . . . 951
8.966.2.3 arrfileInfo . . . 951
8.966.2.4 mode . . . 951
8.966.2.5 numOfFiles . . . 951
8.966.2.6 sessionType . . . 951
8.966.2.7 stage . . . 951
8.967 uim_registerRefresh Struct Reference . . . 951
8.967.1 Detailed Description . . . 951
8.967.2 Field Documentation . . . 952
8.967.2.1 arrfileInfo . . . 952
8.967.2.2 numFiles . . . 952
8.967.2.3 registerFlag . . . 952
8.967.2.4 voteForInit . . . 952
8.968 uim_remainingRetries Struct Reference . . . 952
8.968.1 Detailed Description . . . 952
8.968.2 Field Documentation . . . 952
8.968.2.1 unblockLeft . . . 952
8.968.2.2 verifyLeft . . . 953
8.969 uim_sessionInformation Struct Reference . . . 953
8.969.1 Detailed Description . . . 953
8.969.2 Field Documentation . . . 953
8.969.2.1 aid . . . 953
8.969.2.2 aidLength . . . 953
8.969.2.3 sessionType . . . 953
8.970 uim_setPINProtection Struct Reference . . . 953
8.970.1 Detailed Description . . . 954
8.970.2 Field Documentation . . . 954
8.970.2.1 pinID . . . 954
8.970.2.2 pinLength . . . 954

8.970.2.3 pinOperation . . . 954
8.970.2.4 pinValue . . . 954
8.971 uim_slotInfo Struct Reference . . . 954
8.971.1 Detailed Description . . . 955
8.971.2 Field Documentation . . . 956
8.971.2.1 AppStatus . . . 956
8.971.2.2 cardState . . . 956
8.971.2.3 errorState . . . 956
8.971.2.4 numApp . . . 956
8.971.2.5 upinRetries . . . 956
8.971.2.6 upinState . . . 956
8.971.2.7 upukRetries . . . 956
8.972 uim_UIMSessionInformation Struct Reference . . . 956
8.972.1 Detailed Description . . . 956
8.972.2 Field Documentation . . . 957
8.972.2.1 aid . . . 957
8.972.2.2 aidLength . . . 957
8.972.2.3 sessionType . . . 957
8.973 uim_unblockUIMPIN Struct Reference . . . 957
8.973.1 Detailed Description . . . 957
8.973.2 Field Documentation . . . 957
8.973.2.1 newPINLen . . . 957
8.973.2.2 newPINVal . . . 957
8.973.2.3 pinID . . . 957
8.973.2.4 pukLen . . . 958
8.973.2.5 pukVal . . . 958
8.974 uim_verifyUIMPIN Struct Reference . . . 958
8.974.1 Detailed Description . . . 958
8.974.2 Field Documentation . . . 958
8.974.2.1 pinID . . . 958
8.974.2.2 pinLen . . . 958
8.974.2.3 pinVal . . . 958
8.975 UIMAuthenticateReq Struct Reference . . . 958
8.975.1 Detailed Description . . . 958
8.975.2 Field Documentation . . . 959
8.975.2.1 authData . . . 959
8.975.2.2 pIndicationToken . . . 959
8.975.2.3 sessionInfo . . . 959
8.976 UIMAuthenticateResp Struct Reference . . . 959
8.976.1 Detailed Description . . . 959

8.976.2 Field Documentation . . . 960
8.976.2.1 pAuthenticateResult . . . 960
8.976.2.2 pCardResult . . . 960
8.976.2.3 pIndicationToken . . . 960
8.977 UIMChangePinReq Struct Reference . . . 960
8.977.1 Detailed Description . . . 960
8.977.2 Field Documentation . . . 960
8.977.2.1 changePIN . . . 960
8.977.2.2 pIndicationToken . . . 960
8.977.2.3 pKeyReferenceID . . . 960
8.977.2.4 sessionInfo . . . 960
8.978 UIMDepersonalizationReq Struct Reference . . . 961
8.978.1 Detailed Description . . . 961
8.978.2 Field Documentation . . . 961
8.978.2.1 depersonilisationInfo . . . 961
8.979 UIMDepersonalizationResp Struct Reference . . . 961
8.979.1 Detailed Description . . . 961
8.979.2 Field Documentation . . . 961
8.979.2.1 pRemainingRetries . . . 961
8.980 UIMEventRegisterReqResp Struct Reference . . . 961
8.980.1 Detailed Description . . . 962
8.980.2 Field Documentation . . . 962
8.980.2.1 eventMask . . . 962
8.981 UIMGetCardStatusResp Struct Reference . . . 962
8.981.1 Detailed Description . . . 962
8.981.2 Field Documentation . . . 962
8.981.2.1 pCardStatus . . . 962
8.981.2.2 pHotSwapStatus . . . 962
8.982 UIMGetConfigurationReq Struct Reference . . . 962
8.982.1 Detailed Description . . . 963
8.982.2 Field Documentation . . . 963
8.982.2.1 pConfigurationMask . . . 963
8.983 UIMGetConfigurationResp Struct Reference . . . 963
8.983.1 Detailed Description . . . 963
8.983.2 Field Documentation . . . 964
8.983.2.1 pAutoSelection . . . 964
8.983.2.2 pHaltSubscription . . . 964
8.983.2.3 pPersonalizationStatus . . . 964
8.984 UIMGetFileAttributesReq Struct Reference . . . 964
8.984.1 Detailed Description . . . 964

8.984.2 Field Documentation . . . 964
8.984.2.1 fileIndex . . . 964
8.984.2.2 pIndicationToken . . . 964
8.984.2.3 sessionInfo . . . 964
8.985 UIMGetFileAttributesResp Struct Reference . . . 964
8.985.1 Detailed Description . . . 965
8.985.2 Field Documentation . . . 965
8.985.2.1 pCardResult . . . 965
8.985.2.2 pFileAttributes . . . 965
8.985.2.3 pIndicationToken . . . 965
8.986 UIMGetSlotsStatusResp Struct Reference . . . 965
8.986.1 Detailed Description . . . 965
8.986.2 Field Documentation . . . 965
8.986.2.1 pNumberOfPhySlot . . . 965
8.986.2.2 pUimSlotsStatus . . . 966
8.987 UIMPinResp Struct Reference . . . 966
8.987.1 Detailed Description . . . 966
8.987.2 Field Documentation . . . 966
8.987.2.1 pEncryptedPIN1 . . . 966
8.987.2.2 pIndicationToken . . . 966
8.987.2.3 pRemainingRetries . . . 966
8.988 UIMPowerDownReq Struct Reference . . . 966
8.988.1 Detailed Description . . . 966
8.988.2 Field Documentation . . . 967
8.988.2.1 slot . . . 967
8.989 UIMPowerUpReq Struct Reference . . . 967
8.989.1 Detailed Description . . . 967
8.989.2 Field Documentation . . . 967
8.989.2.1 pIgnoreHotSwapSwitch . . . 967
8.989.2.2 slot . . . 967
8.990 UIMReadTransparentReq Struct Reference . . . 967
8.990.1 Detailed Description . . . 968
8.990.2 Field Documentation . . . 968
8.990.2.1 fileIndex . . . 968
8.990.2.2 pEncryptData . . . 968
8.990.2.3 pIndicationToken . . . 968
8.990.2.4 readTransparent . . . 968
8.990.2.5 sessionInfo . . . 968
8.991 UIMReadTransparentResp Struct Reference . . . 968
8.991.1 Detailed Description . . . 968

8.991.2 Field Documentation . . . 969
8.991.2.1 pCardResult . . . 969
8.991.2.2 pEncryptedData . . . 969
8.991.2.3 pIndicationToken . . . 969
8.991.2.4 pReadResult . . . 969
8.992 UIMRefreshCompleteReq Struct Reference . . . 969
8.992.1 Detailed Description . . . 969
8.992.2 Field Documentation . . . 970
8.992.2.1 refreshComplete . . . 970
8.992.2.2 sessionInfo . . . 970
8.993 UIMRefreshEvent Struct Reference . . . 970
8.993.1 Detailed Description . . . 970
8.993.2 Field Documentation . . . 971
8.993.2.1 aid . . . 971
8.993.2.2 aidLength . . . 971
8.993.2.3 arrfileInfo . . . 971
8.993.2.4 mode . . . 971
8.993.2.5 numOfFiles . . . 971
8.993.2.6 sessionType . . . 971
8.993.2.7 stage . . . 971
8.994 UIMRefreshGetLastEventReq Struct Reference . . . 971
8.994.1 Detailed Description . . . 972
8.994.2 Field Documentation . . . 972
8.994.2.1 sessionInfo . . . 972
8.995 UIMRefreshGetLastEventResp Struct Reference . . . 972
8.995.1 Detailed Description . . . 972
8.995.2 Field Documentation . . . 972
8.995.2.1 pRefreshEvent . . . 972
8.996 UIMRefreshOKReq Struct Reference . . . 972
8.996.1 Detailed Description . . . 972
8.996.2 Field Documentation . . . 973
8.996.2.1 OKtoRefresh . . . 973
8.996.2.2 sessionInfo . . . 973
8.997 UIMRefreshRegisterReq Struct Reference . . . 973
8.997.1 Detailed Description . . . 973
8.997.2 Field Documentation . . . 973
8.997.2.1 regRefresh . . . 973
8.997.2.2 sessionInfo . . . 973
8.998 UIMSessionInformation Struct Reference . . . 973
8.998.1 Detailed Description . . . 974

8.998.2 Field Documentation . . . 974
8.998.2.1 aid . . . 974
8.998.2.2 aidLength . . . 974
8.998.2.3 sessionType . . . 974
8.999 UIMSetPinProtectionReq Struct Reference . . . 974
8.999.1 Detailed Description . . . 974
8.999.2 Field Documentation . . . 975
8.999.2.1 pIndicationToken . . . 975
8.999.2.2 pinProtection . . . 975
8.999.2.3 pKeyReferenceID . . . 975
8.999.2.4 sessionInfo . . . 975
8.1000 UIMSlotsStatus Struct Reference . . . 975
8.1000.1 Detailed Description . . . 975
8.1000.2 Field Documentation . . . 975
8.1000.2.1 uimSlotStatus . . . 975
8.1001 UIMSlotStatus Struct Reference . . . 975
8.1001.1 Detailed Description . . . 976
8.1001.2 Field Documentation . . . 976
8.1001.2.1 bICCID . . . 976
8.1001.2.2 bICCIDLength . . . 976
8.1001.2.3 bLogicalSlot . . . 976
8.1001.2.4 uPhyCardStatus . . . 976
8.1001.2.5 uPhySlotStatus . . . 976
8.1002 UIMSlotStatusChangeInfo Struct Reference . . . 977
8.1002.1 Detailed Description . . . 977
8.1002.2 Field Documentation . . . 977
8.1002.2.1 bNumberOfPhySlots . . . 977
8.1002.2.2 slotsstatusChange . . . 977
8.1003 UIMStatusChangeInfo Struct Reference . . . 977
8.1003.1 Detailed Description . . . 977
8.1003.2 Field Documentation . . . 977
8.1003.2.1 statusChange . . . 977
8.1004 UIMSwitchSlotReq Struct Reference . . . 977
8.1004.1 Detailed Description . . . 978
8.1004.2 Field Documentation . . . 978
8.1004.2.1 bLogicalSlot . . . 978
8.1004.2.2 ulPhysicalSlot . . . 978
8.1005 UIMUnblockPinReq Struct Reference . . . 978
8.1005.1 Detailed Description . . . 978
8.1005.2 Field Documentation . . . 979

8.1005.2.1 pIndicationToken . . . 979
8.1005.2.2 pKeyReferenceID . . . 979
8.1005.2.3 sessionInfo . . . 979
8.1005.2.4 unblockPIN . . . 979
8.1006 UIMVerifyPinReq Struct Reference . . . 979
8.1006.1 Detailed Description . . . 979
8.1006.2 Field Documentation . . . 980
8.1006.2.1 pEncryptedPIN1 . . . 980
8.1006.2.2 pIndicationToken . . . 980
8.1006.2.3 pKeyReferenceID . . . 980
8.1006.2.4 sessionInfo . . . 980
8.1006.2.5 verifyPIN . . . 980
8.1007 UMTSInfo Struct Reference . . . 980
8.1007.1 Detailed Description . . . 980
8.1007.2 Field Documentation . . . 981
8.1007.2.1 cellID . . . 981
8.1007.2.2 ecio . . . 981
8.1007.2.3 geranInst . . . 981
8.1007.2.4 GeranInstInfo . . . 981
8.1007.2.5 lac . . . 981
8.1007.2.6 plmn . . . 982
8.1007.2.7 psc . . . 982
8.1007.2.8 rscp . . . 982
8.1007.2.9 uarfcn . . . 982
8.1007.2.10 umtsInst . . . 982
8.1007.2.11 UMTSInstInfo . . . 982
8.1008 UMTSinstInfo Struct Reference . . . 982
8.1008.1 Detailed Description . . . 982
8.1008.2 Field Documentation . . . 982
8.1008.2.1 umtsEcio . . . 982
8.1008.2.2 umtsPsc . . . 982
8.1008.2.3 umtsRscp . . . 982
8.1008.2.4 umtsUarfcn . . . 982
8.1009 umtsLTENbrCell Struct Reference . . . 982
8.1009.1 Detailed Description . . . 983
8.1009.2 Field Documentation . . . 983
8.1009.2.1 cellIsTDD . . . 983
8.1009.2.2 earfcn . . . 983
8.1009.2.3 pci . . . 983
8.1009.2.4 rsrp . . . 983

8.1009.2.5 rsrq . . . 983
8.1009.2.6 srxlev . . . 983
8.1010 UMTSMinQoS Struct Reference . . . 983
8.1010.1 Detailed Description . . . 984
8.1010.2 Field Documentation . . . 985
8.1010.2.1 deliveryErrSDU . . . 985
8.1010.2.2 grntDownlinkBitrate . . . 986
8.1010.2.3 grntUplinkBitrate . . . 986
8.1010.2.4 maxDownlinkBitrate . . . 986
8.1010.2.5 maxSDUSize . . . 986
8.1010.2.6 maxUplinkBitrate . . . 986
8.1010.2.7 qosDeliveryOrder . . . 986
8.1010.2.8 resBerRatio . . . 986
8.1010.2.9 sduErrorRatio . . . 986
8.1010.2.10 trafficClass . . . 986
8.1010.2.11 trafficPriority . . . 986
8.1010.2.12 transferDelay . . . 986
8.1011 UMTSQoS Struct Reference . . . 986
8.1011.1 Detailed Description . . . 986
8.1011.2 Field Documentation . . . 988
8.1011.2.1 deliveryErrSDU . . . 988
8.1011.2.2 grntDownlinkBitrate . . . 988
8.1011.2.3 grntUplinkBitrate . . . 988
8.1011.2.4 maxDownlinkBitrate . . . 988
8.1011.2.5 maxSDUSize . . . 988
8.1011.2.6 maxUplinkBitrate . . . 988
8.1011.2.7 qosDeliveryOrder . . . 988
8.1011.2.8 resBerRatio . . . 988
8.1011.2.9 sduErrorRatio . . . 988
8.1011.2.10 trafficClass . . . 988
8.1011.2.11 trafficPriority . . . 988
8.1011.2.12 transferDelay . . . 988
8.1012 UMTSReqQoSSigInd Struct Reference . . . 988
8.1012.1 Detailed Description . . . 989
8.1012.2 Field Documentation . . . 989
8.1012.2.1 sigInd . . . 989
8.1012.2.2 UMTSReqQoS . . . 989
8.1013 unblockUIMPIN Struct Reference . . . 989
8.1013.1 Detailed Description . . . 989
8.1013.2 Field Documentation . . . 990

8.1013.2.1 newPINLen . . . 990
8.1013.2.2 newPINVal . . . 990
8.1013.2.3 pinID . . . 990
8.1013.2.4 pukLen . . . 990
8.1013.2.5 pukVal . . . 990
8.1014 UniversalTime Struct Reference . . . 990
8.1014.1 Detailed Description . . . 990
8.1014.2 Field Documentation . . . 991
8.1014.2.1 day . . . 991
8.1014.2.2 dayOfWeek . . . 991
8.1014.2.3 hour . . . 991
8.1014.2.4 minute . . . 991
8.1014.2.5 month . . . 991
8.1014.2.6 second . . . 991
8.1014.2.7 year . . . 991
8.1015 unpack_dms_GetActivationState_t Struct Reference . . . 991
8.1015.1 Detailed Description . . . 991
8.1015.2 Field Documentation . . . 992
8.1015.2.1 state . . . 992
8.1016 unpack_dms_GetBandCapability_t Struct Reference . . . 992
8.1016.1 Field Documentation . . . 992
8.1016.1.1 BandCapability . . . 992
8.1016.1.2 Tlvresult . . . 992
8.1017 unpack_dms_GetCrashAction_t Struct Reference . . . 992
8.1017.1 Field Documentation . . . 992
8.1017.1.1 DevCrashState . . . 992
8.1017.1.2 Tlvresult . . . 992
8.1018 unpack_dms_GetCustFeature_t Struct Reference . . . 992
8.1018.1 Field Documentation . . . 993
8.1018.1.1 DHCPRelayEnabled . . . 993
8.1018.1.2 DisableIMSI . . . 993
8.1018.1.3 GpsEnable . . . 993
8.1018.1.4 GPSLPM . . . 993
8.1018.1.5 GPSSel . . . 993
8.1018.1.6 IPFamSupport . . . 993
8.1018.1.7 IsVoiceEnabled . . . 993
8.1018.1.8 RMAutoConnect . . . 993
8.1018.1.9 SMSSupport . . . 993
8.1018.1.10 Tlvresult . . . 993
8.1019 unpack_dms_GetCustFeaturesV2_t Struct Reference . . . 993

8.1019.1 Detailed Description . . . 993
8.1019.2 Field Documentation . . . 993
8.1019.2.1 GetCustomFeatureV2 . . . 993
8.1019.2.2 Tlvresult . . . 993
8.1020 unpack_dms_GetDeviceCap_t Struct Reference . . . 994
8.1020.1 Field Documentation . . . 994
8.1020.1.1 DataServiceCapability . . . 994
8.1020.1.2 MaxRXChannelRate . . . 994
8.1020.1.3 MaxTXChannelRate . . . 994
8.1020.1.4 RadioIfaces . . . 994
8.1020.1.5 RadioIfacesSize . . . 994
8.1020.1.6 SimCapability . . . 994
8.1020.1.7 Tlvresult . . . 994
8.1021 unpack_dms_GetDeviceCapabilities_t Struct Reference . . . 994
8.1021.1 Detailed Description . . . 994
8.1021.2 Field Documentation . . . 995
8.1021.2.1 dataServiceCaCapability . . . 995
8.1021.2.2 maxRxChannelRate . . . 995
8.1021.2.3 maxTxChannelRate . . . 995
8.1021.2.4 RadioIfaces . . . 995
8.1021.2.5 radioIfacesSize . . . 995
8.1021.2.6 simCapability . . . 995
8.1022 unpack_dms_GetDeviceHardwareRev_t Struct Reference . . . 995
8.1022.1 Field Documentation . . . 995
8.1022.1.1 String . . . 995
8.1022.1.2 stringSize . . . 995
8.1022.1.3 Tlvresult . . . 995
8.1023 unpack_dms_GetDeviceMfr_t Struct Reference . . . 995
8.1023.1 Field Documentation . . . 995
8.1023.1.1 String . . . 995
8.1023.1.2 stringSize . . . 995
8.1023.1.3 Tlvresult . . . 995
8.1024 unpack_dms_GetDeviceSerialNumbers_t Struct Reference . . . 995
8.1024.1 Field Documentation . . . 996
8.1024.1.1 esnSize . . . 996
8.1024.1.2 ESNString . . . 996
8.1024.1.3 meiSize . . . 996
8.1024.1.4 MEIString . . . 996
8.1024.1.5 meiSvnSize . . . 996
8.1024.1.6 meiSvnString . . . 996

8.1024.1.7 meidSize . . . 996
8.1024.1.8 MEIDString . . . 996
8.1024.1.9 Tlvresult . . . 996
8.1025 unpack_dms_GetFirmwareInfo_t Struct Reference . . . 996
8.1025.1 Detailed Description . . . 996
8.1025.2 Field Documentation . . . 997
8.1025.2.1 appversion_str . . . 997
8.1025.2.2 bootversion_str . . . 997
8.1025.2.3 carrier_str . . . 997
8.1025.2.4 cur_carr_name . . . 997
8.1025.2.5 cur_carr_rev . . . 997
8.1025.2.6 modelid_str . . . 997
8.1025.2.7 packageid_str . . . 997
8.1025.2.8 priversion_str . . . 997
8.1025.2.9 sku_str . . . 997
8.1025.2.10 Tlvresult . . . 997
8.1026 unpack_dms_GetFirmwareRevision_t Struct Reference . . . 997
8.1026.1 Field Documentation . . . 997
8.1026.1.1 amssSize . . . 997
8.1026.1.2 AMSSString . . . 997
8.1026.1.3 PRIString . . . 997
8.1026.1.4 Tlvresult . . . 997
8.1027 unpack_dms_GetFirmwareRevisions_t Struct Reference . . . 997
8.1027.1 Detailed Description . . . 998
8.1027.2 Field Documentation . . . 998
8.1027.2.1 amssSize . . . 998
8.1027.2.2 AMSSString . . . 998
8.1027.2.3 bootSize . . . 998
8.1027.2.4 BootString . . . 998
8.1027.2.5 priSize . . . 998
8.1027.2.6 PRIString . . . 998
8.1027.2.7 Tlvresult . . . 998
8.1028 unpack_dms_GetFSN_t Struct Reference . . . 998
8.1028.1 Field Documentation . . . 998
8.1028.1.1 String . . . 998
8.1028.1.2 Tlvresult . . . 998
8.1029 unpack_dms_GetHardwareRevision_t Struct Reference . . . 998
8.1029.1 Detailed Description . . . 998
8.1029.2 Field Documentation . . . 999
8.1029.2.1 hwVer . . . 999

8.1030 unpack_dms_GetIMSI_t Struct Reference . . . 999
8.1030.1 Field Documentation . . . 999
8.1030.1.1 imsi . . . 999
8.1030.1.2 TlvResult . . . 999
8.1031 unpack_dms_GetManufacturer_t Struct Reference . . . 999
8.1031.1 Detailed Description . . . 999
8.1031.2 Field Documentation . . . 999
8.1031.2.1 manufacturer . . . 999
8.1031.2.2 TlvResult . . . 999
8.1032 unpack_dms_GetModelID_t Struct Reference . . . 999
8.1032.1 Detailed Description . . . 1000
8.1032.2 Field Documentation . . . 1000
8.1032.2.1 modelId . . . 1000
8.1032.2.2 TlvResult . . . 1000
8.1033 unpack_dms_GetNetworkTime_t Struct Reference . . . 1000
8.1033.1 Detailed Description . . . 1000
8.1033.2 Field Documentation . . . 1000
8.1033.2.1 source . . . 1000
8.1033.2.2 timestamp . . . 1000
8.1033.2.3 TlvResult . . . 1000
8.1034 unpack_dms_GetOfflineReason_t Struct Reference . . . 1001
8.1034.1 Detailed Description . . . 1001
8.1034.2 Field Documentation . . . 1001
8.1034.2.1 pbPlatform . . . 1001
8.1034.2.2 pReasonMask . . . 1001
8.1034.2.3 TlvResult . . . 1001
8.1035 unpack_dms_GetPower_t Struct Reference . . . 1001
8.1035.1 Detailed Description . . . 1001
8.1035.2 Field Documentation . . . 1002
8.1035.2.1 HardwareControlledMode . . . 1002
8.1035.2.2 OfflineReason . . . 1002
8.1035.2.3 OperationMode . . . 1002
8.1035.2.4 TlvResult . . . 1002
8.1036 unpack_dms_GetPRLVersion_t Struct Reference . . . 1002
8.1036.1 Field Documentation . . . 1002
8.1036.1.1 TlvResult . . . 1002
8.1036.1.2 u16PRLVersion . . . 1002
8.1036.1.3 u8PRLPreference . . . 1002
8.1037 unpack_dms_GetSerialNumbers_t Struct Reference . . . 1002
8.1037.1 Detailed Description . . . 1002

8.1037.2 Field Documentation . . . 1002
8.1037.2.1 esn . . . 1003
8.1037.2.2 mei_no . . . 1003
8.1037.2.3 meisv_svn . . . 1003
8.1037.2.4 meid . . . 1003
8.1038 npack_dms_GetUSBComp_t Struct Reference . . . 1003
8.1038.1 Field Documentation . . . 1003
8.1038.1.1 NumSupUSBComps . . . 1003
8.1038.1.2 SupUSBComps . . . 1003
8.1038.1.3 Tlvresult . . . 1003
8.1038.1.4 USBComp . . . 1003
8.1039 npack_dms_GetVoiceNumber_t Struct Reference . . . 1003
8.1039.1 Field Documentation . . . 1003
8.1039.1.1 MIN . . . 1003
8.1039.1.2 minSize . . . 1003
8.1039.1.3 Tlvresult . . . 1003
8.1039.1.4 VoiceNumber . . . 1003
8.1039.1.5 voiceNumberSize . . . 1003
8.1040 npack_dms_ResetToFactoryDefaults_t Struct Reference . . . 1003
8.1040.1 Detailed Description . . . 1004
8.1040.2 Field Documentation . . . 1004
8.1040.2.1 Tlvresult . . . 1004
8.1041 npack_dms_SetActivationStatusCallback_t Struct Reference . . . 1004
8.1041.1 Detailed Description . . . 1004
8.1041.2 Field Documentation . . . 1004
8.1041.2.1 Tlvresult . . . 1004
8.1042 npack_dms_SetCrashAction_t Struct Reference . . . 1004
8.1042.1 Detailed Description . . . 1004
8.1042.2 Field Documentation . . . 1004
8.1042.2.1 notused . . . 1005
8.1043 npack_dms_SetCustFeature_t Struct Reference . . . 1005
8.1043.1 Field Documentation . . . 1005
8.1043.1.1 Tlvresult . . . 1005
8.1044 npack_dms_SetCustFeaturesV2_t Struct Reference . . . 1005
8.1044.1 Detailed Description . . . 1005
8.1044.2 Field Documentation . . . 1005
8.1044.2.1 Tlvresult . . . 1005
8.1045 npack_dms_SetEventReport_ind_t Struct Reference . . . 1005
8.1045.1 Detailed Description . . . 1005
8.1045.2 Field Documentation . . . 1006

8.1045.2.1 ActivationStatusTlv . . . 1006
8.1045.2.2 OperatingModeTlv . . . 1006
8.1045.2.3 Tlvresult . . . 1006
8.1046 npack_dms_SetEventReport_t Struct Reference . . . 1006
8.1046.1 Field Documentation . . . 1006
8.1046.1.1 Tlvresult . . . 1006
8.1047 npack_dms_SetFirmwarePreference_t Struct Reference . . . 1006
8.1047.1 Field Documentation . . . 1006
8.1047.1.1 Tlvresult . . . 1006
8.1048 npack_dms_SetPower_t Struct Reference . . . 1006
8.1048.1 Field Documentation . . . 1006
8.1048.1.1 Tlvresult . . . 1006
8.1049 npack_dms_SetUSBComp_t Struct Reference . . . 1006
8.1049.1 Field Documentation . . . 1007
8.1049.1.1 Tlvresult . . . 1007
8.1050 npack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference . . . 1007
8.1050.1 Detailed Description . . . 1007
8.1050.2 Field Documentation . . . 1007
8.1050.2.1 source . . . 1007
8.1050.2.2 Tlvresult . . . 1007
8.1050.2.3 type . . . 1007
8.1051 npack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference . . . 1008
8.1051.1 Detailed Description . . . 1008
8.1051.2 Field Documentation . . . 1008
8.1051.2.1 source . . . 1008
8.1051.2.2 Tlvresult . . . 1008
8.1051.2.3 type . . . 1008
8.1052 npack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 1008
8.1052.1 Detailed Description . . . 1009
8.1052.2 Field Documentation . . . 1009
8.1052.2.1 Tlvresult . . . 1009
8.1053 npack_dms_SLQSGetBandCapability_t Struct Reference . . . 1009
8.1053.1 Detailed Description . . . 1009
8.1053.2 Field Documentation . . . 1012
8.1053.2.1 bandCapability . . . 1012
8.1053.2.2 is_LteBandCapability_Available . . . 1012
8.1053.2.3 is_TdsBandCapability_Available . . . 1012
8.1053.2.4 LteBandCapability . . . 1012
8.1053.2.5 TdsBandCapability . . . 1012
8.1054 npack_dms_SLQSGetERIFile_t Struct Reference . . . 1012

8.1054.1 Detailed Description . . . 1012
8.1054.2 Field Documentation . . . 1012
8.1054.2.1 eriFile . . . 1012
8.1054.2.2 Tlvresult . . . 1012
8.1055 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference . . . 1012
8.1055.1 Detailed Description . . . 1013
8.1055.2 Field Documentation . . . 1013
8.1055.2.1 Tlvresult . . . 1013
8.1056 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 1013
8.1056.1 Detailed Description . . . 1013
8.1056.2 Field Documentation . . . 1013
8.1056.2.1 crashInfoParam . . . 1013
8.1056.2.2 Tlvresult . . . 1013
8.1057 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference . . . 1013
8.1057.1 Detailed Description . . . 1013
8.1057.2 Field Documentation . . . 1014
8.1057.2.1 pGetDyingGaspCfg . . . 1014
8.1057.2.2 Tlvresult . . . 1014
8.1058 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference . . . 1014
8.1058.1 Detailed Description . . . 1014
8.1058.2 Field Documentation . . . 1014
8.1058.2.1 pGetDyingGaspStatistics . . . 1014
8.1058.2.2 Tlvresult . . . 1014
8.1059 unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference . . . 1014
8.1059.1 Detailed Description . . . 1014
8.1059.2 Field Documentation . . . 1015
8.1059.2.1 carrier . . . 1015
8.1059.2.2 fwvers . . . 1015
8.1059.2.3 numEntries . . . 1015
8.1059.2.4 pCurrImgInfo . . . 1015
8.1059.2.5 pkgver . . . 1015
8.1059.2.6 priver . . . 1015
8.1060 unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference . . . 1015
8.1060.1 Detailed Description . . . 1015
8.1060.2 Field Documentation . . . 1016
8.1060.2.1 imgType . . . 1016
8.1060.2.2 logString . . . 1016
8.1060.2.3 refData . . . 1016
8.1060.2.4 refString . . . 1016
8.1060.2.5 ResCode . . . 1016

8.1060.2.6 Tlvresult . . . 1017
8.1061 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference . . . 1017
8.1061.1 Detailed Description . . . 1017
8.1061.2 Field Documentation . . . 1017
8.1061.2.1 manString . . . 1017
8.1061.2.2 modelString . . . 1017
8.1061.2.3 plasmaIDString . . . 1017
8.1061.2.4 swVerString . . . 1017
8.1061.2.5 Tlvresult . . . 1017
8.1062 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference . . . 1017
8.1062.1 Detailed Description . . . 1018
8.1062.2 Field Documentation . . . 1018
8.1062.2.1 nameString . . . 1018
8.1062.2.2 Tlvresult . . . 1018
8.1062.2.3 versionString . . . 1018
8.1063 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference . . . 1018
8.1063.1 Detailed Description . . . 1018
8.1063.2 Field Documentation . . . 1018
8.1063.2.1 meidString . . . 1018
8.1063.2.2 Tlvresult . . . 1018
8.1064 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 1018
8.1064.1 Detailed Description . . . 1019
8.1064.2 Field Documentation . . . 1019
8.1064.2.1 Tlvresult . . . 1019
8.1065 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 1019
8.1065.1 Detailed Description . . . 1019
8.1065.2 Field Documentation . . . 1019
8.1065.2.1 Tlvresult . . . 1019
8.1066 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 1019
8.1066.1 Detailed Description . . . 1019
8.1066.2 Field Documentation . . . 1020
8.1066.2.1 Tlvresult . . . 1020
8.1067 unpack_dms_SLQSUIMGetState_t Struct Reference . . . 1020
8.1067.1 Detailed Description . . . 1020
8.1067.2 Field Documentation . . . 1020
8.1067.2.1 state . . . 1020
8.1067.2.2 Tlvresult . . . 1020
8.1068 unpack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 1020
8.1068.1 Detailed Description . . . 1020
8.1068.2 Field Documentation . . . 1021

8.1068.2.1 facilityState . . . 1021
8.1068.2.2 TlvResult . . . 1021
8.1068.2.3 unblockRetriesLeft . . . 1021
8.1068.2.4 verifyRetriesLeft . . . 1021
8.1069 unpack_dms_UIMGetICCID_t Struct Reference . . . 1021
8.1069.1 Detailed Description . . . 1021
8.1069.2 Field Documentation . . . 1021
8.1069.2.1 String . . . 1022
8.1069.2.2 stringSize . . . 1022
8.1069.2.3 TlvResult . . . 1022
8.1070 unpack_dms_UIMGetPINStatus_t Struct Reference . . . 1022
8.1070.1 Detailed Description . . . 1022
8.1070.2 Field Documentation . . . 1023
8.1070.2.1 p1Status . . . 1023
8.1070.2.2 p1UnblockRetriesLeft . . . 1023
8.1070.2.3 p1VerifyRetriesLeft . . . 1023
8.1070.2.4 p2Status . . . 1023
8.1070.2.5 p2UnblockRetriesLeft . . . 1023
8.1070.2.6 p2VerifyRetriesLeft . . . 1023
8.1070.2.7 TlvResult . . . 1023
8.1071 unpack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 1023
8.1071.1 Detailed Description . . . 1023
8.1071.2 Field Documentation . . . 1024
8.1071.2.1 TlvResult . . . 1024
8.1071.2.2 verifyRetriesLeft . . . 1024
8.1072 unpack_dms_UIMSetPINProtection_t Struct Reference . . . 1024
8.1072.1 Detailed Description . . . 1024
8.1072.2 Field Documentation . . . 1024
8.1072.2.1 TlvResult . . . 1024
8.1072.2.2 unblockRetriesLeft . . . 1024
8.1072.2.3 verifyRetriesLeft . . . 1024
8.1073 unpack_dms_UIMUnblockControlKey_t Struct Reference . . . 1024
8.1073.1 Detailed Description . . . 1025
8.1073.2 Field Documentation . . . 1025
8.1073.2.1 TlvResult . . . 1025
8.1073.2.2 unblockRetriesLeft . . . 1025
8.1074 unpack_fms_GetImagesPreference_t Struct Reference . . . 1025
8.1074.1 Detailed Description . . . 1025
8.1074.2 Field Documentation . . . 1025
8.1074.2.1 ImageListSize . . . 1025

8.1074.2.2 pImageList . . . 1025
8.1074.2.3 Tlvresult . . . 1025
8.1075 Unpack_fms_GetStoredImages_t Struct Reference . . . 1026
8.1075.1 Detailed Description . . . 1026
8.1075.2 Field Documentation . . . 1026
8.1075.2.1 imageList . . . 1026
8.1075.2.2 imagelistSize . . . 1026
8.1075.2.3 Tlvresult . . . 1026
8.1076 Unpack_fms_SetImagesPreference_t Struct Reference . . . 1026
8.1076.1 Detailed Description . . . 1026
8.1076.2 Field Documentation . . . 1027
8.1076.2.1 ImageTypes . . . 1027
8.1076.2.2 imageTypesSize . . . 1027
8.1076.2.3 Tlvresult . . . 1027
8.1077 Unpack_loc_BestAvailPos_Ind_t Struct Reference . . . 1027
8.1077.1 Detailed Description . . . 1028
8.1077.2 Field Documentation . . . 1031
8.1077.2.1 pAltitudeWrtEllipsoid . . . 1031
8.1077.2.2 pAltitudeWrtMeanSeaLevel . . . 1031
8.1077.2.3 pGpsTime . . . 1031
8.1077.2.4 pHeading . . . 1031
8.1077.2.5 pHeadingUnc . . . 1031
8.1077.2.6 pHorCirConf . . . 1031
8.1077.2.7 pHorEllpConf . . . 1031
8.1077.2.8 pHorReliability . . . 1032
8.1077.2.9 pHorUncCircular . . . 1032
8.1077.2.10 pHorUncEllipseOrientAzimuth . . . 1032
8.1077.2.11 pHorUncEllipseSemiMajor . . . 1032
8.1077.2.12 pHorUncEllipseSemiMinor . . . 1032
8.1077.2.13 pLatitude . . . 1032
8.1077.2.14 pLongitude . . . 1032
8.1077.2.15 pMagneticDeviation . . . 1032
8.1077.2.16 pPrecisionDilution . . . 1032
8.1077.2.17 pSensorDataUsage . . . 1032
8.1077.2.18 pSpeedHorizontal . . . 1032
8.1077.2.19 pSpeedUnc . . . 1032
8.1077.2.20 pSpeedVertical . . . 1032
8.1077.2.21 pSpeedVerticalUnc . . . 1032
8.1077.2.22 pSvUsedforFix . . . 1032
8.1077.2.23 pTechnologyMask . . . 1032

8.1077.2.24 pTimeSrc . . . 1032
8.1077.2.25 pTimestampUtc . . . 1032
8.1077.2.26 pTimeUnc . . . 1032
8.1077.2.27 pVertConfidence . . . 1032
8.1077.2.28 pVertReliability . . . 1032
8.1077.2.29 pVertUnc . . . 1032
8.1077.2.30 pXid . . . 1032
8.1077.2.31 status . . . 1032
8.1077.2.32 tlvresult . . . 1032
8.1078 qmi_loc_Delete_Assist_Data_t Struct Reference . . . 1032
8.1078.1 Detailed Description . . . 1033
8.1078.2 Field Documentation . . . 1033
8.1078.2.1 tlvresult . . . 1033
8.1079 qmi_loc_DeleteAssistData_Ind_t Struct Reference . . . 1033
8.1079.1 Detailed Description . . . 1033
8.1079.2 Field Documentation . . . 1033
8.1079.2.1 status . . . 1034
8.1079.2.2 tlvresult . . . 1034
8.1080 qmi_loc_EngineState_Ind_t Struct Reference . . . 1034
8.1080.1 Detailed Description . . . 1034
8.1080.2 Field Documentation . . . 1034
8.1080.2.1 engineState . . . 1034
8.1080.2.2 tlvresult . . . 1034
8.1081 qmi_loc_EventRegister_t Struct Reference . . . 1034
8.1081.1 Detailed Description . . . 1034
8.1081.2 Field Documentation . . . 1034
8.1081.2.1 tlvresult . . . 1035
8.1082 qmi_loc_GnssSvInfo_Ind_t Struct Reference . . . 1035
8.1082.1 Detailed Description . . . 1035
8.1082.2 Field Documentation . . . 1035
8.1082.2.1 altitudeAssumed . . . 1035
8.1082.2.2 pSatelliteInfo . . . 1035
8.1082.2.3 tlvresult . . . 1035
8.1083 qmi_loc_PositionRpt_Ind_t Struct Reference . . . 1035
8.1083.1 Detailed Description . . . 1036
8.1083.2 Field Documentation . . . 1040
8.1083.2.1 pAltitudeAssumed . . . 1040
8.1083.2.2 pAltitudeWrtEllipsoid . . . 1040
8.1083.2.3 pAltitudeWrtMeanSeaLevel . . . 1040
8.1083.2.4 pFixId . . . 1040

8.1083.2.5 pGpsTime . . . 1040
8.1083.2.6 pHeading . . . 1041
8.1083.2.7 pHeadingUnc . . . 1041
8.1083.2.8 pHorConfidence . . . 1041
8.1083.2.9 pHorReliability . . . 1041
8.1083.2.10 pHorUncCircular . . . 1041
8.1083.2.11 pHorUncEllipseOrientAzimuth . . . 1041
8.1083.2.12 pHorUncEllipseSemiMajor . . . 1041
8.1083.2.13 pHorUncEllipseSemiMinor . . . 1041
8.1083.2.14 pLatitude . . . 1041
8.1083.2.15 pLeapSeconds . . . 1041
8.1083.2.16 pLongitude . . . 1041
8.1083.2.17 pMagneticDeviation . . . 1041
8.1083.2.18 pPrecisionDilution . . . 1041
8.1083.2.19 pSensorDataUsage . . . 1041
8.1083.2.20 pSpeedHorizontal . . . 1041
8.1083.2.21 pSpeedUnc . . . 1041
8.1083.2.22 pSpeedVertical . . . 1041
8.1083.2.23 pSvUsedforFix . . . 1041
8.1083.2.24 pTechnologyMask . . . 1041
8.1083.2.25 pTimeSrc . . . 1041
8.1083.2.26 pTimestampUtc . . . 1041
8.1083.2.27 pTimeUnc . . . 1041
8.1083.2.28 pVertConfidence . . . 1041
8.1083.2.29 pVertReliability . . . 1041
8.1083.2.30 pVertUnc . . . 1041
8.1083.2.31 sessionId . . . 1041
8.1083.2.32 sessionStatus . . . 1041
8.1083.2.33 tlvresult . . . 1041
8.1084 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference . . . 1042
8.1084.1 Detailed Description . . . 1042
8.1084.2 Field Documentation . . . 1042
8.1084.2.1 status . . . 1042
8.1084.2.2 Tlvresult . . . 1042
8.1085 unpack_loc_SetExtPowerState_t Struct Reference . . . 1042
8.1085.1 Detailed Description . . . 1042
8.1085.2 Field Documentation . . . 1043
8.1085.2.1 Tlvresult . . . 1043
8.1086 unpack_loc_SetOperationMode_Ind_t Struct Reference . . . 1043
8.1086.1 Detailed Description . . . 1043

8.1086.2 Field Documentation . . . 1043
8.1086.2.1 status . . . 1043
8.1086.2.2 TlvResult . . . 1043
8.1087 Qmi::Unpack_loc_SetOperationMode_t Struct Reference . . . 1043
8.1087.1 Detailed Description . . . 1043
8.1087.2 Field Documentation . . . 1044
8.1087.2.1 TlvResult . . . 1044
8.1088 Qmi::Unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 1044
8.1088.1 Detailed Description . . . 1044
8.1088.2 Field Documentation . . . 1044
8.1088.2.1 TlvResult . . . 1044
8.1089 Qmi::Unpack_loc_Start_t Struct Reference . . . 1044
8.1089.1 Detailed Description . . . 1044
8.1089.2 Field Documentation . . . 1044
8.1089.2.1 TlvResult . . . 1044
8.1090 Qmi::Unpack_loc_Stop_t Struct Reference . . . 1044
8.1090.1 Detailed Description . . . 1045
8.1090.2 Field Documentation . . . 1045
8.1090.2.1 TlvResult . . . 1045
8.1091 Qmi::Unpack_nas_GetCDMANetworkParameters_t Struct Reference . . . 1045
8.1091.1 Detailed Description . . . 1045
8.1091.2 Field Documentation . . . 1045
8.1091.2.1 Application . . . 1045
8.1091.2.2 Broadcast . . . 1046
8.1091.2.3 CustomSCP . . . 1046
8.1091.2.4 ForceRev0 . . . 1046
8.1091.2.5 Protocol . . . 1046
8.1091.2.6 RegForeignNID . . . 1046
8.1091.2.7 RegForeignSID . . . 1046
8.1091.2.8 RegHomeSID . . . 1046
8.1091.2.9 Roaming . . . 1046
8.1091.2.10 SCI . . . 1046
8.1091.2.11 SCM . . . 1046
8.1092 Qmi::Unpack_nas_GetHomeNetwork3GPP2_t Struct Reference . . . 1046
8.1092.1 Detailed Description . . . 1046
8.1092.2 Field Documentation . . . 1047
8.1092.2.1 nameSize . . . 1047
8.1092.2.2 pMCC . . . 1047
8.1092.2.3 pMNC . . . 1047
8.1092.2.4 pName . . . 1048

8.1092.2.5 pNID . . . 1048
8.1092.2.6 pNw2DescDisp . . . 1048
8.1092.2.7 pNw2DescEnc . . . 1048
8.1092.2.8 pNw2DescLen . . . 1048
8.1092.2.9 pNw2MCC . . . 1048
8.1092.2.10 pNw2MNC . . . 1048
8.1092.2.11 pNw2Name . . . 1048
8.1092.2.12 pSID . . . 1048
8.1093 npack_nas_GetHomeNetwork_t Struct Reference . . . 1048
8.1093.1 Detailed Description . . . 1048
8.1093.2 Field Documentation . . . 1048
8.1093.2.1 mcc . . . 1048
8.1093.2.2 mnc . . . 1048
8.1093.2.3 name . . . 1048
8.1093.2.4 nid . . . 1048
8.1093.2.5 sid . . . 1048
8.1094 npack_nas_GetNetworkPreference_t Struct Reference . . . 1048
8.1094.1 Detailed Description . . . 1049
8.1094.2 Field Documentation . . . 1049
8.1094.2.1 ActiveTechPref . . . 1049
8.1094.2.2 Duration . . . 1049
8.1094.2.3 PersistentTechPref . . . 1049
8.1094.2.4 Tlvresult . . . 1049
8.1095 npack_nas_GetRFInfo_t Struct Reference . . . 1049
8.1095.1 Detailed Description . . . 1050
8.1095.2 Field Documentation . . . 1050
8.1095.2.1 instancesSize . . . 1050
8.1095.2.2 RFBandInfoElements . . . 1050
8.1096 npack_nas_GetServingNetwork_t Struct Reference . . . 1050
8.1096.1 Detailed Description . . . 1050
8.1096.2 Field Documentation . . . 1051
8.1096.2.1 CSDomain . . . 1051
8.1096.2.2 DataCaps . . . 1051
8.1096.2.3 DataCapsLen . . . 1051
8.1096.2.4 MCC . . . 1051
8.1096.2.5 MNC . . . 1051
8.1096.2.6 Name . . . 1051
8.1096.2.7 nameSize . . . 1051
8.1096.2.8 PSDomain . . . 1051
8.1096.2.9 RadioIfaces . . . 1051

8.1096.2.1 RadioIfacesSize . . . 1051
8.1096.2.1 RAN . . . 1051
8.1096.2.1 RegistrationState . . . 1051
8.1096.2.1 Roaming . . . 1051
8.1097 unpack_nas_GetServingNetworkCapabilities_t Struct Reference . . . 1051
8.1097.1 Detailed Description . . . 1051
8.1097.2 Field Documentation . . . 1051
8.1097.2.1 DataCaps . . . 1051
8.1097.2.2 DataCapsLen . . . 1051
8.1098 unpack_nas_GetSignalStrengths_t Struct Reference . . . 1051
8.1098.1 Detailed Description . . . 1052
8.1098.2 Field Documentation . . . 1052
8.1098.2.1 len . . . 1052
8.1098.2.2 radio . . . 1052
8.1098.2.3 rssi . . . 1052
8.1099 unpack_nas_PerformNetworkScan_t Struct Reference . . . 1052
8.1099.1 Detailed Description . . . 1052
8.1099.2 Field Documentation . . . 1052
8.1099.2.1 p3GppNetworkInfoInstances . . . 1052
8.1099.2.2 p3GppNetworkInstanceSize . . . 1052
8.1099.2.3 pPCSInstance . . . 1052
8.1099.2.4 pPCSInstanceSize . . . 1052
8.1099.2.5 pRATINstance . . . 1052
8.1099.2.6 pRATInstanceSize . . . 1052
8.1099.2.7 pScanResult . . . 1052
8.1100 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference . . . 1053
8.1100.1 Detailed Description . . . 1053
8.1100.2 Field Documentation . . . 1053
8.1100.2.1 dataCaps . . . 1053
8.1100.2.2 dataCapsSize . . . 1053
8.1101 unpack_nas_SetEventReportInd_t Struct Reference . . . 1053
8.1101.1 Detailed Description . . . 1053
8.1101.2 Field Documentation . . . 1053
8.1101.2.1 RFTlv . . . 1053
8.1101.2.2 RRTlv . . . 1053
8.1101.2.3 SLQSSSTlv . . . 1053
8.1101.2.4 SSTlv . . . 1053
8.1102 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference . . . 1053
8.1102.1 Detailed Description . . . 1054
8.1102.2 Field Documentation . . . 1054

8.1102.2.1 sPhyCaAggPcellInfo . . . 1054
8.1102.2.2 sPhyCaAggScellDlBw . . . 1054
8.1102.2.3 sPhyCaAggScellIndex . . . 1054
8.1102.2.4 sPhyCaAggScellIndType . . . 1054
8.1102.2.5 sPhyCaAggScellInfo . . . 1054
8.1103 qmi_nas_SetNetworkPreference_t Struct Reference . . . 1054
8.1103.1 Detailed Description . . . 1054
8.1103.2 Field Documentation . . . 1055
8.1103.2.1 TlvResult . . . 1055
8.1104 qmi_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . 1055
8.1104.1 Detailed Description . . . 1055
8.1104.2 Field Documentation . . . 1056
8.1104.2.1 roaming . . . 1056
8.1105 qmi_nas_SetServingSystemCallback_ind_t Struct Reference . . . 1056
8.1105.1 Detailed Description . . . 1056
8.1105.2 Field Documentation . . . 1056
8.1105.2.1 SSInfo . . . 1056
8.1105.2.2 TlvResult . . . 1056
8.1106 qmi_nas_SLQSGetErrorRate_t Struct Reference . . . 1056
8.1106.1 Detailed Description . . . 1056
8.1106.2 Field Documentation . . . 1057
8.1106.2.1 pCDMAFrameErrRate . . . 1057
8.1106.2.2 pGSMBER . . . 1057
8.1106.2.3 pHDRPackErrRate . . . 1057
8.1106.2.4 pWCDMABER . . . 1057
8.1107 qmi_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . 1057
8.1107.1 Detailed Description . . . 1057
8.1107.2 Field Documentation . . . 1057
8.1107.2.1 LTECphyCaInfo . . . 1057
8.1107.2.2 TlvResult . . . 1057
8.1108 qmi_nas_SLQSGetNetworkTime_t Struct Reference . . . 1058
8.1108.1 Detailed Description . . . 1058
8.1108.2 Field Documentation . . . 1058
8.1108.2.1 p3GPP2TimeInfo . . . 1058
8.1108.2.2 p3GPPTimeInfo . . . 1058
8.1109 qmi_nas_SLQSGetOperatorNameData_t Struct Reference . . . 1058
8.1109.1 Detailed Description . . . 1058
8.1109.2 Field Documentation . . . 1059
8.1109.2.1 pNITZInformation . . . 1059
8.1109.2.2 pOperatorNameString . . . 1059

8.1109.2.3 pOperatorPLMNList . . . 1059
8.1109.2.4 pPLMNNetworkName . . . 1059
8.1109.2.5 pSrvcProviderName . . . 1059
8.1110 unpack_nas_SLQSGetPLMNName_t Struct Reference . . . 1059
8.1110.1 Field Documentation . . . 1059
8.1110.1.1 longName . . . 1059
8.1110.1.2 longNameCI . . . 1059
8.1110.1.3 longNameEn . . . 1059
8.1110.1.4 longNameLen . . . 1059
8.1110.1.5 longNameSB . . . 1059
8.1110.1.6 shortName . . . 1059
8.1110.1.7 shortNameCI . . . 1060
8.1110.1.8 shortNameEn . . . 1060
8.1110.1.9 shortNameLen . . . 1060
8.1110.1.10 shortNameSB . . . 1060
8.1110.1.11 spn . . . 1060
8.1110.1.12 spnEncoding . . . 1060
8.1110.1.13 spnLength . . . 1060
8.1111 unpack_nas_SLQSGetServingSystem_t Struct Reference . . . 1060
8.1111.1 Detailed Description . . . 1060
8.1111.2 Field Documentation . . . 1061
8.1111.2.1 BasestationID . . . 1061
8.1111.2.2 BasestationLatitude . . . 1061
8.1111.2.3 BasestationLongitude . . . 1061
8.1111.2.4 CallBarStatus . . . 1061
8.1111.2.5 CDMA_P_Rev . . . 1061
8.1111.2.6 CDMASystemInfoExt . . . 1061
8.1111.2.7 CellID . . . 1061
8.1111.2.8 ConcSvcInfo . . . 1061
8.1111.2.9 CurrentPLMN . . . 1061
8.1111.2.10 DataSrvCapabilities . . . 1061
8.1111.2.11 DefaultRoamInd . . . 1061
8.1111.2.12 DetailedSvcInfo . . . 1061
8.1111.2.13 DTMInd . . . 1061
8.1111.2.14 hpp2TimeZone . . . 1062
8.1111.2.15 HppNetworkDSTAdjustment . . . 1062
8.1111.2.16 HppTimeZone . . . 1062
8.1111.2.17 HdrPersonality . . . 1062
8.1111.2.18 lac . . . 1062
8.1111.2.19 NetworkID . . . 1062

8.1111.2.20 PRLInd . . . 1062
8.1111.2.21 RoamIndicatorVal . . . 1062
8.1111.2.22 RoamingIndicatorList . . . 1062
8.1111.2.23 ServingSystem . . . 1062
8.1111.2.24 SystemID . . . 1062
8.1111.2.25 TrackAreaCode . . . 1062
8.1112 qmi_nas_SLQSGetSignalStrength_t Struct Reference . . . 1062
8.1112.1 Detailed Description . . . 1062
8.1112.2 Field Documentation . . . 1063
8.1112.2.1 ecioList . . . 1063
8.1112.2.2 ecioListLen . . . 1063
8.1112.2.3 errorRateList . . . 1063
8.1112.2.4 errorRateListLen . . . 1063
8.1112.2.5 io . . . 1063
8.1112.2.6 rsrp . . . 1063
8.1112.2.7 snr . . . 1063
8.1112.2.8 srqInfo . . . 1063
8.1112.2.9 rxSignalStrengthList . . . 1063
8.1112.2.10 rxSignalStrengthListLen . . . 1063
8.1112.2.11 signalStrengthReqMask . . . 1063
8.1112.2.12 snr . . . 1063
8.1113 qmi_nas_SLQSGetSysInfo_t Struct Reference . . . 1063
8.1113.1 Detailed Description . . . 1064
8.1113.2 Field Documentation . . . 1065
8.1113.2.1 pAddCDMASysInfo . . . 1065
8.1113.2.2 pAddGSMSysInfo . . . 1065
8.1113.2.3 pAddHDRSysInfo . . . 1065
8.1113.2.4 pAddLTESysInfo . . . 1065
8.1113.2.5 pAddWCDMASysInfo . . . 1065
8.1113.2.6 pCDMASrvStatusInfo . . . 1065
8.1113.2.7 pCDMASysInfo . . . 1065
8.1113.2.8 pGSMCallBarringSysInfo . . . 1065
8.1113.2.9 pGSMCipherDomainSysInfo . . . 1065
8.1113.2.10 pGSMSrvStatusInfo . . . 1065
8.1113.2.11 pGSMSysInfo . . . 1065
8.1113.2.12 pHDRSrvStatusInfo . . . 1065
8.1113.2.13 pHDRSysInfo . . . 1065
8.1113.2.14 pLTESrvStatusInfo . . . 1065
8.1113.2.15 pLTESysInfo . . . 1066
8.1113.2.16 pLTEVoiceSupportSysInfo . . . 1066

8.1113.2.17 pWCDMACallBarringSysInfo . . . 1066
8.1113.2.18 pWCDMACipherDomainSysInfo . . . 1066
8.1113.2.19 pWCDMASrvStatusInfo . . . 1066
8.1113.2.20 pWCDMASysInfo . . . 1066
8.1114 qmi_nas_SLQSGetSysSelectionPref_t Struct Reference . . . 1066
8.1114.1 Detailed Description . . . 1066
8.1114.2 Field Documentation . . . 1069
8.1114.2.1 pBandPref . . . 1069
8.1114.2.2 pEmerMode . . . 1069
8.1114.2.3 pGWAcqOrderPref . . . 1069
8.1114.2.4 pLTEBandPref . . . 1069
8.1114.2.5 pModePref . . . 1069
8.1114.2.6 pNetSelPref . . . 1069
8.1114.2.7 pPRLPref . . . 1069
8.1114.2.8 pRoamPref . . . 1069
8.1114.2.9 pSrvDomainPref . . . 1069
8.1115 qmi_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 1069
8.1115.1 Detailed Description . . . 1070
8.1115.2 Field Documentation . . . 1070
8.1115.2.1 pCDMAChannel . . . 1070
8.1115.2.2 pDirNum . . . 1070
8.1115.2.3 pHomeSIDNID . . . 1070
8.1115.2.4 pMinBasedIMSI . . . 1070
8.1115.2.5 pNAMNameInfo . . . 1070
8.1115.2.6 pTrueIMSI . . . 1070
8.1116 qmi_nas_SLQSNasGetCellLocationInfo_t Struct Reference . . . 1070
8.1116.1 Detailed Description . . . 1071
8.1116.2 Field Documentation . . . 1071
8.1116.2.1 pCDMAInfo . . . 1071
8.1116.2.2 pGERANInfo . . . 1071
8.1116.2.3 pLTEInfoInterfreq . . . 1071
8.1116.2.4 pLTEInfoIntrafreq . . . 1071
8.1116.2.5 pLTEInfoNeighboringGSM . . . 1071
8.1116.2.6 pLTEInfoNeighboringWCDMA . . . 1071
8.1116.2.7 pUMTSCellID . . . 1071
8.1116.2.8 pUMTSInfo . . . 1071
8.1116.2.9 pWCDMAInfoLTENeighborCell . . . 1072
8.1117 qmi_nas_SLQSNasGetHDRColorCode_t Struct Reference . . . 1072
8.1117.1 Detailed Description . . . 1072
8.1117.2 Field Documentation . . . 1072

8.1117.2.1 pColorCode . . . 1072
8.1118 qmi_nas_SLQSNasGetSigInfo_t Struct Reference . . . 1072
8.1118.1 Detailed Description . . . 1072
8.1118.2 Field Documentation . . . 1072
8.1118.2.1 CDMASSInfo . . . 1072
8.1118.2.2 GSMSSInfo . . . 1072
8.1118.2.3 HDRSSInfo . . . 1073
8.1118.2.4 LTESSInfo . . . 1073
8.1118.2.5 WCDMASSInfo . . . 1073
8.1119 qmi_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 1073
8.1119.1 Detailed Description . . . 1073
8.1119.2 Field Documentation . . . 1073
8.1119.2.1 pRXChain0Info . . . 1073
8.1119.2.2 pRXChain1Info . . . 1073
8.1119.2.3 pTXInfo . . . 1073
8.1120 qmi_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . 1073
8.1120.1 Detailed Description . . . 1073
8.1120.2 Field Documentation . . . 1074
8.1120.2.1 pDayltSavAdj . . . 1074
8.1120.2.2 pRadioInterface . . . 1074
8.1120.2.3 pTimeZone . . . 1074
8.1120.2.4 universalTime . . . 1074
8.1121 qmi_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . 1074
8.1121.1 Detailed Description . . . 1074
8.1121.2 Field Documentation . . . 1075
8.1121.2.1 pCDMASigInfo . . . 1075
8.1121.2.2 pGSMSigInfo . . . 1075
8.1121.2.3 pHDRSigInfo . . . 1075
8.1121.2.4 pLTESigInfo . . . 1075
8.1121.2.5 pRscp . . . 1075
8.1121.2.6 pTDSCDMASigInfoExt . . . 1075
8.1121.2.7 pWCDMASigInfo . . . 1075
8.1122 qmi_nas_SLQSNASSwiGetChannelLock_t Struct Reference . . . 1075
8.1122.1 Detailed Description . . . 1075
8.1122.2 Field Documentation . . . 1075
8.1122.2.1 pLteEARFCN . . . 1075
8.1122.2.2 pLtePCI . . . 1076
8.1122.2.3 pWcdmaUARFCN . . . 1076
8.1123 qmi_nas_SLQSNasSwiModemStatus_t Struct Reference . . . 1076
8.1123.1 Detailed Description . . . 1076

8.1123.2 Field Documentation . . . 1076
8.1123.2.1 commonInfo . . . 1076
8.1123.2.2 pLTEInfo . . . 1076
8.1124 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . 1076
8.1124.1 Detailed Description . . . 1076
8.1124.2 Field Documentation . . . 1076
8.1124.2.1 Info . . . 1076
8.1124.2.2 Tlvresult . . . 1077
8.1125 unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference . . . 1077
8.1125.1 Detailed Description . . . 1077
8.1125.2 Field Documentation . . . 1077
8.1125.2.1 t3396_apn . . . 1077
8.1125.2.2 t3396_plmn_id . . . 1077
8.1125.2.3 t3396_val . . . 1077
8.1126 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . 1077
8.1126.1 Detailed Description . . . 1077
8.1126.2 Field Documentation . . . 1078
8.1126.2.1 Info . . . 1078
8.1126.2.2 Tlvresult . . . 1078
8.1127 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference . . . 1078
8.1127.1 Detailed Description . . . 1078
8.1127.2 Field Documentation . . . 1078
8.1127.2.1 pCurrentPersonality . . . 1078
8.1127.2.2 pPersonalityListLength . . . 1078
8.1127.2.3 pProtocolSubtypeElement . . . 1078
8.1128 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference . . . 1078
8.1128.1 Detailed Description . . . 1078
8.1128.2 Field Documentation . . . 1079
8.1128.2.1 pAppSubType . . . 1079
8.1128.2.2 pCurrentPrsnlty . . . 1079
8.1128.2.3 pPersonalityListLength . . . 1079
8.1128.2.4 pProtoSubTypElmnt . . . 1079
8.1129 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference . . . 1079
8.1129.1 Detailed Description . . . 1079
8.1129.2 Field Documentation . . . 1079
8.1129.2.1 pDRCParams . . . 1080
8.1129.2.2 pPilotSetData . . . 1080
8.1129.2.3 pUATI . . . 1080
8.1130 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . 1080
8.1130.1 Detailed Description . . . 1080

8.1130.2 Field Documentation . . . 1080
8.1130.2.1 CQIValueCW0 . . . 1080
8.1130.2.2 CQIValueCW1 . . . 1080
8.1130.2.3 ValidityCW0 . . . 1080
8.1130.2.4 ValidityCW1 . . . 1080
8.1131 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference . . . 1080
8.1131.1 Detailed Description . . . 1081
8.1131.2 Field Documentation . . . 1081
8.1131.2.1 pSccRxInfo . . . 1081
8.1132 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference . . . 1081
8.1132.1 Detailed Description . . . 1081
8.1132.2 Field Documentation . . . 1081
8.1132.2.1 pDataStatusDetail . . . 1081
8.1132.2.2 pDeviceConfigDetail . . . 1081
8.1132.2.3 pNetworkStat1x . . . 1081
8.1132.2.4 pNetworkStatEVDO . . . 1081
8.1132.2.5 pObjectVer . . . 1082
8.1133 unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . 1082
8.1133.1 Detailed Description . . . 1082
8.1133.2 Field Documentation . . . 1084
8.1133.2.1 pAddCDMASysInfo . . . 1084
8.1133.2.2 pAddGSMSysInfo . . . 1084
8.1133.2.3 pAddHDRSysInfo . . . 1084
8.1133.2.4 pAddLTESysInfo . . . 1084
8.1133.2.5 pAddWCDMASysInfo . . . 1084
8.1133.2.6 pCDMASrvStatusInfo . . . 1084
8.1133.2.7 pCDMASysInfo . . . 1084
8.1133.2.8 pGSMCallBarringSysInfo . . . 1084
8.1133.2.9 pGSMCipherDomainSysInfo . . . 1084
8.1133.2.10 pGSMSrvStatusInfo . . . 1084
8.1133.2.11 pGSMSysInfo . . . 1084
8.1133.2.12 pHDRSrvStatusInfo . . . 1084
8.1133.2.13 pHDRSysInfo . . . 1084
8.1133.2.14 pLTESrvStatusInfo . . . 1084
8.1133.2.15 pLTESysInfo . . . 1084
8.1133.2.16 pLTEVoiceSupportSysInfo . . . 1084
8.1133.2.17 pSysInfoNoChange . . . 1084
8.1133.2.18 pWCDMACallBarringSysInfo . . . 1084
8.1133.2.19 pWCDMACipherDomainSysInfo . . . 1084
8.1133.2.20 pWCDMASrvStatusInfo . . . 1084

8.1133.2.21 pWCDMASysInfo . . . 1084
8.1134 unpack_omaDmConfigTlv_t Struct Reference . . . 1084
8.1134.1 Detailed Description . . . 1085
8.1134.2 Field Documentation . . . 1085
8.1134.2.1 alertmsg . . . 1085
8.1134.2.2 alertmsglength . . . 1085
8.1134.2.3 state . . . 1085
8.1134.2.4 userInputReq . . . 1085
8.1134.2.5 userInputTimeout . . . 1085
8.1135 unpack_omaDmFotaTlv_t Struct Reference . . . 1085
8.1135.1 Detailed Description . . . 1086
8.1135.2 Field Documentation . . . 1087
8.1135.2.1 description . . . 1087
8.1135.2.2 descriptionlength . . . 1087
8.1135.2.3 fwdloadsize . . . 1087
8.1135.2.4 fwloadComplete . . . 1087
8.1135.2.5 namelength . . . 1087
8.1135.2.6 package_name . . . 1087
8.1135.2.7 sessionType . . . 1087
8.1135.2.8 severity . . . 1087
8.1135.2.9 state . . . 1087
8.1135.2.10 updateCompleteStatus . . . 1087
8.1135.2.11 userInputReq . . . 1087
8.1135.2.12 userInputTimeout . . . 1087
8.1135.2.13 version . . . 1087
8.1135.2.14 versionlength . . . 1087
8.1136 unpack_omaDmNotificationsTlv_t Struct Reference . . . 1087
8.1136.1 Field Documentation . . . 1088
8.1136.1.1 notification . . . 1088
8.1136.1.2 sessionStatus . . . 1088
8.1137 unpack_pds_GetPDSDefaults_t Struct Reference . . . 1088
8.1137.1 Detailed Description . . . 1088
8.1137.2 Field Documentation . . . 1088
8.1137.2.1 pAccuracy . . . 1088
8.1137.2.2 pInterval . . . 1088
8.1137.2.3 pOperation . . . 1088
8.1137.2.4 pTimeout . . . 1088
8.1138 unpack_pds_GetPDSState_t Struct Reference . . . 1088
8.1138.1 Detailed Description . . . 1089
8.1138.2 Field Documentation . . . 1089

8.1138.2.1 pEnabledStatus . . . 1089
8.1138.2.2 pTrackingStatus . . . 1089
8.1139 unpack_pds_GetPortAutomaticTracking_t Struct Reference . . . 1089
8.1139.1 Detailed Description . . . 1089
8.1139.2 Field Documentation . . . 1089
8.1139.2.1 pbAuto . . . 1089
8.1140 unpack_qmi_t Struct Reference . . . 1089
8.1140.1 Detailed Description . . . 1090
8.1140.2 Field Documentation . . . 1090
8.1140.2.1 msgid . . . 1090
8.1140.2.2 type . . . 1090
8.1140.2.3 xid . . . 1090
8.1141 unpack_qos_dataRate_t Struct Reference . . . 1090
8.1141.1 Detailed Description . . . 1090
8.1141.2 Field Documentation . . . 1090
8.1141.2.1 dataRateMax . . . 1090
8.1141.2.2 guaranteedRate . . . 1090
8.1142 unpack_qos_IPv4Addr_t Struct Reference . . . 1090
8.1142.1 Detailed Description . . . 1091
8.1142.2 Field Documentation . . . 1091
8.1142.2.1 addr . . . 1091
8.1142.2.2 subnetMask . . . 1091
8.1143 unpack_qos_IPv6Addr_t Struct Reference . . . 1091
8.1143.1 Detailed Description . . . 1091
8.1143.2 Field Documentation . . . 1091
8.1143.2.1 addr . . . 1091
8.1143.2.2 prefixLen . . . 1091
8.1144 unpack_qos_IPv6TrafCls_t Struct Reference . . . 1091
8.1144.1 Detailed Description . . . 1091
8.1144.2 Field Documentation . . . 1092
8.1144.2.1 mask . . . 1092
8.1144.2.2 val . . . 1092
8.1145 unpack_qos_pktErrRate_t Struct Reference . . . 1092
8.1145.1 Detailed Description . . . 1092
8.1145.2 Field Documentation . . . 1092
8.1145.2.1 exponent . . . 1092
8.1145.2.2 multiplier . . . 1092
8.1146 unpack_qos_Port_t Struct Reference . . . 1092
8.1146.1 Detailed Description . . . 1092
8.1146.2 Field Documentation . . . 1093

8.1146.2.1 port . . . 1093
8.1146.2.2 range . . . 1093
8.1147 unpack_qos_QosFlowInfo_t Struct Reference . . . 1093
8.1147.1 Detailed Description . . . 1093
8.1147.2 Field Documentation . . . 1094
8.1147.2.1 BearerID . . . 1094
8.1147.2.2 is_RxQFlowGranted_Available . . . 1094
8.1147.2.3 is_TxQFlowGranted_Available . . . 1094
8.1147.2.4 NumRxFilters . . . 1094
8.1147.2.5 NumTxFilters . . . 1094
8.1147.2.6 QFlowState . . . 1094
8.1147.2.7 RxQFilter . . . 1094
8.1147.2.8 RxQFlowGranted . . . 1094
8.1147.2.9 TxQFilter . . . 1094
8.1147.2.10 TxQFlowGranted . . . 1094
8.1148 unpack_qos_QosFlowInfoState_t Struct Reference . . . 1094
8.1148.1 Detailed Description . . . 1094
8.1148.2 Field Documentation . . . 1095
8.1148.2.1 id . . . 1095
8.1148.2.2 isNewFlow . . . 1095
8.1148.2.3 state . . . 1095
8.1149 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . 1095
8.1149.1 Detailed Description . . . 1095
8.1149.2 Field Documentation . . . 1095
8.1149.2.1 NWQoSStatus . . . 1095
8.1150 unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 1095
8.1150.1 Detailed Description . . . 1096
8.1150.2 Field Documentation . . . 1096
8.1150.2.1 ambr_dl . . . 1096
8.1150.2.2 ambr_dl_ext . . . 1096
8.1150.2.3 ambr_dl_ext2 . . . 1096
8.1150.2.4 ambr_ul . . . 1096
8.1150.2.5 ambr_ul_ext . . . 1096
8.1150.2.6 ambr_ul_ext2 . . . 1096
8.1150.2.7 apnId . . . 1097
8.1151 unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 1097
8.1151.1 Detailed Description . . . 1097
8.1151.2 Field Documentation . . . 1098
8.1151.2.1 apnId . . . 1098
8.1151.2.2 numQosFlow . . . 1098

8.1151.2.3 qosFlow . . . 1098
8.1151.2.4 total_rx_bytes . . . 1098
8.1151.2.5 total_rx_pkt . . . 1098
8.1151.2.6 total_tx_bytes . . . 1098
8.1151.2.7 total_tx_bytes_drp . . . 1098
8.1151.2.8 total_tx_pkt . . . 1098
8.1151.2.9 total_tx_pkt_drp . . . 1098
8.1152 qmi_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . 1098
8.1152.1 Detailed Description . . . 1098
8.1152.2 Field Documentation . . . 1098
8.1152.2.1 NumFlows . . . 1098
8.1152.2.2 QosFlowInfo . . . 1098
8.1153 qmi_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . 1098
8.1153.1 Detailed Description . . . 1099
8.1153.2 Field Documentation . . . 1099
8.1153.2.1 status . . . 1099
8.1154 qmi_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . 1099
8.1154.1 Detailed Description . . . 1099
8.1154.2 Field Documentation . . . 1099
8.1154.2.1 event . . . 1099
8.1155 qmi_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . 1099
8.1155.1 Detailed Description . . . 1099
8.1155.2 Field Documentation . . . 1100
8.1155.2.1 event . . . 1100
8.1155.2.2 id . . . 1100
8.1155.2.3 reason . . . 1100
8.1155.2.4 status . . . 1100
8.1156 qmi_qos_swiQosFilter_t Struct Reference . . . 1101
8.1156.1 Detailed Description . . . 1101
8.1156.2 Field Documentation . . . 1103
8.1156.2.1 EspSpi . . . 1103
8.1156.2.2 id . . . 1103
8.1156.2.3 index . . . 1103
8.1156.2.4 IPv4DstAddr . . . 1103
8.1156.2.5 IPv4SrcAddr . . . 1103
8.1156.2.6 IPv4Tos . . . 1103
8.1156.2.7 IPv6DstAddr . . . 1103
8.1156.2.8 IPv6Label . . . 1103
8.1156.2.9 IPv6SrcAddr . . . 1103
8.1156.2.10 IPv6TrafCls . . . 1103

8.1156.2.11 is_EspSpi_Available . . . 1103
8.1156.2.12 is_Id_Available . . . 1103
8.1156.2.13 is_IPv4DstAddr_Available . . . 1103
8.1156.2.14 is_IPv4SrcAddr_Available . . . 1103
8.1156.2.15 is_IPv4Tos_Available . . . 1103
8.1156.2.16 is_IPv6DstAddr_Available . . . 1103
8.1156.2.17 is_IPv6Label_Available . . . 1103
8.1156.2.18 is_IPv6SrcAddr_Available . . . 1103
8.1156.2.19 is_IPv6TrafCls_Available . . . 1103
8.1156.2.20 is_NxtHdrProto_Available . . . 1103
8.1156.2.21 is_Precedence_Available . . . 1103
8.1156.2.22 is_TCPDstPort_Available . . . 1103
8.1156.2.23 is_TCPSrcPort_Available . . . 1104
8.1156.2.24 is_TranDstPort_Available . . . 1104
8.1156.2.25 is_TranSrcPort_Available . . . 1104
8.1156.2.26 is_UDPDstPort_Available . . . 1104
8.1156.2.27 is_UDPSrcPort_Available . . . 1104
8.1156.2.28 NxtHdrProto . . . 1104
8.1156.2.29 Precedence . . . 1104
8.1156.2.30 TCPDstPort . . . 1104
8.1156.2.31 TCPSrcPort . . . 1104
8.1156.2.32 TranDstPort . . . 1104
8.1156.2.33 TranSrcPort . . . 1104
8.1156.2.34 UDPDstPort . . . 1104
8.1156.2.35 UDPSrcPort . . . 1104
8.1156.2.36 Version . . . 1104
8.1157 unpack_qos_swiQosFlow_t Struct Reference . . . 1104
8.1157.1 Detailed Description . . . 1105
8.1157.2 Field Documentation . . . 1107
8.1157.2.1 DataRate . . . 1107
8.1157.2.2 Index . . . 1107
8.1157.2.3 is_DataRate_Available . . . 1107
8.1157.2.4 is_Jitter_Available . . . 1107
8.1157.2.5 is_Latency_Available . . . 1107
8.1157.2.6 is_LteQci_Available . . . 1107
8.1157.2.7 is_MaxAllowedPktSz_Available . . . 1107
8.1157.2.8 is_MinPolicedPktSz_Available . . . 1107
8.1157.2.9 is_PktErrRate_Available . . . 1107
8.1157.2.10 is_ProfileId3GPP2_Available . . . 1107
8.1157.2.11 is_TokenBucket_Available . . . 1107

8.1157.2.12 is_TrafficClass_Available . . . 1107
8.1157.2.13 is_val_3GPP2Pri_Available . . . 1107
8.1157.2.14 is_val_3GPPImCn_Available . . . 1107
8.1157.2.15 is_val_3GPPResResidualBER_Available . . . 1107
8.1157.2.16 is_val_3GPPSigInd_Available . . . 1107
8.1157.2.17 is_val_3GPPTraHdlPri_Available . . . 1107
8.1157.2.18 Jitter . . . 1107
8.1157.2.19 Latency . . . 1108
8.1157.2.20 QciQci . . . 1108
8.1157.2.21 MaxAllowedPktSz . . . 1108
8.1157.2.22 MinPolicedPktSz . . . 1108
8.1157.2.23 PktErrRate . . . 1108
8.1157.2.24 ProfileId3GPP2 . . . 1108
8.1157.2.25 TokenBucket . . . 1108
8.1157.2.26 TrafficClass . . . 1108
8.1157.2.27 val_3GPP2Pri . . . 1108
8.1157.2.28 val_3GPPImCn . . . 1108
8.1157.2.29 val_3GPPResResidualBER . . . 1108
8.1157.2.30 val_3GPPSigInd . . . 1108
8.1157.2.31 val_3GPPTraHdlPri . . . 1108
8.1158 unpack_qos_tokenBucket_t Struct Reference . . . 1108
8.1158.1 Detailed Description . . . 1108
8.1158.2 Field Documentation . . . 1108
8.1158.2.1 bucketSz . . . 1108
8.1158.2.2 peakRate . . . 1108
8.1158.2.3 tokenRate . . . 1108
8.1159 unpack_qos_Tos_t Struct Reference . . . 1109
8.1159.1 Detailed Description . . . 1109
8.1159.2 Field Documentation . . . 1109
8.1159.2.1 mask . . . 1109
8.1159.2.2 val . . . 1109
8.1160 unpack_QosFlowStat_t Struct Reference . . . 1109
8.1160.1 Detailed Description . . . 1109
8.1160.2 Field Documentation . . . 1110
8.1160.2.1 bearerId . . . 1110
8.1160.2.2 tx_bytes . . . 1110
8.1160.2.3 tx_bytes_drp . . . 1110
8.1160.2.4 tx_pkt . . . 1110
8.1160.2.5 tx_pkt_drp . . . 1110
8.1161 unpack_rms_GetSMSWake_t Struct Reference . . . 1110

8.1161.1 Detailed Description . . . 1110
8.1161.2 Field Documentation . . . 1110
8.1161.2.1 enabled . . . 1110
8.1161.2.2 wake_mask . . . 1110
8.1162 npack_RMTransferStatistics_ind_t Struct Reference . . . 1110
8.1162.1 Detailed Description . . . 1111
8.1162.2 Field Documentation . . . 1111
8.1162.2.1 RxDropConutTlv . . . 1111
8.1162.2.2 RxOkByteCountTlv . . . 1111
8.1162.2.3 RxOkConutTlv . . . 1111
8.1162.2.4 TxDropConutTlv . . . 1111
8.1162.2.5 TxOkByteCountTlv . . . 1111
8.1162.2.6 TxOkConutTlv . . . 1111
8.1163 npack_sar_SLQSGetRfSarState_t Struct Reference . . . 1111
8.1163.1 Detailed Description . . . 1111
8.1163.2 Field Documentation . . . 1112
8.1163.2.1 pSarRFState . . . 1112
8.1164 npack_sms_GetSMSCAddress_t Struct Reference . . . 1112
8.1164.1 Detailed Description . . . 1112
8.1164.2 Field Documentation . . . 1112
8.1164.2.1 addressSize . . . 1112
8.1164.2.2 pSMSCAddress . . . 1113
8.1164.2.3 pSMSCType . . . 1113
8.1164.2.4 typeSize . . . 1113
8.1165 npack_sms_SaveSMS_t Struct Reference . . . 1113
8.1165.1 Detailed Description . . . 1113
8.1165.2 Field Documentation . . . 1113
8.1165.2.1 pMessageIndex . . . 1113
8.1166 npack_sms_SendSMS_t Struct Reference . . . 1113
8.1166.1 Detailed Description . . . 1113
8.1166.2 Field Documentation . . . 1113
8.1166.2.1 messageFailureCode . . . 1113
8.1166.2.2 messageID . . . 1113
8.1167 npack_sms_SetNewSMSCallback_ind_t Struct Reference . . . 1113
8.1167.1 Detailed Description . . . 1114
8.1167.2 Field Documentation . . . 1114
8.1167.2.1 ETWSPLMNTlv . . . 1114
8.1167.2.2 ETWSTlv . . . 1114
8.1167.2.3 IMSTlv . . . 1114
8.1167.2.4 MMTlv . . . 1114

8.1167.2.5 NewMMTlv . . . 1114
8.1167.2.6 SMSCTlv . . . 1115
8.1167.2.7 TRMessageTlv . . . 1115
8.1168 unpack_sms_SetNewSMSCallback_t Struct Reference . . . 1115
8.1169 unpack_sms_SLQSDeleteSMS_t Struct Reference . . . 1115
8.1170 unpack_sms_SLQSGetIndicationRegister_t Struct Reference . . . 1115
8.1170.1 Detailed Description . . . 1115
8.1170.2 Field Documentation . . . 1115
8.1170.2.1 pGetIndicationRegInfo . . . 1115
8.1171 unpack_sms_SLQSGetMessageWaiting_t Struct Reference . . . 1115
8.1171.1 Detailed Description . . . 1115
8.1171.2 Field Documentation . . . 1115
8.1171.2.1 pGetMsgWaitingInfoResp . . . 1115
8.1172 unpack_sms_SLQSGetSMS_t Struct Reference . . . 1116
8.1172.1 Detailed Description . . . 1116
8.1172.2 Field Documentation . . . 1116
8.1172.2.1 message . . . 1116
8.1172.2.2 messageFormat . . . 1116
8.1172.2.3 messageSize . . . 1116
8.1172.2.4 messageTag . . . 1116
8.1173 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 1116
8.1173.1 Detailed Description . . . 1117
8.1173.2 Field Documentation . . . 1117
8.1173.2.1 pBroadcastConfig . . . 1117
8.1173.2.2 pCDMABroadcastConfig . . . 1117
8.1174 unpack_sms_SLQSGetSMSList_t Struct Reference . . . 1117
8.1174.1 Detailed Description . . . 1117
8.1174.2 Field Documentation . . . 1117
8.1174.2.1 messageList . . . 1117
8.1174.2.2 messageListSize . . . 1117
8.1175 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference . . . 1117
8.1175.1 Detailed Description . . . 1118
8.1175.2 Field Documentation . . . 1118
8.1175.2.1 pGetTransLayerInfo . . . 1118
8.1176 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference . . . 1118
8.1176.1 Detailed Description . . . 1118
8.1176.2 Field Documentation . . . 1118
8.1176.2.1 pGetTransNWRegInfo . . . 1118
8.1177 unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . 1118
8.1178 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 1118

8.1178.1 Detailed Description . . . 1119
8.1178.2 Field Documentation . . . 1119
8.1178.2.1 pMaxStorageSizeResp . . . 1119
8.1179 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference . . . 1119
8.1179.1 Detailed Description . . . 1119
8.1179.2 Field Documentation . . . 1119
8.1179.2.1 pMessageProtocol . . . 1119
8.1180 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference . . . 1119
8.1180.1 Detailed Description . . . 1119
8.1180.2 Field Documentation . . . 1120
8.1180.2.1 pSmsStorage . . . 1120
8.1181 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . 1120
8.1181.1 Detailed Description . . . 1120
8.1181.2 Field Documentation . . . 1120
8.1181.2.1 messageMode . . . 1120
8.1181.2.2 storageType . . . 1120
8.1182 unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . 1120
8.1182.1 Detailed Description . . . 1121
8.1182.2 Field Documentation . . . 1121
8.1182.2.1 fix_rate . . . 1122
8.1182.2.2 fix_rate_reported . . . 1122
8.1182.2.3 fix_type . . . 1122
8.1182.2.4 fix_type_reported . . . 1122
8.1182.2.5 function . . . 1122
8.1182.2.6 function_reported . . . 1122
8.1182.2.7 max_dist . . . 1122
8.1182.2.8 max_dist_reported . . . 1122
8.1182.2.9 max_time . . . 1122
8.1182.2.10 max_time_reported . . . 1122
8.1183 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference . . . 1122
8.1183.1 Detailed Description . . . 1122
8.1183.2 Field Documentation . . . 1122
8.1183.2.1 eventType . . . 1123
8.1183.2.2 SessionInfoConfig . . . 1123
8.1183.2.3 SessionInfoFota . . . 1123
8.1183.2.4 SessionInfoNotification . . . 1123
8.1184 unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 1123
8.1184.1 Detailed Description . . . 1123
8.1184.2 Field Documentation . . . 1125
8.1184.2.1 Date . . . 1125

8.1184.2.2 DateLength . . . 1125
8.1184.2.3 PkgDescLength . . . 1125
8.1184.2.4 PkgDescription . . . 1125
8.1184.2.5 PkgName . . . 1125
8.1184.2.6 PkgNameLength . . . 1125
8.1184.2.7 RetryCount . . . 1125
8.1184.2.8 SessionState . . . 1125
8.1184.2.9 SessionType . . . 1125
8.1184.2.10 Severity . . . 1125
8.1184.2.11 Source . . . 1125
8.1184.2.12 SourceLength . . . 1125
8.1184.2.13 Status . . . 1125
8.1184.2.14 Time . . . 1125
8.1184.2.15 TimeLength . . . 1125
8.1184.2.16 UpdateCompleteStatus . . . 1125
8.1185 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . 1125
8.1185.1 Detailed Description . . . 1126
8.1185.2 Field Documentation . . . 1127
8.1185.2.1 Autosdm . . . 1127
8.1185.2.2 FOTAdownload . . . 1127
8.1185.2.3 FOTAUpdate . . . 1127
8.1185.2.4 FwAutoCheck . . . 1127
8.1185.2.5 OMADMEnabled . . . 1127
8.1186 unpack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 1127
8.1186.1 Detailed Description . . . 1127
8.1186.2 Field Documentation . . . 1127
8.1186.2.1 FwAvailability . . . 1127
8.1187 unpack_uim_ChangePin_t Struct Reference . . . 1127
8.1187.1 Detailed Description . . . 1127
8.1187.2 Field Documentation . . . 1128
8.1187.2.1 pEncryptedPIN1 . . . 1128
8.1187.2.2 pIndicationToken . . . 1128
8.1187.2.3 pRemainingRetries . . . 1128
8.1187.2.4 Tlvresult . . . 1128
8.1188 unpack_uim_GetCardStatus_t Struct Reference . . . 1128
8.1188.1 Detailed Description . . . 1128
8.1188.2 Field Documentation . . . 1128
8.1188.2.1 pCardStatus . . . 1129
8.1188.2.2 pHotSwapStatus . . . 1129
8.1188.2.3 Tlvresult . . . 1129

8.1189 unpack_uim_ReadTransparent_t Struct Reference . . . 1129
8.1189.1 Detailed Description . . . 1129
8.1189.2 Field Documentation . . . 1129
8.1189.2.1 pCardResult . . . 1129
8.1189.2.2 pEncryptedData . . . 1129
8.1189.2.3 pIndicationToken . . . 1129
8.1189.2.4 pReadResult . . . 1129
8.1189.2.5 Tlvresult . . . 1129
8.1190 unpack_uim_SetPinProtection_t Struct Reference . . . 1130
8.1190.1 Detailed Description . . . 1130
8.1190.2 Field Documentation . . . 1130
8.1190.2.1 pEncryptedPIN1 . . . 1130
8.1190.2.2 pIndicationToken . . . 1130
8.1190.2.3 pRemainingRetries . . . 1130
8.1190.2.4 Tlvresult . . . 1130
8.1191 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference . . . 1130
8.1191.1 Detailed Description . . . 1130
8.1191.2 Field Documentation . . . 1131
8.1191.2.1 bNumberOfPhySlots . . . 1131
8.1191.2.2 slotsstatusChange . . . 1131
8.1192 unpack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 1131
8.1192.1 Detailed Description . . . 1131
8.1192.2 Field Documentation . . . 1131
8.1192.2.1 pAuthenticateResult . . . 1131
8.1192.2.2 pCardResult . . . 1131
8.1192.2.3 pIndicationToken . . . 1131
8.1193 unpack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 1131
8.1193.1 Detailed Description . . . 1132
8.1193.2 Field Documentation . . . 1132
8.1193.2.1 pRemainingRetries . . . 1132
8.1194 unpack_uim_SLQSUIMEventRegister_t Struct Reference . . . 1132
8.1194.1 Detailed Description . . . 1132
8.1194.2 Field Documentation . . . 1132
8.1194.2.1 eventMask . . . 1132
8.1195 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 1132
8.1195.1 Detailed Description . . . 1132
8.1195.2 Field Documentation . . . 1133
8.1195.2.1 pAutoSelection . . . 1133
8.1195.2.2 pHaltSubscription . . . 1133
8.1195.2.3 pPersonalizationStatus . . . 1133

8.1196 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 1133
8.1196.1 Detailed Description . . . 1133
8.1196.2 Field Documentation . . . 1134
8.1196.2.1 pCardResult . . . 1134
8.1196.2.2 pFileAttributes . . . 1134
8.1196.2.3 pIndicationToken . . . 1134
8.1197 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference . . . 1134
8.1197.1 Detailed Description . . . 1134
8.1197.2 Field Documentation . . . 1134
8.1197.2.1 pNumberOfPhySlot . . . 1134
8.1197.2.2 pUimSlotsStatus . . . 1134
8.1198 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 1134
8.1198.1 Detailed Description . . . 1134
8.1198.2 Field Documentation . . . 1135
8.1198.2.1 pRefreshEvent . . . 1135
8.1199 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference . . . 1135
8.1199.1 Detailed Description . . . 1135
8.1199.2 Field Documentation . . . 1135
8.1199.2.1 pCardStatus . . . 1135
8.1200 unpack_uim_UnblockPin_t Struct Reference . . . 1135
8.1200.1 Detailed Description . . . 1135
8.1200.2 Field Documentation . . . 1136
8.1200.2.1 pEncryptedPIN1 . . . 1136
8.1200.2.2 pIndicationToken . . . 1136
8.1200.2.3 pRemainingRetries . . . 1136
8.1200.2.4 Tlvresult . . . 1136
8.1201 unpack_uim_VerifyPin_t Struct Reference . . . 1136
8.1201.1 Detailed Description . . . 1136
8.1201.2 Field Documentation . . . 1136
8.1201.2.1 pEncryptedPIN1 . . . 1136
8.1201.2.2 pIndicationToken . . . 1136
8.1201.2.3 pRemainingRetries . . . 1136
8.1201.2.4 Tlvresult . . . 1136
8.1202 unpack_voice_SLQSOriginateUSSD_t Struct Reference . . . 1136
8.1202.1 Field Documentation . . . 1137
8.1202.1.1 pAlphaIDInfo . . . 1137
8.1202.1.2 pCallId . . . 1137
8.1202.1.3 pCcResultType . . . 1137
8.1202.1.4 pCCSuppsType . . . 1137
8.1202.1.5 pfailureCause . . . 1137

8.1202.1.6 pUSSDInfo . . . 1137
8.1203 npack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 1137
8.1203.1 Detailed Description . . . 1137
8.1203.2 Field Documentation . . . 1137
8.1203.2.1 pCallId . . . 1137
8.1204 npack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 1137
8.1204.1 Detailed Description . . . 1137
8.1204.2 Field Documentation . . . 1138
8.1204.2.1 pCallID . . . 1138
8.1205 npack_voice_SLQSVoiceDialCall_t Struct Reference . . . 1138
8.1205.1 Detailed Description . . . 1138
8.1205.2 Field Documentation . . . 1138
8.1205.2.1 pAlphaIDInfo . . . 1138
8.1205.2.2 pCallID . . . 1139
8.1205.2.3 pCCResultType . . . 1139
8.1205.2.4 pCCSUPSType . . . 1139
8.1206 npack_voice_SLQSVoiceEndCall_t Struct Reference . . . 1139
8.1206.1 Detailed Description . . . 1139
8.1206.2 Field Documentation . . . 1139
8.1206.2.1 pCallId . . . 1139
8.1207 npack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference . . . 1139
8.1207.1 Detailed Description . . . 1139
8.1207.2 Field Documentation . . . 1141
8.1207.2.1 pArrAlertingPattern . . . 1141
8.1207.2.2 pArrAlertingType . . . 1141
8.1207.2.3 pArrAlphaID . . . 1141
8.1207.2.4 pArrCalledPartyNum . . . 1141
8.1207.2.5 pArrCallEndReason . . . 1141
8.1207.2.6 pArrCallInfo . . . 1141
8.1207.2.7 pArrConnectPartyNum . . . 1141
8.1207.2.8 pArrDiagInfo . . . 1141
8.1207.2.9 pArrRedirPartyNum . . . 1141
8.1207.2.10 pArrRemotePartyName . . . 1141
8.1207.2.11 pArrRemotePartyNum . . . 1141
8.1207.2.12 pArrSvcOption . . . 1142
8.1207.2.13 pArrUUSInfo . . . 1142
8.1207.2.14 pOTASPStatus . . . 1142
8.1207.2.15 pVoicePrivacy . . . 1142
8.1208 npack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 1142
8.1208.1 Detailed Description . . . 1142

8.1208.2 Field Documentation . . . 1143
8.1208.2.1 pAlphaIDInfo . . . 1143
8.1208.2.2 pCallID . . . 1143
8.1208.2.3 pCCResType . . . 1143
8.1208.2.4 pCCSUPSType . . . 1143
8.1208.2.5 pFailCause . . . 1143
8.1208.2.6 pSvcClass . . . 1143
8.1209 unpack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 1143
8.1209.1 Detailed Description . . . 1143
8.1209.2 Field Documentation . . . 1144
8.1209.2.1 pAlphaIDInfo . . . 1144
8.1209.2.2 pCallID . . . 1144
8.1209.2.3 pCCResType . . . 1144
8.1209.2.4 pCCSUPSType . . . 1144
8.1209.2.5 pFailCause . . . 1144
8.1209.2.6 pGetCallFWExtInfo . . . 1144
8.1209.2.7 pGetCallFWInfo . . . 1144
8.1210 unpack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 1144
8.1210.1 Detailed Description . . . 1145
8.1210.2 Field Documentation . . . 1147
8.1210.2.1 pAlertingPattern . . . 1147
8.1210.2.2 pAlertType . . . 1147
8.1210.2.3 pAlphaIDInfo . . . 1147
8.1210.2.4 pCallInfo . . . 1147
8.1210.2.5 pConnectNumInfo . . . 1147
8.1210.2.6 pDiagInfo . . . 1147
8.1210.2.7 pOTASPStatus . . . 1147
8.1210.2.8 pRemotePartyName . . . 1147
8.1210.2.9 pRemotePartyNum . . . 1147
8.1210.2.10 pSrvOpt . . . 1147
8.1210.2.11 pUUSInfo . . . 1147
8.1210.2.12 pVoicePrivacy . . . 1147
8.1211 unpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 1147
8.1211.1 Detailed Description . . . 1148
8.1211.2 Field Documentation . . . 1148
8.1211.2.1 pAlphaIDInfo . . . 1148
8.1211.2.2 pCallID . . . 1148
8.1211.2.3 pCCResType . . . 1148
8.1211.2.4 pCCSUPSType . . . 1148
8.1211.2.5 pFailCause . . . 1148

8.1211.2.6 pSvcClass . . . 1149
8.1212 unpack_voice_SLQSVoiceGetCLIP_t Struct Reference . . . 1149
8.1212.1 Detailed Description . . . 1149
8.1212.2 Field Documentation . . . 1150
8.1212.2.1 pAlphaIDInfo . . . 1150
8.1212.2.2 pCallID . . . 1150
8.1212.2.3 pCCResType . . . 1150
8.1212.2.4 pCCSUPSType . . . 1150
8.1212.2.5 pCLIPResp . . . 1150
8.1212.2.6 pFailCause . . . 1150
8.1213 unpack_voice_SLQSVoiceGetCLIR_t Struct Reference . . . 1150
8.1213.1 Detailed Description . . . 1150
8.1213.2 Field Documentation . . . 1151
8.1213.2.1 pAlphaIDInfo . . . 1151
8.1213.2.2 pCallID . . . 1151
8.1213.2.3 pCCResType . . . 1151
8.1213.2.4 pCCSUPSType . . . 1151
8.1213.2.5 pCLIRResp . . . 1151
8.1213.2.6 pFailCause . . . 1151
8.1214 unpack_voice_SLQSVoiceGetCNAP_t Struct Reference . . . 1151
8.1214.1 Detailed Description . . . 1151
8.1214.2 Field Documentation . . . 1152
8.1214.2.1 pAlphaIDInfo . . . 1152
8.1214.2.2 pCallID . . . 1152
8.1214.2.3 pCCResType . . . 1152
8.1214.2.4 pCCSUPSType . . . 1152
8.1214.2.5 pCNAPResp . . . 1152
8.1214.2.6 pFailCause . . . 1152
8.1215 unpack_voice_SLQSVoiceGetCOLP_t Struct Reference . . . 1152
8.1215.1 Detailed Description . . . 1153
8.1215.2 Field Documentation . . . 1153
8.1215.2.1 pAlphaIDInfo . . . 1153
8.1215.2.2 pCallID . . . 1153
8.1215.2.3 pCCResType . . . 1153
8.1215.2.4 pCCSUPSType . . . 1154
8.1215.2.5 pCOLPResp . . . 1154
8.1215.2.6 pFailCause . . . 1154
8.1216 unpack_voice_SLQSVoiceGetCOLR_t Struct Reference . . . 1154
8.1216.1 Detailed Description . . . 1154
8.1216.2 Field Documentation . . . 1155

8.1216.2.1 pAlphaIDInfo . . . 1155
8.1216.2.2 pCallID . . . 1155
8.1216.2.3 pCCResType . . . 1155
8.1216.2.4 pCCSUPSType . . . 1155
8.1216.2.5 pCOLRResp . . . 1155
8.1216.2.6 pFailCause . . . 1155
8.1217 unpack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 1155
8.1217.1 Detailed Description . . . 1155
8.1217.2 Field Documentation . . . 1156
8.1217.2.1 pAirTimerCnt . . . 1156
8.1217.2.2 pAutoAnswerStat . . . 1156
8.1217.2.3 pCurAMRConfig . . . 1156
8.1217.2.4 pCurPrefVoiceSO . . . 1156
8.1217.2.5 pCurrTTYMode . . . 1156
8.1217.2.6 pCurVoiceDomainPref . . . 1157
8.1217.2.7 pCurVoicePrivacyPref . . . 1157
8.1217.2.8 pRoamTimerCnt . . . 1157
8.1218 unpack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 1157
8.1218.1 Detailed Description . . . 1157
8.1218.2 Field Documentation . . . 1157
8.1218.2.1 pFailCause . . . 1157
8.1219 unpack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 1157
8.1219.1 Detailed Description . . . 1157
8.1219.2 Field Documentation . . . 1157
8.1219.2.1 pCallID . . . 1157
8.1220 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 1157
8.1220.1 Detailed Description . . . 1158
8.1220.2 Field Documentation . . . 1158
8.1220.2.1 pAlphaIDInfo . . . 1158
8.1220.2.2 pCallID . . . 1158
8.1220.2.3 pCCResType . . . 1158
8.1220.2.4 pCCSUPSType . . . 1159
8.1220.2.5 pFailCause . . . 1159
8.1221 unpack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 1159
8.1221.1 Detailed Description . . . 1159
8.1221.2 Field Documentation . . . 1160
8.1221.2.1 pAirTimerStatus . . . 1160
8.1221.2.2 pAutoAnsStatus . . . 1160
8.1221.2.3 pPrefVoiceSOStatus . . . 1160
8.1221.2.4 pRoamTimerStatus . . . 1160

8.1221.2.5 pTTYConfigStatus . . . 1160
8.1221.2.6 pVoiceDomainPrefStatus . . . 1160
8.1222 npack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 1160
8.1222.1 Detailed Description . . . 1160
8.1222.2 Field Documentation . . . 1161
8.1222.2.1 pAlphaIDInfo . . . 1161
8.1222.2.2 pCallID . . . 1161
8.1222.2.3 pCCResultType . . . 1161
8.1222.2.4 pCCSUPSType . . . 1161
8.1222.2.5 pFailCause . . . 1161
8.1223 npack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 1161
8.1223.1 Detailed Description . . . 1161
8.1223.2 Field Documentation . . . 1162
8.1223.2.1 pCallID . . . 1162
8.1224 npack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 1162
8.1224.1 Detailed Description . . . 1162
8.1224.2 Field Documentation . . . 1162
8.1224.2.1 callID . . . 1162
8.1225 npack_wds_DHCPv4ClientLease_ind_t Struct Reference . . . 1162
8.1225.1 Field Documentation . . . 1162
8.1225.1.1 DHCPv4LeaseOptTlv . . . 1162
8.1225.1.2 DHCPv4LeaseStateTlv . . . 1163
8.1225.1.3 IPv4AddrTlv . . . 1163
8.1225.1.4 ProfileIdTlv . . . 1163
8.1226 npack_wds_GetAutoconnect_t Struct Reference . . . 1163
8.1226.1 Detailed Description . . . 1163
8.1226.2 Field Documentation . . . 1163
8.1226.2.1 psetting . . . 1163
8.1227 npack_wds_GetByteTotals_t Struct Reference . . . 1163
8.1227.1 Detailed Description . . . 1163
8.1227.2 Field Documentation . . . 1163
8.1227.2.1 pRXTotalBytes . . . 1163
8.1227.2.2 pTXTotalBytes . . . 1163
8.1228 npack_wds_GetConnectionRate_t Struct Reference . . . 1164
8.1228.1 Detailed Description . . . 1164
8.1228.2 Field Documentation . . . 1164
8.1228.2.1 currentChannelRXRate . . . 1164
8.1228.2.2 currentChannelTXRate . . . 1164
8.1228.2.3 maxChannelRXRate . . . 1164
8.1228.2.4 maxChannelTXRate . . . 1164

8.1229 unpack_wds_GetDataBearerTechnology_t Struct Reference . . . 1164
8.1229.1 Detailed Description . . . 1164
8.1229.2 Field Documentation . . . 1165
8.1229.2.1 pDataBearer . . . 1165
8.1230 unpack_wds_GetDefaultProfile_t Struct Reference . . . 1165
8.1230.1 Detailed Description . . . 1165
8.1230.2 Field Documentation . . . 1166
8.1230.2.1 apnname . . . 1166
8.1230.2.2 apnsize . . . 1166
8.1230.2.3 auth . . . 1166
8.1230.2.4 ipaddr . . . 1166
8.1230.2.5 ipaddrv6 . . . 1166
8.1230.2.6 name . . . 1166
8.1230.2.7 namesize . . . 1166
8.1230.2.8 pdptype . . . 1166
8.1230.2.9 pridns . . . 1166
8.1230.2.10 pridnsv6 . . . 1166
8.1230.2.11 secdns . . . 1166
8.1230.2.12 secdnsv6 . . . 1166
8.1230.2.13 username . . . 1166
8.1230.2.14 usersize . . . 1166
8.1231 unpack_wds_GetDefaultProfileNum_t Struct Reference . . . 1166
8.1231.1 Detailed Description . . . 1166
8.1231.2 Field Documentation . . . 1167
8.1231.2.1 index . . . 1167
8.1232 unpack_wds_GetDormancyState_t Struct Reference . . . 1167
8.1232.1 Detailed Description . . . 1167
8.1232.2 Field Documentation . . . 1167
8.1232.2.1 dormancyState . . . 1167
8.1233 unpack_wds_GetLastMobileIPError_t Struct Reference . . . 1167
8.1233.1 Detailed Description . . . 1167
8.1233.2 Field Documentation . . . 1167
8.1233.2.1 error . . . 1167
8.1234 unpack_wds_GetMobileIP_t Struct Reference . . . 1167
8.1234.1 Detailed Description . . . 1167
8.1234.2 Field Documentation . . . 1168
8.1234.2.1 mipMode . . . 1168
8.1235 unpack_wds_GetMobileIPProfile_t Struct Reference . . . 1168
8.1235.1 Detailed Description . . . 1168
8.1235.2 Field Documentation . . . 1168

8.1235.2.1 AAASPI . . . 1168
8.1235.2.2 AAAState . . . 1168
8.1235.2.3 address . . . 1168
8.1235.2.4 enabled . . . 1168
8.1235.2.5 HASPI . . . 1168
8.1235.2.6 HAState . . . 1168
8.1235.2.7 NAI . . . 1168
8.1235.2.8 naiSize . . . 1168
8.1235.2.9 primaryHA . . . 1169
8.1235.2.10 revTunneling . . . 1169
8.1235.2.11 secondaryHA . . . 1169
8.1236 unpack_wds_GetPacketStatistics_t Struct Reference . . . 1169
8.1236.1 Detailed Description . . . 1169
8.1236.2 Field Documentation . . . 1170
8.1236.2.1 pRXDroppedCount . . . 1170
8.1236.2.2 pRXOkBytesCount . . . 1170
8.1236.2.3 pRXOKBytesLastCall . . . 1170
8.1236.2.4 pRXPacketErrors . . . 1170
8.1236.2.5 pRXPacketOverflows . . . 1170
8.1236.2.6 pRXPacketSuccesses . . . 1170
8.1236.2.7 pTXDroppedCount . . . 1170
8.1236.2.8 pTXOkBytesCount . . . 1170
8.1236.2.9 pTXOKBytesLastCall . . . 1170
8.1236.2.10 pTXPacketErrors . . . 1170
8.1236.2.11 pTXPacketOverflows . . . 1170
8.1236.2.12 pTXPacketSuccesses . . . 1170
8.1237 unpack_wds_GetPacketStatus_t Struct Reference . . . 1170
8.1237.1 Detailed Description . . . 1171
8.1237.2 Field Documentation . . . 1171
8.1237.2.1 rXDroppedCount . . . 1171
8.1237.2.2 rXOkBytesCount . . . 1171
8.1237.2.3 rXOKBytesLastCall . . . 1171
8.1237.2.4 rXPacketErrors . . . 1171
8.1237.2.5 rXPacketOverflows . . . 1171
8.1237.2.6 rXPacketSuccesses . . . 1171
8.1237.2.7 tXDroppedCount . . . 1171
8.1237.2.8 tXOkBytesCount . . . 1171
8.1237.2.9 tXOKBytesLastCall . . . 1171
8.1237.2.10 tXPacketErrors . . . 1171
8.1237.2.11 tXPacketOverflows . . . 1171

8.1237.2.12 tXPacketSuccesses . . . 1171
8.1238 unpack_wds_GetSessionDuration_t Struct Reference . . . 1171
8.1238.1 Detailed Description . . . 1172
8.1238.2 Field Documentation . . . 1172
8.1238.2.1 callDuration . . . 1172
8.1239 unpack_wds_GetSessionState_t Struct Reference . . . 1172
8.1239.1 Detailed Description . . . 1172
8.1239.2 Field Documentation . . . 1172
8.1239.2.1 connectionStatus . . . 1172
8.1240 unpack_wds_RMSetTransferStatistics_t Struct Reference . . . 1172
8.1241 unpack_wds_SetMobileIPProfile_t Struct Reference . . . 1172
8.1242 unpack_wds_SLQSCreateProfile_t Struct Reference . . . 1172
8.1242.1 Detailed Description . . . 1172
8.1242.2 Field Documentation . . . 1172
8.1242.2.1 pCreateProfileOut . . . 1172
8.1242.2.2 pProfileID . . . 1173
8.1242.2.3 Tlvresult . . . 1173
8.1243 unpack_wds_SLQSDeleteProfile_t Struct Reference . . . 1173
8.1243.1 Detailed Description . . . 1173
8.1243.2 Field Documentation . . . 1173
8.1243.2.1 extendedErrorCode . . . 1173
8.1244 unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . 1173
8.1244.1 Detailed Description . . . 1173
8.1244.2 Field Documentation . . . 1174
8.1244.2.1 _3gppRelease . . . 1174
8.1244.2.2 defaultPDNEnabled . . . 1174
8.1244.2.3 LTEAttachProfile . . . 1174
8.1244.2.4 LTEAttachProfileList . . . 1174
8.1244.2.5 LTEAttachProfileListLen . . . 1174
8.1244.2.6 profileList . . . 1174
8.1245 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 1174
8.1245.1 Detailed Description . . . 1174
8.1245.2 Field Documentation . . . 1175
8.1245.2.1 currNetworkInfo . . . 1175
8.1245.2.2 networkInfoLen . . . 1175
8.1245.2.3 prefNetwork . . . 1175
8.1246 unpack_wds_SLQSGetCurrentChannelRate_t Struct Reference . . . 1175
8.1246.1 Detailed Description . . . 1175
8.1246.2 Field Documentation . . . 1176
8.1246.2.1 current_channel_rx_rate . . . 1176

8.1246.2.2 current_channel_tx_rate . . . 1176
8.1246.2.3 max_channel_rx_rate . . . 1176
8.1246.2.4 max_channel_tx_rate . . . 1176
8.1247 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 1176
8.1247.1 Detailed Description . . . 1176
8.1247.2 Field Documentation . . . 1176
8.1247.2.1 curDataBearerTechnology . . . 1176
8.1247.2.2 dataBearerMask . . . 1176
8.1247.2.3 lastCallDataBearerTechnology . . . 1176
8.1248 unpack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 1176
8.1248.1 Detailed Description . . . 1177
8.1248.2 Field Documentation . . . 1177
8.1248.2.1 callEndReason . . . 1177
8.1248.2.2 channelRate . . . 1177
8.1248.2.3 connectionStatus . . . 1177
8.1248.2.4 dataBearerTech . . . 1177
8.1248.2.5 dormancyStatus . . . 1177
8.1248.2.6 lastCallDataBearerTech . . . 1177
8.1248.2.7 lastCallRXOKBytesCnt . . . 1177
8.1248.2.8 lastCallTXOKBytesCnt . . . 1177
8.1248.2.9 mdmCallDurationActive . . . 1177
8.1248.2.10 rxOKBytesCount . . . 1177
8.1248.2.11 txOKBytesCount . . . 1177
8.1249 unpack_wds_SLQSGetProfileSettings_t Struct Reference . . . 1177
8.1249.1 Field Documentation . . . 1177
8.1249.1.1 pProfileSettings . . . 1178
8.1249.1.2 ProfileType . . . 1178
8.1249.1.3 Tlvresult . . . 1178
8.1250 unpack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 1178
8.1250.1 Detailed Description . . . 1178
8.1250.2 Field Documentation . . . 1179
8.1250.2.1 APNName . . . 1179
8.1250.2.2 Authentication . . . 1179
8.1250.2.3 DomainList . . . 1179
8.1250.2.4 GPRSGrantedQoS . . . 1179
8.1250.2.5 GWAddressV4 . . . 1179
8.1250.2.6 IMCNflag . . . 1179
8.1250.2.7 IPFamilyPreference . . . 1179
8.1250.2.8 IPv4 . . . 1179
8.1250.2.9 IPV6AddrInfo . . . 1179

8.1250.2.10 IPV6GWAddrInfo . . . 1179
8.1250.2.11 Mtu . . . 1179
8.1250.2.12 PCSCFAddrPCO . . . 1179
8.1250.2.13 PCSCFFQDNAddrList . . . 1179
8.1250.2.14 PDPType . . . 1179
8.1250.2.15 PrimaryDNSV4 . . . 1179
8.1250.2.16 PrimaryDNSV6 . . . 1179
8.1250.2.17 ProfileID . . . 1179
8.1250.2.18 ProfileName . . . 1179
8.1250.2.19 SecondaryDNSV4 . . . 1180
8.1250.2.20 SecondaryDNSV6 . . . 1180
8.1250.2.21 ServerAddrList . . . 1180
8.1250.2.22 SubnetMaskV4 . . . 1180
8.1250.2.23 Technology . . . 1180
8.1250.2.24 UMTSGrantedQoS . . . 1180
8.1250.2.25 Username . . . 1180
8.1251 unpack_wds_SLQSModifyProfile_t Struct Reference . . . 1180
8.1251.1 Detailed Description . . . 1180
8.1251.2 Field Documentation . . . 1180
8.1251.2.1 pExtErrorCode . . . 1180
8.1252 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 1180
8.1252.1 Detailed Description . . . 1180
8.1252.2 Field Documentation . . . 1180
8.1252.2.1 Tlvresult . . . 1180
8.1253 unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference . . . 1180
8.1253.1 Detailed Description . . . 1181
8.1253.2 Field Documentation . . . 1181
8.1253.2.1 bearerID . . . 1181
8.1253.2.2 conn_status . . . 1181
8.1253.2.3 ipFamily . . . 1181
8.1253.2.4 reconfigReqd . . . 1181
8.1253.2.5 sessionEndReason . . . 1181
8.1253.2.6 techName . . . 1181
8.1253.2.7 verboseSessnEndReason . . . 1181
8.1253.2.8 verboseSessnEndReasonType . . . 1181
8.1254 unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference . . . 1181
8.1254.1 Detailed Description . . . 1182
8.1254.2 Field Documentation . . . 1182
8.1254.2.1 currDBTechAvail . . . 1182
8.1254.2.2 currNWInfo . . . 1182

8.1254.2.3 dataSysStatAvail . . . 1182
8.1254.2.4 dBTechAvail . . . 1183
8.1254.2.5 dBtechExtAvail . . . 1183
8.1254.2.6 dBTechExtRatValue . . . 1183
8.1254.2.7 dBTechExtSoMask . . . 1183
8.1254.2.8 dBTechnology . . . 1183
8.1254.2.9 dBtechnologyExt . . . 1183
8.1254.2.10 dormancyStatAvail . . . 1183
8.1254.2.11 dormancyStatus . . . 1183
8.1254.2.12 ipstatAvail . . . 1183
8.1254.2.13 ipStatus . . . 1183
8.1254.2.14 netInfoLen . . . 1183
8.1254.2.15 prefNetwork . . . 1183
8.1254.2.16 ratMask . . . 1183
8.1254.2.17 rx_bytes . . . 1183
8.1254.2.18 rx_pkts . . . 1183
8.1254.2.19 soMask . . . 1183
8.1254.2.20 tx_bytes . . . 1183
8.1254.2.21 tx_pkts . . . 1183
8.1254.2.22 ueStatAvail . . . 1183
8.1255 qmi_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 1183
8.1255.1 Detailed Description . . . 1183
8.1255.2 Field Documentation . . . 1183
8.1255.2.1 pHwConfig . . . 1184
8.1255.2.2 pRequestOptionList . . . 1184
8.1256 qmi_wds_SLQSSGetLoopback_t Struct Reference . . . 1184
8.1256.1 Detailed Description . . . 1184
8.1256.2 Field Documentation . . . 1184
8.1256.2.1 ByteLoopbackMode . . . 1184
8.1256.2.2 ByteLoopbackMultiplier . . . 1184
8.1257 qmi_wds_SLQSStartDataSession_t Struct Reference . . . 1184
8.1257.1 Detailed Description . . . 1184
8.1257.2 Field Documentation . . . 1185
8.1257.2.1 pFailureReason . . . 1185
8.1257.2.2 psid . . . 1185
8.1257.2.3 pVerboseFailReasonType . . . 1185
8.1257.2.4 pVerboseFailureReason . . . 1185
8.1258 qmi_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 1185
8.1258.1 Detailed Description . . . 1185
8.1258.2 Field Documentation . . . 1186

8.1258.2.1 apnName . . . 1186
8.1258.2.2 bearerId . . . 1186
8.1258.2.3 contextId . . . 1186
8.1258.2.4 ipv4Address . . . 1186
8.1258.2.5 ipv4GWAddress . . . 1186
8.1258.2.6 ipv6Address . . . 1186
8.1258.2.7 ipv6GWAddress . . . 1186
8.1258.2.8 prDNSIPv4Address . . . 1186
8.1258.2.9 prDNSIPv6Address . . . 1186
8.1258.2.10 prPCSCFIPv4Address . . . 1186
8.1258.2.11 prPCSCFIPv6Address . . . 1186
8.1258.2.12 seDNSIPv4Address . . . 1186
8.1258.2.13 seDNSIPv6Address . . . 1186
8.1258.2.14 sePCSCFIPv4Address . . . 1186
8.1258.2.15 sePCSCFIPv6Address . . . 1186
8.1259 UnPackGetProfileSettingOut Struct Reference . . . 1186
8.1259.1 Field Documentation . . . 1187
8.1259.1.1 curProfile . . . 1187
8.1259.1.2 pExtErrCode . . . 1187
8.1260 UnpackWdsProfileParam Union Reference . . . 1187
8.1260.1 Field Documentation . . . 1187
8.1260.1.1 SlqsProfile3GPP . . . 1187
8.1260.1.2 SlqsProfile3GPP2 . . . 1187
8.1261 USBCompConfig Struct Reference . . . 1187
8.1261.1 Detailed Description . . . 1187
8.1261.2 Field Documentation . . . 1188
8.1261.2.1 pUSBComp . . . 1188
8.1262 USBCompParams Struct Reference . . . 1188
8.1262.1 Detailed Description . . . 1188
8.1262.2 Field Documentation . . . 1190
8.1262.2.1 pNumSupUSBComps . . . 1190
8.1262.2.2 pSupUSBComps . . . 1190
8.1262.2.3 pUSBComp . . . 1190
8.1263 USSDNoWaitIndicationInfo Struct Reference . . . 1190
8.1263.1 Detailed Description . . . 1190
8.1263.2 Field Documentation . . . 1191
8.1263.2.1 pAlphaIdentifier . . . 1191
8.1263.2.2 pError . . . 1191
8.1263.2.3 pFailureCause . . . 1191
8.1263.2.4 pUSSDData . . . 1191

8.1264 USSDRespFNetwork Struct Reference . . . 1191
8.1264.1 Detailed Description . . . 1191
8.1264.2 Field Documentation . . . 1191
8.1264.2.1 pRespData . . . 1191
8.1264.2.2 pTypeCode . . . 1191
8.1265 USSInfo Struct Reference . . . 1191
8.1265.1 Detailed Description . . . 1192
8.1265.2 Field Documentation . . . 1192
8.1265.2.1 ussData . . . 1192
8.1265.2.2 ussDCS . . . 1192
8.1265.2.3 ussLen . . . 1192
8.1266 USSResp Struct Reference . . . 1192
8.1266.1 Field Documentation . . . 1192
8.1266.1.1 pAlphaIDInfo . . . 1192
8.1266.1.2 pCallId . . . 1192
8.1266.1.3 pCcResultType . . . 1192
8.1266.1.4 pCCSuppsType . . . 1192
8.1266.1.5 pfailureCause . . . 1193
8.1266.1.6 pUSSDInfo . . . 1193
8.1267 UUSInfo Struct Reference . . . 1193
8.1267.1 Detailed Description . . . 1193
8.1267.2 Field Documentation . . . 1194
8.1267.2.1 UUSData . . . 1194
8.1267.2.2 UUSDatalen . . . 1194
8.1267.2.3 UUSDcs . . . 1194
8.1267.2.4 UUSType . . . 1194
8.1268 VerifyUIMPIN Struct Reference . . . 1194
8.1268.1 Detailed Description . . . 1194
8.1268.2 Field Documentation . . . 1194
8.1268.2.1 pinID . . . 1194
8.1268.2.2 pinLen . . . 1194
8.1268.2.3 pinVal . . . 1194
8.1269 Voice_airTimer Struct Reference . . . 1194
8.1269.1 Detailed Description . . . 1195
8.1269.2 Field Documentation . . . 1195
8.1269.2.1 airTimerValue . . . 1195
8.1269.2.2 namID . . . 1195
8.1270 Voice_allCallsAlphaIDInfo Struct Reference . . . 1195
8.1270.1 Detailed Description . . . 1195
8.1270.2 Field Documentation . . . 1195

8.1270.2.1 AlphaIDInfo . . . 1195
8.1270.2.2 callID . . . 1195
8.1271 voice_allCallsDiagInfo Struct Reference . . . 1195
8.1271.1 Detailed Description . . . 1196
8.1271.2 Field Documentation . . . 1196
8.1271.2.1 callID . . . 1196
8.1271.2.2 DiagInfo . . . 1196
8.1272 voice_allCallsUUSInfo Struct Reference . . . 1196
8.1272.1 Detailed Description . . . 1196
8.1272.2 Field Documentation . . . 1196
8.1272.2.1 callID . . . 1196
8.1272.2.2 uusInfo . . . 1196
8.1273 voice_alphaIDInfo Struct Reference . . . 1196
8.1273.1 Detailed Description . . . 1197
8.1273.2 Field Documentation . . . 1197
8.1273.2.1 alphaDcs . . . 1197
8.1273.2.2 alphaLen . . . 1197
8.1273.2.3 alphaText . . . 1197
8.1274 voice_arrAlertingPattern Struct Reference . . . 1197
8.1274.1 Detailed Description . . . 1197
8.1274.2 Field Documentation . . . 1198
8.1274.2.1 alertingPattern . . . 1198
8.1274.2.2 callID . . . 1198
8.1274.2.3 numInstances . . . 1198
8.1275 voice_arrAlertingType Struct Reference . . . 1198
8.1275.1 Detailed Description . . . 1198
8.1275.2 Field Documentation . . . 1199
8.1275.2.1 AlertingType . . . 1199
8.1275.2.2 callID . . . 1199
8.1275.2.3 numInstances . . . 1199
8.1276 voice_arrAlphaID Struct Reference . . . 1199
8.1276.1 Detailed Description . . . 1199
8.1276.2 Field Documentation . . . 1199
8.1276.2.1 allCallsAlphaIDInfoArr . . . 1199
8.1276.2.2 numInstances . . . 1199
8.1277 voice_arrCalledPartyNum Struct Reference . . . 1199
8.1277.1 Detailed Description . . . 1199
8.1277.2 Field Documentation . . . 1200
8.1277.2.1 CalledPartyNum . . . 1200
8.1277.2.2 numInstances . . . 1200

8.1278 voice_arrCallEndReason Struct Reference . . . 1200
8.1278.1 Detailed Description . . . 1200
8.1278.2 Field Documentation . . . 1200
8.1278.2.1 callEndReason . . . 1200
8.1278.2.2 callID . . . 1200
8.1278.2.3 numInstances . . . 1200
8.1279 voice_arrCallInfo Struct Reference . . . 1200
8.1279.1 Detailed Description . . . 1201
8.1279.2 Field Documentation . . . 1201
8.1279.2.1 getAllCallInfo . . . 1201
8.1279.2.2 numInstances . . . 1201
8.1280 voice_arrConnectPartyNum Struct Reference . . . 1201
8.1280.1 Detailed Description . . . 1201
8.1280.2 Field Documentation . . . 1201
8.1280.2.1 ConnectedPartyNum . . . 1201
8.1280.2.2 numInstances . . . 1201
8.1281 voice_arrDiagInfo Struct Reference . . . 1202
8.1281.1 Detailed Description . . . 1202
8.1281.2 Field Documentation . . . 1202
8.1281.2.1 DiagInfo . . . 1202
8.1281.2.2 numInstances . . . 1202
8.1282 voice_arrRedirPartyNum Struct Reference . . . 1202
8.1282.1 Detailed Description . . . 1202
8.1282.2 Field Documentation . . . 1202
8.1282.2.1 numInstances . . . 1203
8.1282.2.2 RedirPartyNum . . . 1203
8.1283 voice_arrRemotePartyName Struct Reference . . . 1203
8.1283.1 Detailed Description . . . 1203
8.1283.2 Field Documentation . . . 1203
8.1283.2.1 GetAllCallRmtPtyName . . . 1203
8.1283.2.2 numInstances . . . 1203
8.1284 voice_arrRemotePartyNum Struct Reference . . . 1203
8.1284.1 Detailed Description . . . 1203
8.1284.2 Field Documentation . . . 1204
8.1284.2.1 numInstances . . . 1204
8.1284.2.2 RmtPtyNum . . . 1204
8.1285 voice_arrSvcOption Struct Reference . . . 1204
8.1285.1 Detailed Description . . . 1204
8.1285.2 Field Documentation . . . 1204
8.1285.2.1 callID . . . 1204

8.1285.2.2 numInstances . . . 1204
8.1285.2.3 srvOption . . . 1204
8.1286 voice_arrUUSInfo Struct Reference . . . 1204
8.1286.1 Detailed Description . . . 1204
8.1286.2 Field Documentation . . . 1205
8.1286.2.1 AllCallsUUSInfo . . . 1205
8.1286.2.2 numInstances . . . 1205
8.1287 voice_burstDTMFInfo Struct Reference . . . 1205
8.1287.1 Detailed Description . . . 1205
8.1287.2 Field Documentation . . . 1205
8.1287.2.1 digitCnt . . . 1205
8.1287.2.2 pCallID . . . 1205
8.1287.2.3 pDigitBuff . . . 1205
8.1288 voice_calledPartySubAdd Struct Reference . . . 1206
8.1288.1 Detailed Description . . . 1206
8.1288.2 Field Documentation . . . 1206
8.1288.2.1 extBit . . . 1206
8.1288.2.2 oddEvenInd . . . 1206
8.1288.2.3 subAddr . . . 1206
8.1288.2.4 subAddrLen . . . 1206
8.1288.2.5 subAddrType . . . 1206
8.1289 voice_callFwdTypeAndPlan Struct Reference . . . 1206
8.1289.1 Detailed Description . . . 1207
8.1289.2 Field Documentation . . . 1207
8.1289.2.1 numberPlan . . . 1207
8.1289.2.2 numberType . . . 1207
8.1290 voice_callFWExtInfo Struct Reference . . . 1207
8.1290.1 Detailed Description . . . 1208
8.1290.2 Field Documentation . . . 1209
8.1290.2.1 noReplyTimer . . . 1209
8.1290.2.2 number . . . 1209
8.1290.2.3 numLen . . . 1209
8.1290.2.4 numPlan . . . 1209
8.1290.2.5 numType . . . 1209
8.1290.2.6 PI . . . 1209
8.1290.2.7 SI . . . 1209
8.1290.2.8 SvcClass . . . 1209
8.1290.2.9 SvcStatus . . . 1209
8.1291 voice_callFWInfo Struct Reference . . . 1209
8.1291.1 Detailed Description . . . 1210

8.1291.2 Field Documentation . . . 1210
8.1291.2.1 noReplyTimer . . . 1210
8.1291.2.2 number . . . 1210
8.1291.2.3 numLen . . . 1210
8.1291.2.4 SvcClass . . . 1210
8.1291.2.5 SvcStatus . . . 1210
8.1292 voice_callInfo Struct Reference . . . 1210
8.1292.1 Detailed Description . . . 1210
8.1292.2 Field Documentation . . . 1211
8.1292.2.1 callID . . . 1211
8.1292.2.2 callState . . . 1212
8.1292.2.3 callType . . . 1212
8.1292.2.4 direction . . . 1212
8.1292.2.5 mode . . . 1212
8.1293 voice_ccSUPSType Struct Reference . . . 1212
8.1293.1 Detailed Description . . . 1212
8.1293.2 Field Documentation . . . 1212
8.1293.2.1 reason . . . 1212
8.1293.2.2 svcType . . . 1212
8.1294 voice_CLIPResp Struct Reference . . . 1212
8.1294.1 Detailed Description . . . 1213
8.1294.2 Field Documentation . . . 1213
8.1294.2.1 ActiveStatus . . . 1213
8.1294.2.2 ProvisionStatus . . . 1213
8.1295 voice_CLIRResp Struct Reference . . . 1213
8.1295.1 Detailed Description . . . 1213
8.1295.2 Field Documentation . . . 1214
8.1295.2.1 ActiveStatus . . . 1214
8.1295.2.2 ProvisionStatus . . . 1214
8.1296 voice_CNAPResp Struct Reference . . . 1214
8.1296.1 Detailed Description . . . 1214
8.1296.2 Field Documentation . . . 1214
8.1296.2.1 ActiveStatus . . . 1214
8.1296.2.2 ProvisionStatus . . . 1214
8.1297 voice_COLPResp Struct Reference . . . 1215
8.1297.1 Detailed Description . . . 1215
8.1297.2 Field Documentation . . . 1215
8.1297.2.1 ActiveStatus . . . 1215
8.1297.2.2 ProvisionStatus . . . 1215
8.1298 voice_COLRResp Struct Reference . . . 1215

8.1298.1 Detailed Description . . . 1215
8.1298.2 Field Documentation . . . 1216
8.1298.2.1 ActiveStatus . . . 1216
8.1298.2.2 ProvisionStatus . . . 1216
8.1299 voice_connectNumInfo Struct Reference . . . 1216
8.1299.1 Detailed Description . . . 1216
8.1299.2 Field Documentation . . . 1217
8.1299.2.1 callerID . . . 1217
8.1299.2.2 callerIDLen . . . 1217
8.1299.2.3 numPlan . . . 1217
8.1299.2.4 numPresInd . . . 1218
8.1299.2.5 numType . . . 1218
8.1299.2.6 screeningInd . . . 1218
8.1300 voice_CUGInfo Struct Reference . . . 1218
8.1300.1 Detailed Description . . . 1218
8.1300.2 Field Documentation . . . 1218
8.1300.2.1 CUGIndex . . . 1218
8.1300.2.2 SuppOA . . . 1218
8.1300.2.3 SuppPrefCUG . . . 1218
8.1301 voice_curAMRConfig Struct Reference . . . 1218
8.1301.1 Detailed Description . . . 1218
8.1301.2 Field Documentation . . . 1219
8.1301.2.1 gsmAmrStat . . . 1219
8.1301.2.2 wcdmaAmrStat . . . 1219
8.1302 voice_diagInfo Struct Reference . . . 1219
8.1302.1 Detailed Description . . . 1219
8.1302.2 Field Documentation . . . 1219
8.1302.2.1 diagInfoLen . . . 1219
8.1302.2.2 diagnosticInfo . . . 1219
8.1303 voice_DTMFLengths Struct Reference . . . 1220
8.1303.1 Detailed Description . . . 1220
8.1303.2 Field Documentation . . . 1220
8.1303.2.1 DTMFInterdigitInterval . . . 1220
8.1303.2.2 DTMFPulseWidth . . . 1220
8.1304 voice_getAllCallInformation Struct Reference . . . 1220
8.1304.1 Detailed Description . . . 1220
8.1304.2 Field Documentation . . . 1221
8.1304.2.1 ALS . . . 1221
8.1304.2.2 Callinfo . . . 1221
8.1304.2.3 isEmpty . . . 1221

8.1305 voice_getAllCallRmtPtyName Struct Reference . . . 1221
8.1305.1 Detailed Description . . . 1221
8.1305.2 Field Documentation . . . 1221
8.1305.2.1 callID . . . 1221
8.1305.2.2 RemotePartyName . . . 1221
8.1306 voice_getAllCallRmtPtyNum Struct Reference . . . 1221
8.1306.1 Detailed Description . . . 1222
8.1306.2 Field Documentation . . . 1222
8.1306.2.1 callID . . . 1222
8.1306.2.2 RemotePartyNum . . . 1222
8.1307 voice_getCallFWExtInfo Struct Reference . . . 1222
8.1307.1 Detailed Description . . . 1222
8.1307.2 Field Documentation . . . 1222
8.1307.2.1 CallFWExtInfo . . . 1222
8.1307.2.2 numInstances . . . 1222
8.1308 voice_getCallFWInfo Struct Reference . . . 1223
8.1308.1 Detailed Description . . . 1223
8.1308.2 Field Documentation . . . 1223
8.1308.2.1 CallFWInfo . . . 1223
8.1308.2.2 numInstances . . . 1223
8.1309 voice_peerNumberInfo Struct Reference . . . 1223
8.1309.1 Detailed Description . . . 1223
8.1309.2 Field Documentation . . . 1225
8.1309.2.1 callID . . . 1225
8.1309.2.2 number . . . 1225
8.1309.2.3 numLen . . . 1225
8.1309.2.4 numPI . . . 1225
8.1309.2.5 numPlan . . . 1225
8.1309.2.6 numSI . . . 1225
8.1309.2.7 numType . . . 1225
8.1310 voice_prefVoiceSO Struct Reference . . . 1225
8.1310.1 Detailed Description . . . 1225
8.1310.2 Field Documentation . . . 1227
8.1310.2.1 evrcCapability . . . 1227
8.1310.2.2 homeOrigVoiceSO . . . 1227
8.1310.2.3 homePageVoiceSO . . . 1227
8.1310.2.4 namID . . . 1227
8.1310.2.5 roamOrigVoiceSO . . . 1227
8.1311 voice_remotePartyName Struct Reference . . . 1227
8.1311.1 Detailed Description . . . 1227

8.1311.2 Field Documentation . . . 1227
8.1311.2.1 callerName . . . 1228
8.1311.2.2 codingScheme . . . 1228
8.1311.2.3 nameLen . . . 1228
8.1311.2.4 namePI . . . 1228
8.1312 voice_remotePartyNum Struct Reference . . . 1228
8.1312.1 Detailed Description . . . 1228
8.1312.2 Field Documentation . . . 1228
8.1312.2.1 numLen . . . 1228
8.1312.2.2 presentationInd . . . 1228
8.1312.2.3 remPartyNumber . . . 1228
8.1313 voice_roamTimer Struct Reference . . . 1228
8.1313.1 Detailed Description . . . 1229
8.1313.2 Field Documentation . . . 1229
8.1313.2.1 namID . . . 1229
8.1313.2.2 roamTimerValue . . . 1229
8.1314 voice_USSInfo Struct Reference . . . 1229
8.1314.1 Detailed Description . . . 1229
8.1314.2 Field Documentation . . . 1229
8.1314.2.1 ussData . . . 1230
8.1314.2.2 ussDCS . . . 1230
8.1314.2.3 ussLen . . . 1230
8.1315 voice_UUSInfo Struct Reference . . . 1230
8.1315.1 Detailed Description . . . 1230
8.1315.2 Field Documentation . . . 1231
8.1315.2.1 UUSData . . . 1231
8.1315.2.2 UUSDatalen . . . 1231
8.1315.2.3 UUSDcs . . . 1231
8.1315.2.4 UUSType . . . 1231
8.1316 voiceALSSelectLineInfo Struct Reference . . . 1231
8.1316.1 Detailed Description . . . 1231
8.1316.2 Field Documentation . . . 1231
8.1316.2.1 lineValue . . . 1231
8.1317 voiceALSSetLineSwitchInfo Struct Reference . . . 1231
8.1317.1 Detailed Description . . . 1231
8.1317.2 Field Documentation . . . 1231
8.1317.2.1 switchOption . . . 1232
8.1318 voiceAnswerCall Struct Reference . . . 1232
8.1318.1 Detailed Description . . . 1232
8.1318.2 Field Documentation . . . 1232

8.1318.2.1 pCallId . . . 1232
8.1319 voiceBindSubscriptionInfo Struct Reference . . . 1232
8.1319.1 Detailed Description . . . 1232
8.1319.2 Field Documentation . . . 1232
8.1319.2.1 subsType . . . 1232
8.1320 voiceBurstDTMFInfo Struct Reference . . . 1232
8.1320.1 Detailed Description . . . 1233
8.1320.2 Field Documentation . . . 1233
8.1320.2.1 BurstDTMFInfo . . . 1233
8.1320.2.2 pBurstDTMFLengths . . . 1233
8.1321 voiceCallInfoReq Struct Reference . . . 1233
8.1321.1 Detailed Description . . . 1233
8.1321.2 Field Documentation . . . 1233
8.1321.2.1 callID . . . 1233
8.1322 voiceCallInfoResp Struct Reference . . . 1233
8.1322.1 Detailed Description . . . 1234
8.1322.2 Field Documentation . . . 1236
8.1322.2.1 pAlertingPattern . . . 1236
8.1322.2.2 pAlertType . . . 1236
8.1322.2.3 pAlphaIDInfo . . . 1236
8.1322.2.4 pCallInfo . . . 1236
8.1322.2.5 pConnectNumInfo . . . 1236
8.1322.2.6 pDiagInfo . . . 1236
8.1322.2.7 pOTASPStatus . . . 1236
8.1322.2.8 pRemotePartyName . . . 1236
8.1322.2.9 pRemotePartyNum . . . 1236
8.1322.2.10 pSrvOpt . . . 1236
8.1322.2.11 pUUSInfo . . . 1236
8.1322.2.12 pVoicePrivacy . . . 1236
8.1323 voiceCallRequestParams Struct Reference . . . 1236
8.1323.1 Detailed Description . . . 1237
8.1323.2 Field Documentation . . . 1238
8.1323.2.1 callNumber . . . 1238
8.1323.2.2 pCallPartySubAdd . . . 1238
8.1323.2.3 pCallType . . . 1238
8.1323.2.4 pCLIRType . . . 1238
8.1323.2.5 pCUGInfo . . . 1238
8.1323.2.6 pEmergencyCategory . . . 1238
8.1323.2.7 pSvcType . . . 1238
8.1323.2.8 pUUSInFo . . . 1238

8.1324 VoiceCallResponseParams Struct Reference . . . 1238
8.1324.1 Detailed Description . . . 1238
8.1324.2 Field Documentation . . . 1239
8.1324.2.1 pAlphaIDInfo . . . 1239
8.1324.2.2 pCallID . . . 1239
8.1324.2.3 pCCResultType . . . 1239
8.1324.2.4 pCCSUPSType . . . 1239
8.1325 VoiceContDTMFinfo Struct Reference . . . 1239
8.1325.1 Detailed Description . . . 1239
8.1325.2 Field Documentation . . . 1239
8.1325.2.1 DTMFdigit . . . 1239
8.1325.2.2 pCallID . . . 1240
8.1326 VoiceDTMFEventInfo Struct Reference . . . 1240
8.1326.1 Detailed Description . . . 1240
8.1326.2 Field Documentation . . . 1240
8.1326.2.1 DTMFInformation . . . 1240
8.1326.2.2 pOffLength . . . 1240
8.1326.2.3 pOnLength . . . 1240
8.1327 VoiceFlashInfo Struct Reference . . . 1240
8.1327.1 Detailed Description . . . 1241
8.1327.2 Field Documentation . . . 1241
8.1327.2.1 pCallID . . . 1241
8.1327.2.2 pFlashPayLd . . . 1241
8.1327.2.3 pFlashType . . . 1241
8.1328 VoiceGetAllCallInfo Struct Reference . . . 1241
8.1328.1 Detailed Description . . . 1242
8.1328.2 Field Documentation . . . 1243
8.1328.2.1 pArrAlertingPattern . . . 1243
8.1328.2.2 pArrAlertingType . . . 1243
8.1328.2.3 pArrAlphaID . . . 1243
8.1328.2.4 pArrCalledPartyNum . . . 1243
8.1328.2.5 pArrCallEndReason . . . 1243
8.1328.2.6 pArrCallInfo . . . 1243
8.1328.2.7 pArrConnectPartyNum . . . 1243
8.1328.2.8 pArrDiagInfo . . . 1243
8.1328.2.9 pArrRedirPartyNum . . . 1243
8.1328.2.10 pArrRemotePartyName . . . 1243
8.1328.2.11 pArrRemotePartyNum . . . 1243
8.1328.2.12 pArrSvcOption . . . 1244
8.1328.2.13 pArrUUSInfo . . . 1244

8.1328.2.14 pOTASPStatus . . . 1244
8.1328.2.15 pVoicePrivacy . . . 1244
8.1329 VoiceGetCallBarringReq Struct Reference . . . 1244
8.1329.1 Detailed Description . . . 1244
8.1329.2 Field Documentation . . . 1244
8.1329.2.1 pSvcClass . . . 1244
8.1329.2.2 Reason . . . 1244
8.1330 VoiceGetCallBarringResp Struct Reference . . . 1244
8.1330.1 Detailed Description . . . 1245
8.1330.2 Field Documentation . . . 1245
8.1330.2.1 pAlphaIDInfo . . . 1246
8.1330.2.2 pCallID . . . 1246
8.1330.2.3 pCCResType . . . 1246
8.1330.2.4 pCCSUPSType . . . 1246
8.1330.2.5 pFailCause . . . 1246
8.1330.2.6 pSvcClass . . . 1246
8.1331 VoiceGetCallFWReq Struct Reference . . . 1246
8.1331.1 Detailed Description . . . 1246
8.1331.2 Field Documentation . . . 1246
8.1331.2.1 pSvcClass . . . 1246
8.1331.2.2 Reason . . . 1246
8.1332 VoiceGetCallFWResp Struct Reference . . . 1246
8.1332.1 Detailed Description . . . 1247
8.1332.2 Field Documentation . . . 1248
8.1332.2.1 pAlphaIDInfo . . . 1248
8.1332.2.2 pCallID . . . 1248
8.1332.2.3 pCCResType . . . 1248
8.1332.2.4 pCCSUPSType . . . 1248
8.1332.2.5 pFailCause . . . 1248
8.1332.2.6 pGetCallFWExtInfo . . . 1248
8.1332.2.7 pGetCallFWInfo . . . 1248
8.1333 VoiceGetCallWaitInfo Struct Reference . . . 1248
8.1333.1 Detailed Description . . . 1248
8.1333.2 Field Documentation . . . 1249
8.1333.2.1 pAlphaIDInfo . . . 1249
8.1333.2.2 pCallID . . . 1249
8.1333.2.3 pCCResType . . . 1249
8.1333.2.4 pCCSUPSType . . . 1249
8.1333.2.5 pFailCause . . . 1249
8.1333.2.6 pSvcClass . . . 1249

8.1334 VoiceGetCLIPResp Struct Reference . . . 1249
8.1334.1 Detailed Description . . . 1249
8.1334.2 Field Documentation . . . 1250
8.1334.2.1 pAlphaIDInfo . . . 1250
8.1334.2.2 pCallID . . . 1250
8.1334.2.3 pCCResType . . . 1250
8.1334.2.4 pCCSUPSType . . . 1250
8.1334.2.5 pCLIPResp . . . 1250
8.1334.2.6 pFailCause . . . 1250
8.1335 VoiceGetCLIRResp Struct Reference . . . 1250
8.1335.1 Detailed Description . . . 1251
8.1335.2 Field Documentation . . . 1251
8.1335.2.1 pAlphaIDInfo . . . 1251
8.1335.2.2 pCallID . . . 1252
8.1335.2.3 pCCResType . . . 1252
8.1335.2.4 pCCSUPSType . . . 1252
8.1335.2.5 pCLIRResp . . . 1252
8.1335.2.6 pFailCause . . . 1252
8.1336 VoiceGetCNAPResp Struct Reference . . . 1252
8.1336.1 Detailed Description . . . 1252
8.1336.2 Field Documentation . . . 1253
8.1336.2.1 pAlphaIDInfo . . . 1253
8.1336.2.2 pCallID . . . 1253
8.1336.2.3 pCCResType . . . 1253
8.1336.2.4 pCCSUPSType . . . 1253
8.1336.2.5 pCNAPResp . . . 1253
8.1336.2.6 pFailCause . . . 1253
8.1337 VoiceGetCOLPResp Struct Reference . . . 1253
8.1337.1 Detailed Description . . . 1253
8.1337.2 Field Documentation . . . 1254
8.1337.2.1 pAlphaIDInfo . . . 1254
8.1337.2.2 pCallID . . . 1254
8.1337.2.3 pCCResType . . . 1254
8.1337.2.4 pCCSUPSType . . . 1254
8.1337.2.5 pCOLPResp . . . 1254
8.1337.2.6 pFailCause . . . 1254
8.1338 VoiceGetCOLRResp Struct Reference . . . 1254
8.1338.1 Detailed Description . . . 1254
8.1338.2 Field Documentation . . . 1255
8.1338.2.1 pAlphaIDInfo . . . 1255

8.1338.2.2 pCallID . . . 1255
8.1338.2.3 pCCResType . . . 1255
8.1338.2.4 pCCSUPSType . . . 1255
8.1338.2.5 pCOLRResp . . . 1255
8.1338.2.6 pFailCause . . . 1255
8.1339 VoiceGetConfigReq Struct Reference . . . 1255
8.1339.1 Detailed Description . . . 1256
8.1339.2 Field Documentation . . . 1257
8.1339.2.1 pAirTimer . . . 1257
8.1339.2.2 pAMRStatus . . . 1257
8.1339.2.3 pAutoAnswer . . . 1257
8.1339.2.4 pNamID . . . 1257
8.1339.2.5 pPrefVoicePrivacy . . . 1257
8.1339.2.6 pPrefVoiceSO . . . 1257
8.1339.2.7 pRoamTimer . . . 1257
8.1339.2.8 pTTYMode . . . 1257
8.1339.2.9 pVoiceDomainPref . . . 1257
8.1340 VoiceGetConfigResp Struct Reference . . . 1257
8.1340.1 Detailed Description . . . 1258
8.1340.2 Field Documentation . . . 1259
8.1340.2.1 pAirTimerCnt . . . 1259
8.1340.2.2 pAutoAnswerStat . . . 1259
8.1340.2.3 pCurAMRConfig . . . 1259
8.1340.2.4 pCurPrefVoiceSO . . . 1259
8.1340.2.5 pCurrTTYMode . . . 1259
8.1340.2.6 pCurVoiceDomainPref . . . 1259
8.1340.2.7 pCurVoicePrivacyPref . . . 1259
8.1340.2.8 pRoamTimerCnt . . . 1259
8.1341 VoiceIndicationRegisterInfo Struct Reference . . . 1259
8.1341.1 Detailed Description . . . 1259
8.1341.2 Field Documentation . . . 1260
8.1341.2.1 pRegDTMFEvents . . . 1260
8.1341.2.2 pRegVoicePrivacyEvents . . . 1260
8.1341.2.3 pSuppsNotifEvents . . . 1260
8.1342 VoiceInfoRec Struct Reference . . . 1260
8.1342.1 Detailed Description . . . 1260
8.1342.2 Field Documentation . . . 1262
8.1342.2.1 callID . . . 1262
8.1342.2.2 pCalledPartyInfo . . . 1262
8.1342.2.3 pCallerIDInfo . . . 1262

8.1342.2.4 pCallerNameInfo . . . 1262
8.1342.2.5 pCallingPartyInfo . . . 1262
8.1342.2.6 pCallWaitInd . . . 1262
8.1342.2.7 pCLIRCause . . . 1262
8.1342.2.8 pConnectNumInfo . . . 1262
8.1342.2.9 pDispInfo . . . 1262
8.1342.2.10 pExtDispInfo . . . 1262
8.1342.2.11 pExtDispRecInfo . . . 1262
8.1342.2.12 pLineCtrlInfo . . . 1262
8.1342.2.13 pNSSAudioCtrl . . . 1262
8.1342.2.14 pNSSRelease . . . 1262
8.1342.2.15 pRedirNumInfo . . . 1262
8.1342.2.16 pSignalInfo . . . 1262
8.1343 VoiceManageCallsReq Struct Reference . . . 1262
8.1343.1 Detailed Description . . . 1262
8.1343.2 Field Documentation . . . 1263
8.1343.2.1 pCallID . . . 1263
8.1343.2.2 SUPSType . . . 1263
8.1344 VoiceManageCallsResp Struct Reference . . . 1263
8.1344.1 Detailed Description . . . 1263
8.1344.2 Field Documentation . . . 1264
8.1344.2.1 pFailCause . . . 1264
8.1345 VoiceOrigUSSDNoWaitInfo Struct Reference . . . 1264
8.1345.1 Detailed Description . . . 1264
8.1345.2 Field Documentation . . . 1264
8.1345.2.1 USSInformation . . . 1264
8.1346 VoiceOTASPStatusInfo Struct Reference . . . 1264
8.1346.1 Detailed Description . . . 1264
8.1346.2 Field Documentation . . . 1265
8.1346.2.1 callID . . . 1265
8.1346.2.2 OTASPStatus . . . 1265
8.1347 VoicePrivacyInfo Struct Reference . . . 1265
8.1347.1 Detailed Description . . . 1265
8.1347.2 Field Documentation . . . 1266
8.1347.2.1 callID . . . 1266
8.1347.2.2 voicePrivacy . . . 1266
8.1348 VoiceSetAllCallStatusCbkInfo Struct Reference . . . 1266
8.1348.1 Detailed Description . . . 1266
8.1348.2 Field Documentation . . . 1268
8.1348.2.1 arrCallInfomation . . . 1268

8.1348.2.2 pArrAlertingPattern . . . 1268
8.1348.2.3 pArrAlertingType . . . 1268
8.1348.2.4 pArrAlphaID . . . 1268
8.1348.2.5 pArrCalledPartyNum . . . 1268
8.1348.2.6 pArrCallEndReason . . . 1268
8.1348.2.7 pArrConnectPartyNum . . . 1268
8.1348.2.8 pArrDiagInfo . . . 1268
8.1348.2.9 pArrRedirPartyNum . . . 1268
8.1348.2.10 pArrRemotePartyName . . . 1268
8.1348.2.11 pArrRemotePartyNum . . . 1268
8.1348.2.12 pArrSvcOption . . . 1268
8.1349 VoiceSetCallBarringPwdInfo Struct Reference . . . 1268
8.1349.1 Detailed Description . . . 1268
8.1349.2 Field Documentation . . . 1269
8.1349.2.1 newPasswd . . . 1269
8.1349.2.2 newPasswdAgain . . . 1269
8.1349.2.3 oldPasswd . . . 1269
8.1349.2.4 Reason . . . 1269
8.1350 VoiceSetCallBarringPwdResp Struct Reference . . . 1269
8.1350.1 Detailed Description . . . 1270
8.1350.2 Field Documentation . . . 1270
8.1350.2.1 pAlphaIDInfo . . . 1270
8.1350.2.2 pCallID . . . 1270
8.1350.2.3 pCCResType . . . 1270
8.1350.2.4 pCCSUPSType . . . 1270
8.1350.2.5 pFailCause . . . 1270
8.1351 VoiceSetConfigReq Struct Reference . . . 1270
8.1351.1 Detailed Description . . . 1271
8.1351.2 Field Documentation . . . 1271
8.1351.2.1 pAirTimerConfig . . . 1272
8.1351.2.2 pAutoAnswer . . . 1272
8.1351.2.3 pPrefVoiceDomain . . . 1272
8.1351.2.4 pPrefVoiceSO . . . 1272
8.1351.2.5 pRoamTimerConfig . . . 1272
8.1351.2.6 pTTYMode . . . 1272
8.1352 VoiceSetConfigResp Struct Reference . . . 1272
8.1352.1 Detailed Description . . . 1272
8.1352.2 Field Documentation . . . 1273
8.1352.2.1 pAirTimerStatus . . . 1273
8.1352.2.2 pAutoAnsStatus . . . 1273

8.1352.2.3 pPrefVoiceSOStatus . . . 1273
8.1352.2.4 pRoamTimerStatus . . . 1273
8.1352.2.5 pTTYConfigStatus . . . 1273
8.1352.2.6 pVoiceDomainPrefStatus . . . 1273
8.1353 VoiceSetPrefPrivacy Struct Reference . . . 1273
8.1353.1 Detailed Description . . . 1273
8.1353.2 Field Documentation . . . 1274
8.1353.2.1 privacyPref . . . 1274
8.1354 VoiceSetSUPSServiceReq Struct Reference . . . 1274
8.1354.1 Detailed Description . . . 1274
8.1354.2 Field Documentation . . . 1276
8.1354.2.1 pCallBarringPasswd . . . 1276
8.1354.2.2 pCallForwardingNumber . . . 1276
8.1354.2.3 pCallFwdTypeAndPlan . . . 1276
8.1354.2.4 pServiceClass . . . 1276
8.1354.2.5 pTimerVal . . . 1276
8.1354.2.6 reason . . . 1276
8.1354.2.7 voiceSvc . . . 1276
8.1355 VoiceSetSUPSServiceResp Struct Reference . . . 1276
8.1355.1 Detailed Description . . . 1276
8.1355.2 Field Documentation . . . 1277
8.1355.2.1 pAlphaIDInfo . . . 1277
8.1355.2.2 pCallID . . . 1277
8.1355.2.3 pCCResultType . . . 1277
8.1355.2.4 pCCSUPSType . . . 1277
8.1355.2.5 pFailCause . . . 1277
8.1356 VoiceStopContDTMFinfo Struct Reference . . . 1277
8.1356.1 Detailed Description . . . 1277
8.1356.2 Field Documentation . . . 1278
8.1356.2.1 callID . . . 1278
8.1357 VoiceSUPSInfo Struct Reference . . . 1278
8.1357.1 Detailed Description . . . 1278
8.1357.2 Field Documentation . . . 1280
8.1357.2.1 pAlphaIDInfo . . . 1280
8.1357.2.2 pCallBarPasswd . . . 1280
8.1357.2.3 pCallFwdInfo . . . 1280
8.1357.2.4 pCallFWNum . . . 1280
8.1357.2.5 pCallFWTimerVal . . . 1280
8.1357.2.6 pCallID . . . 1280
8.1357.2.7 pCLIPstatus . . . 1280

8.1357.2.8 pCLIRstatus . . . 1280
8.1357.2.9 pCNAPstatus . . . 1280
8.1357.2.10 pCOLPstatus . . . 1280
8.1357.2.11 pCOLRstatus . . . 1280
8.1357.2.12 pDataSrc . . . 1280
8.1357.2.13 pFailCause . . . 1280
8.1357.2.14 pNewPwdData . . . 1280
8.1357.2.15 pReason . . . 1280
8.1357.2.16 pSvcClass . . . 1280
8.1357.2.17 pUSSInfo . . . 1280
8.1357.2.18 SUPSInformation . . . 1280
8.1358 VoiceSUPSNotification Struct Reference . . . 1280
8.1358.1 Detailed Description . . . 1281
8.1358.2 Field Documentation . . . 1282
8.1358.2.1 callID . . . 1282
8.1358.2.2 notifType . . . 1282
8.1358.2.3 pCUGIndex . . . 1282
8.1358.2.4 pECTNum . . . 1282
8.1359 WcdmaCellInfo Struct Reference . . . 1282
8.1359.1 Detailed Description . . . 1282
8.1359.2 Field Documentation . . . 1282
8.1359.2.1 cpich_ecno . . . 1282
8.1359.2.2 cpich_rscp . . . 1282
8.1359.2.3 psc . . . 1283
8.1359.2.4 srxlev . . . 1283
8.1360 WCDMAECIOThresh Struct Reference . . . 1283
8.1360.1 Detailed Description . . . 1283
8.1360.2 Field Documentation . . . 1283
8.1360.2.1 pWCDMAECIOThreshList . . . 1283
8.1360.2.2 WCDMAECIOThreshListLen . . . 1283
8.1361 WCDMAInfoLTENeighborCell Struct Reference . . . 1283
8.1361.1 Detailed Description . . . 1283
8.1361.2 Field Documentation . . . 1284
8.1361.2.1 UMTSLTENbrCell . . . 1284
8.1361.2.2 umtsLTENbrCellLen . . . 1284
8.1361.2.3 wcdmaRRCState . . . 1284
8.1362 WcdmaLongMsgDecodingParams Struct Reference . . . 1284
8.1362.1 Detailed Description . . . 1284
8.1362.2 Field Documentation . . . 1285
8.1362.2.1 Date . . . 1285

8.1362.2.2 pIsUDHPresent . . . 1285
8.1362.2.3 pMessage . . . 1286
8.1362.2.4 pPartNum . . . 1286
8.1362.2.5 pReferenceNum . . . 1286
8.1362.2.6 pScAddr . . . 1286
8.1362.2.7 pScAddrLength . . . 1286
8.1362.2.8 pSenderAddr . . . 1286
8.1362.2.9 pSenderAddrLength . . . 1286
8.1362.2.10 pTextMsg . . . 1286
8.1362.2.11 pTextMsgLength . . . 1286
8.1362.2.12 pTotalNum . . . 1286
8.1362.2.13 Time . . . 1286
8.1363 WcdmaMsgDecodingParams Struct Reference . . . 1286
8.1363.1 Detailed Description . . . 1286
8.1363.2 Field Documentation . . . 1287
8.1363.2.1 Date . . . 1287
8.1363.2.2 pMessage . . . 1287
8.1363.2.3 pScAddr . . . 1287
8.1363.2.4 pScAddrLength . . . 1287
8.1363.2.5 pSenderAddr . . . 1287
8.1363.2.6 pSenderAddrLength . . . 1287
8.1363.2.7 pTextMsg . . . 1287
8.1363.2.8 pTextMsgLength . . . 1287
8.1363.2.9 Time . . . 1287
8.1364 WcdmaMsgEncodingParams Struct Reference . . . 1287
8.1364.1 Detailed Description . . . 1288
8.1364.2 Field Documentation . . . 1288
8.1364.2.1 alphabet . . . 1288
8.1364.2.2 messageSize . . . 1288
8.1364.2.3 pDestAddr . . . 1288
8.1364.2.4 pPDUMessage . . . 1288
8.1364.2.5 pTextMsg . . . 1288
8.1365 WCDMARSSIThresh Struct Reference . . . 1288
8.1365.1 Detailed Description . . . 1288
8.1365.2 Field Documentation . . . 1289
8.1365.2.1 pWCDMARSSIThreshList . . . 1289
8.1365.2.2 WCDMARSSIThreshListLen . . . 1289
8.1366 WCDMASysInfo Struct Reference . . . 1289
8.1366.1 Detailed Description . . . 1289
8.1366.2 Field Documentation . . . 1291

8.1366.2.1 cellId . . . 1292
8.1366.2.2 cellIdValid . . . 1292
8.1366.2.3 hsCallStatus . . . 1292
8.1366.2.4 hsCallStatusValid . . . 1292
8.1366.2.5 hsInd . . . 1292
8.1366.2.6 hsIndValid . . . 1292
8.1366.2.7 lac . . . 1292
8.1366.2.8 lacValid . . . 1292
8.1366.2.9 MCC . . . 1292
8.1366.2.10 MNC . . . 1292
8.1366.2.11 networkIdValid . . . 1292
8.1366.2.12 psc . . . 1292
8.1366.2.13 pscValid . . . 1292
8.1366.2.14 regRejectInfoValid . . . 1292
8.1366.2.15 rejCause . . . 1292
8.1366.2.16 rejectSrvDomain . . . 1292
8.1366.2.17 sysInfoWCDMA . . . 1292
8.1367 wcdmaUARFCN Struct Reference . . . 1292
8.1367.1 Detailed Description . . . 1292
8.1367.2 Field Documentation . . . 1293
8.1367.2.1 status . . . 1293
8.1367.2.2 uarfcn . . . 1293
8.1368 wds_currNetworkInfo Struct Reference . . . 1293
8.1368.1 Detailed Description . . . 1293
8.1368.2 Field Documentation . . . 1294
8.1368.2.1 NetworkType . . . 1294
8.1368.2.2 RATMask . . . 1294
8.1368.2.3 SOMask . . . 1294
8.1369 wds_DataULongLongTlv Struct Reference . . . 1294
8.1369.1 Field Documentation . . . 1294
8.1369.1.1 TlvPresent . . . 1294
8.1369.1.2 ullData . . . 1294
8.1370 wds_DataULongTlv Struct Reference . . . 1294
8.1370.1 Field Documentation . . . 1295
8.1370.1.1 TlvPresent . . . 1295
8.1370.1.2 ulData . . . 1295
8.1371 wds_DHCPLeaseOptTlv Struct Reference . . . 1295
8.1371.1 Field Documentation . . . 1295
8.1371.1.1 numOpt . . . 1295
8.1371.1.2 optList . . . 1295

8.1371.1.3 optListData . . . 1295
8.1371.1.4 TlvPresent . . . 1295
8.1372 wds_DHCPLeaseStateTlv Struct Reference . . . 1295
8.1372.1 Field Documentation . . . 1295
8.1372.1.1 leaseState . . . 1295
8.1372.1.2 TlvPresent . . . 1295
8.1373 wds_DHCPOpt Struct Reference . . . 1295
8.1373.1 Field Documentation . . . 1296
8.1373.1.1 optCode . . . 1296
8.1373.1.2 optValLen . . . 1296
8.1373.1.3 pOptVal . . . 1296
8.1374 wds_DHCPProfileIdTlv Struct Reference . . . 1296
8.1374.1 Field Documentation . . . 1296
8.1374.1.1 profileId . . . 1296
8.1374.1.2 profileType . . . 1296
8.1374.1.3 TlvPresent . . . 1296
8.1375 wds_DHCPv4HWConfig Struct Reference . . . 1296
8.1375.1 Detailed Description . . . 1296
8.1375.2 Field Documentation . . . 1296
8.1375.2.1 chaddr . . . 1296
8.1375.2.2 chaddrLen . . . 1297
8.1375.2.3 hwType . . . 1297
8.1376 wds_DHCPv4Option Struct Reference . . . 1297
8.1376.1 Detailed Description . . . 1297
8.1376.2 Field Documentation . . . 1297
8.1376.2.1 optCode . . . 1297
8.1376.2.2 optVal . . . 1297
8.1376.2.3 optValLen . . . 1297
8.1377 wds_DHCPv4OptionList Struct Reference . . . 1297
8.1377.1 Detailed Description . . . 1297
8.1377.2 Field Documentation . . . 1298
8.1377.2.1 numOpt . . . 1298
8.1377.2.2 pOptList . . . 1298
8.1378 wds_DHCPv4ProfileId Struct Reference . . . 1298
8.1378.1 Detailed Description . . . 1298
8.1378.2 Field Documentation . . . 1298
8.1378.2.1 profileId . . . 1298
8.1378.2.2 profileType . . . 1298
8.1379 wds_Domain Struct Reference . . . 1298
8.1379.1 Detailed Description . . . 1298

8.1379.2 Field Documentation . . . 1299
8.1379.2.1 domainLen . . . 1299
8.1379.2.2 domainName . . . 1299
8.1380 wds_DomainNameList Struct Reference . . . 1299
8.1380.1 Detailed Description . . . 1299
8.1380.2 Field Documentation . . . 1299
8.1380.2.1 domain . . . 1299
8.1380.2.2 numInstances . . . 1299
8.1381 wds_GPRSQoS Struct Reference . . . 1299
8.1381.1 Detailed Description . . . 1300
8.1381.2 Field Documentation . . . 1300
8.1381.2.1 delayClass . . . 1300
8.1381.2.2 meanThroughputClass . . . 1300
8.1381.2.3 peakThroughputClass . . . 1300
8.1381.2.4 precedenceClass . . . 1300
8.1381.2.5 reliabilityClass . . . 1300
8.1382 wds_IPv4AdTlv Struct Reference . . . 1300
8.1382.1 Field Documentation . . . 1300
8.1382.1.1 IPv4Addr . . . 1300
8.1382.1.2 TlvPresent . . . 1300
8.1383 wds_IPV6AddressInfo Struct Reference . . . 1300
8.1383.1 Detailed Description . . . 1301
8.1383.2 Field Documentation . . . 1301
8.1383.2.1 IPAddressV6 . . . 1301
8.1383.2.2 IPV6PrefixLen . . . 1301
8.1384 wds_IPV6GWAddressInfo Struct Reference . . . 1301
8.1384.1 Detailed Description . . . 1301
8.1384.2 Field Documentation . . . 1301
8.1384.2.1 gwAddressV6 . . . 1301
8.1384.2.2 gwV6PrefixLen . . . 1301
8.1385 wds_PCSCFFQDNAddress Struct Reference . . . 1301
8.1385.1 Detailed Description . . . 1302
8.1385.2 Field Documentation . . . 1302
8.1385.2.1 fqdnAddr . . . 1302
8.1385.2.2 fqdnLen . . . 1302
8.1386 wds_PCSCFFQDNAddressList Struct Reference . . . 1302
8.1386.1 Detailed Description . . . 1302
8.1386.2 Field Documentation . . . 1302
8.1386.2.1 numInstances . . . 1302
8.1386.2.2 pcsfFQDNAddress . . . 1302

8.1387 wds_PCSCFIPv4ServerAddressList Struct Reference . . . 1302
8.1387.1 Detailed Description . . . 1303
8.1387.2 Field Documentation . . . 1303
8.1387.2.1 numInstances . . . 1303
8.1387.2.2 pscsfIPv4Addr . . . 1303
8.1388 wds_ProfileIdentifier Struct Reference . . . 1303
8.1388.1 Detailed Description . . . 1303
8.1388.2 Field Documentation . . . 1303
8.1388.2.1 profileIndex . . . 1303
8.1388.2.2 profileType . . . 1303
8.1389 wds_profileInfo Union Reference . . . 1303
8.1389.1 Detailed Description . . . 1304
8.1389.2 Field Documentation . . . 1304
8.1389.2.1 SlqsProfile3GPP . . . 1304
8.1389.2.2 SlqsProfile3GPP2 . . . 1304
8.1390 wds_TrStatInd Struct Reference . . . 1304
8.1390.1 Detailed Description . . . 1304
8.1390.2 Field Documentation . . . 1304
8.1390.2.1 statsMask . . . 1304
8.1390.2.2 statsPeriod . . . 1304
8.1391 wds_UMTSMinQoS Struct Reference . . . 1304
8.1391.1 Detailed Description . . . 1305
8.1391.2 Field Documentation . . . 1306
8.1391.2.1 deliveryErrSDU . . . 1306
8.1391.2.2 grntDownlinkBitrate . . . 1307
8.1391.2.3 grntUplinkBitrate . . . 1307
8.1391.2.4 maxDownlinkBitrate . . . 1307
8.1391.2.5 maxSDUSize . . . 1307
8.1391.2.6 maxUplinkBitrate . . . 1307
8.1391.2.7 qosDeliveryOrder . . . 1307
8.1391.2.8 residBerRatio . . . 1307
8.1391.2.9 sduErrorRatio . . . 1307
8.1391.2.10 trafficClass . . . 1307
8.1391.2.11 trafficPriority . . . 1307
8.1391.2.12 transferDelay . . . 1307
8.1392 WdsByteTotals Struct Reference . . . 1307
8.1392.1 Detailed Description . . . 1307
8.1392.2 Field Documentation . . . 1308
8.1392.2.1 ByteTotalsElmntsV4 . . . 1308
8.1392.2.2 ByteTotalsElmntsV6 . . . 1308

8.1392.2.3 pV4sessionId . . . 1308
8.1392.2.4 pV6sessionId . . . 1308
8.1393 WdsByteTotalsElmnts Struct Reference . . . 1308
8.1393.1 Detailed Description . . . 1308
8.1393.2 Field Documentation . . . 1308
8.1393.2.1 pRXTotalBytes . . . 1308
8.1393.2.2 pTXTotalBytes . . . 1308
8.1394 WdsClientLeaseChange Struct Reference . . . 1308
8.1394.1 Detailed Description . . . 1308
8.1394.2 Field Documentation . . . 1309
8.1394.2.1 pEnableNotification . . . 1309
8.1395 WdsConnectionRate Struct Reference . . . 1309
8.1395.1 Detailed Description . . . 1309
8.1395.2 Field Documentation . . . 1309
8.1395.2.1 ConnRateElmntsV4 . . . 1309
8.1395.2.2 ConnRateElmntsV6 . . . 1309
8.1395.2.3 pV4sessionId . . . 1309
8.1395.2.4 pV6sessionId . . . 1309
8.1396 WdsConnectionRateElmnts Struct Reference . . . 1310
8.1396.1 Detailed Description . . . 1310
8.1396.2 Field Documentation . . . 1310
8.1396.2.1 pCurrentChannelRXRate . . . 1310
8.1396.2.2 pCurrentChannelTXRate . . . 1310
8.1396.2.3 pMaxChannelRXRate . . . 1310
8.1396.2.4 pMaxChannelTXRate . . . 1310
8.1397 WdsDHCPv4ClientLeaseInd Struct Reference . . . 1310
8.1397.1 Detailed Description . . . 1310
8.1397.2 Field Documentation . . . 1311
8.1397.2.1 pIPv4Addr . . . 1311
8.1397.2.2 pLeaseState . . . 1311
8.1397.2.3 pOptList . . . 1311
8.1397.2.4 pProfileId . . . 1311
8.1398 WdsDHCPv4Config Struct Reference . . . 1311
8.1398.1 Detailed Description . . . 1311
8.1398.2 Field Documentation . . . 1312
8.1398.2.1 pHwConfig . . . 1312
8.1398.2.2 pProfileId . . . 1312
8.1398.2.3 pRequestOptionList . . . 1312
8.1399 wdsDhcpv4HwConfig Struct Reference . . . 1312
8.1399.1 Detailed Description . . . 1312

8.1399.2 Field Documentation . . . 1312
8.1399.2.1 chaddr . . . 1312
8.1399.2.2 chaddrLen . . . 1312
8.1399.2.3 hwType . . . 1312
8.1400 WdsDHCPv4HWConfig Struct Reference . . . 1312
8.1400.1 Detailed Description . . . 1313
8.1400.2 Field Documentation . . . 1313
8.1400.2.1 chaddr . . . 1313
8.1400.2.2 chaddrLen . . . 1313
8.1400.2.3 hwType . . . 1313
8.1401 wdsDhcpv4Option Struct Reference . . . 1313
8.1401.1 Detailed Description . . . 1313
8.1401.2 Field Documentation . . . 1313
8.1401.2.1 optCode . . . 1313
8.1401.2.2 optVal . . . 1313
8.1401.2.3 optValLen . . . 1313
8.1402 WdsDHCPv4Option Struct Reference . . . 1314
8.1402.1 Detailed Description . . . 1314
8.1402.2 Field Documentation . . . 1314
8.1402.2.1 optCode . . . 1314
8.1402.2.2 optVal . . . 1314
8.1402.2.3 optValLen . . . 1314
8.1403 WdsDHCPv4OptionList Struct Reference . . . 1314
8.1403.1 Detailed Description . . . 1314
8.1403.2 Field Documentation . . . 1315
8.1403.2.1 numOpt . . . 1315
8.1403.2.2 pOptList . . . 1315
8.1404 wdsDhcpv4OptionList Struct Reference . . . 1315
8.1404.1 Detailed Description . . . 1315
8.1404.2 Field Documentation . . . 1315
8.1404.2.1 numOpt . . . 1315
8.1404.2.2 pOptList . . . 1315
8.1405 WdsDHCPv4ProfileId Struct Reference . . . 1315
8.1405.1 Detailed Description . . . 1315
8.1405.2 Field Documentation . . . 1315
8.1405.2.1 profileId . . . 1315
8.1405.2.2 profileType . . . 1316
8.1406 wdsDhcpv4ProfileId Struct Reference . . . 1316
8.1406.1 Detailed Description . . . 1316
8.1406.2 Field Documentation . . . 1316

8.1406.2.1 profileId . . . 1316
8.1406.2.2 profileType . . . 1316
8.1407 WDSGetLoopbackData Struct Reference . . . 1316
8.1407.1 Detailed Description . . . 1316
8.1407.2 Field Documentation . . . 1317
8.1407.2.1 ByteLoopbackMode . . . 1317
8.1407.2.2 ByteLoopbackMultiplier . . . 1317
8.1408 WdsIpAddressInfoReq Struct Reference . . . 1317
8.1408.1 Field Documentation . . . 1317
8.1408.1.1 ip . . . 1317
8.1408.1.2 pv4sessionId . . . 1317
8.1408.1.3 pv6sessionId . . . 1317
8.1409 WdsPktStatisticsElmnts Struct Reference . . . 1317
8.1409.1 Detailed Description . . . 1318
8.1409.2 Field Documentation . . . 1318
8.1409.2.1 pRXDroppedCount . . . 1318
8.1409.2.2 pRXOkBytesCount . . . 1318
8.1409.2.3 pRXOKBytesLastCall . . . 1318
8.1409.2.4 pRXPacketErrors . . . 1319
8.1409.2.5 pRXPacketOverflows . . . 1319
8.1409.2.6 pRXPacketSuccesses . . . 1319
8.1409.2.7 pTXDroppedCount . . . 1319
8.1409.2.8 pTXOkBytesCount . . . 1319
8.1409.2.9 pTXOKBytesLastCall . . . 1319
8.1409.2.10 pTXPacketErrors . . . 1319
8.1409.2.11 pTXPacketOverflows . . . 1319
8.1409.2.12 pTXPacketSuccesses . . . 1319
8.1410 WdsPktStatisticsReq Struct Reference . . . 1319
8.1410.1 Detailed Description . . . 1319
8.1410.2 Field Documentation . . . 1319
8.1410.2.1 pStatMask . . . 1319
8.1411 WdsPktStatisticsResp Struct Reference . . . 1319
8.1411.1 Detailed Description . . . 1319
8.1411.2 Field Documentation . . . 1320
8.1411.2.1 PktStatElmntsV4 . . . 1320
8.1411.2.2 PktStatElmntsV6 . . . 1320
8.1411.2.3 pV4sessionId . . . 1320
8.1411.2.4 pV6sessionId . . . 1320
8.1412 WdsProfileParam Union Reference . . . 1320
8.1412.1 Detailed Description . . . 1320

8.1412.2 Field Documentation . . . 1320
8.1412.2.1 SlqsProfile3GPP . . . 1320
8.1412.2.2 SlqsProfile3GPP2 . . . 1320
8.1413 WdsRunTimeSettings Struct Reference . . . 1320
8.1413.1 Detailed Description . . . 1321
8.1413.2 Field Documentation . . . 1321
8.1413.2.1 rts . . . 1321
8.1413.2.2 v4sessionId . . . 1321
8.1413.2.3 v6sessionId . . . 1321
8.1414 WdsSetEventReportReq Struct Reference . . . 1321
8.1414.1 Detailed Description . . . 1321
8.1414.2 Field Documentation . . . 1323
8.1414.2.1 pCurrChannelRateInd . . . 1323
8.1414.2.2 pCurrDataBearerTechInd . . . 1323
8.1414.2.3 pCurrPrefDataSysInd . . . 1323
8.1414.2.4 pDataBearerTechInd . . . 1323
8.1414.2.5 pDataCallStatusChangeInd . . . 1323
8.1414.2.6 pDataSystemStatusChangeInd . . . 1323
8.1414.2.7 pDormancyStatusInd . . . 1323
8.1414.2.8 pEVDOPageMonPerChangeInd . . . 1323
8.1414.2.9 pMIPStatusInd . . . 1323
8.1414.2.10 pTransferStatInd . . . 1323
8.1415 WDSSetLoopbackData Struct Reference . . . 1323
8.1415.1 Detailed Description . . . 1323
8.1415.2 Field Documentation . . . 1323
8.1415.2.1 pLoopbackMode . . . 1323
8.1415.2.2 pLoopbackMultiplier . . . 1323
8.1416 WDSSWICurrentChannelRates Struct Reference . . . 1323
8.1416.1 Detailed Description . . . 1324
8.1416.2 Field Documentation . . . 1324
8.1416.2.1 current_channel_rx_rate . . . 1324
8.1416.2.2 current_channel_tx_rate . . . 1324
8.1416.2.3 max_channel_rx_rate . . . 1324
8.1416.2.4 max_channel_tx_rate . . . 1324

9 File Documentation **1325**

9.1 apdoxypages.c File Reference . . . 1325
9.1.1 Detailed Description . . . 1325
9.2 common.h File Reference . . . 1325
9.2.1 Macro Definition Documentation . . . 1327

9.2.1.1 DEAULT_LOC_TIMEOUT_IN_SEC . . . 1327
9.2.1.2 MINREQBKLEN . . . 1327
9.2.1.3 MSGID_AND_LEN . . . 1327
9.2.1.4 MSGID_DONT_CARE . . . 1327
9.2.1.5 SDK_VALIDATE_INPUT_PACK_PARAM . . . 1327
9.2.1.6 SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID . . . 1327
9.2.1.7 SDU_HDR_LEN . . . 1327
9.2.1.8 UNUSEDPARAM . . . 1327
9.2.2 Typedef Documentation . . . 1327
9.2.2.1 logger . . . 1327
9.2.3 Enumeration Type Documentation . . . 1327
9.2.3.1 eLOG_LEVEL . . . 1327
9.2.3.2 eQMI_SVC . . . 1328
9.2.3.3 eTimeout . . . 1328
9.2.3.4 msgtype . . . 1329
9.2.4 Function Documentation . . . 1329
9.2.4.1 fill_pack_ctx . . . 1329
9.2.4.2 fill_sdu_hdr . . . 1329
9.2.4.3 get_version . . . 1329
9.2.4.4 helper_get_resp_ctx . . . 1329
9.2.4.5 helper_get_xid . . . 1329
9.2.4.6 helper_set_log_func . . . 1329
9.2.4.7 helper_set_log_lvl . . . 1329
9.2.4.8 liteqmi_GetVersion . . . 1329
9.2.4.9 liteqmi_log . . . 1330
9.2.4.10 unpack_result_code_only . . . 1330
9.2.5 Variable Documentation . . . 1330
9.2.5.1 glog . . . 1330
9.2.5.2 gloglvl . . . 1330
9.3 dms.h File Reference . . . 1330
9.3.1 Macro Definition Documentation . . . 1336
9.3.1.1 ACT_CODE_MAX_SIZE . . . 1336
9.3.1.2 CK_MAX_SIZE . . . 1336
9.3.1.3 DMS_IMGDETAILS_LEN . . . 1336
9.3.1.4 DMS_MAX_CUST_ID_LEN . . . 1336
9.3.1.5 DMS_MAX_CUST_VALUE_LEN . . . 1336
9.3.1.6 DMS_MAX_FWUPDATE_LOG_STR_SZ . . . 1336
9.3.1.7 DMS_MAX_FWUPDATE_REF_STR_SZ . . . 1336
9.3.1.8 DMS_PM_FACTORY . . . 1336
9.3.1.9 DMS_PM_LOW . . . 1336

9.3.1.10 DMS_PM_OFFLINE . . . 1336
9.3.1.11 DMS_PM_ONLINE . . . 1336
9.3.1.12 DMS_PM_PERSISTENT_LOW . . . 1336
9.3.1.13 DMS_PM_RESET . . . 1336
9.3.1.14 DMS_PM_SHUT_DOWN . . . 1336
9.3.1.15 DMS_SET_REPORT_DISABLE . . . 1337
9.3.1.16 DMS_SET_REPORT_ENABLE . . . 1337
9.3.1.17 DMS_SLQSFWINFO_APPVERSION_SZ . . . 1337
9.3.1.18 DMS_SLQSFWINFO_BOOTVERSION_SZ . . . 1337
9.3.1.19 DMS_SLQSFWINFO_CARRIER_SZ . . . 1337
9.3.1.20 DMS_SLQSFWINFO_CUR_CARR_NAME . . . 1337
9.3.1.21 DMS_SLQSFWINFO_CUR_CARR_REV . . . 1337
9.3.1.22 DMS_SLQSFWINFO_MODELID_SZ . . . 1337
9.3.1.23 DMS_SLQSFWINFO_PACKAGEID_SZ . . . 1337
9.3.1.24 DMS_SLQSFWINFO_PRIVERSION_SZ . . . 1337
9.3.1.25 DMS_SLQSFWINFO_SKU_SZ . . . 1337
9.3.1.26 DMS_SWI_SET_IND_DISABLE . . . 1337
9.3.1.27 DMS_SWI_SET_IND_ENABLE . . . 1337
9.3.1.28 DMS_UINT8_MAX_STRING_SZ . . . 1337
9.3.1.29 ERI_DATA_MAX_SIZE . . . 1337
9.3.1.30 MAX_BUILD_ID_LEN . . . 1337
9.3.1.31 MEID_MAX_SIZE . . . 1337
9.3.1.32 SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1337
9.3.1.33 SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1337
9.3.1.34 SPC_SIZE . . . 1337
9.3.1.35 UNIQUE_ID_LEN . . . 1337
9.3.2 Function Documentation . . . 1337
9.3.2.1 pack_dms_ActivateAutomatic . . . 1337
9.3.2.2 pack_dms_GetActivationState . . . 1338
9.3.2.3 pack_dms_GetBandCapability . . . 1338
9.3.2.4 pack_dms_GetCrashAction . . . 1338
9.3.2.5 pack_dms_GetCustFeature . . . 1339
9.3.2.6 pack_dms_GetCustFeaturesV2 . . . 1339
9.3.2.7 pack_dms_GetDeviceCap . . . 1339
9.3.2.8 pack_dms_GetDeviceCapabilities . . . 1339
9.3.2.9 pack_dms_GetDeviceHardwareRev . . . 1340
9.3.2.10 pack_dms_GetDeviceMfr . . . 1340
9.3.2.11 pack_dms_GetDeviceSerialNumbers . . . 1341
9.3.2.12 pack_dms_GetFirmwareInfo . . . 1341
9.3.2.13 pack_dms_GetFirmwareRevision . . . 1341

9.3.2.14 pack_dms_GetFirmwareRevisions . . . 1342
9.3.2.15 pack_dms_GetFSN . . . 1342
9.3.2.16 pack_dms_GetHardwareRevision . . . 1342
9.3.2.17 pack_dms_GetIMSI . . . 1343
9.3.2.18 pack_dms_GetManufacturer . . . 1343
9.3.2.19 pack_dms_GetModelID . . . 1343
9.3.2.20 pack_dms_GetNetworkTime . . . 1344
9.3.2.21 pack_dms_GetOfflineReason . . . 1344
9.3.2.22 pack_dms_GetPower . . . 1344
9.3.2.23 pack_dms_GetPRLVersion . . . 1345
9.3.2.24 pack_dms_GetSerialNumbers . . . 1345
9.3.2.25 pack_dms_GetUSBComp . . . 1346
9.3.2.26 pack_dms_GetVoiceNumber . . . 1346
9.3.2.27 pack_dms_ResetToFactoryDefaults . . . 1346
9.3.2.28 pack_dms_SetActivationStatusCallback . . . 1347
9.3.2.29 pack_dms_SetCrashAction . . . 1347
9.3.2.30 pack_dms_SetCustFeature . . . 1347
9.3.2.31 pack_dms_SetCustFeaturesV2 . . . 1348
9.3.2.32 pack_dms_SetEventReport . . . 1348
9.3.2.33 pack_dms_SetFirmwarePreference . . . 1348
9.3.2.34 pack_dms_SetPower . . . 1349
9.3.2.35 pack_dms_SetUSBComp . . . 1349
9.3.2.36 pack_dms_SLQSDmsSwiGetResetInfo . . . 1350
9.3.2.37 pack_dms_SLQSDmsSwiIndicationRegister . . . 1350
9.3.2.38 pack_dms_SLQSGetBandCapability . . . 1350
9.3.2.39 pack_dms_SLQSGetERIFile . . . 1351
9.3.2.40 pack_dms_SLQSSwiClearDyingGaspStatistics . . . 1351
9.3.2.41 pack_dms_SLQSSwiGetCrashInfo . . . 1351
9.3.2.42 pack_dms_SLQSSwiGetDyingGaspCfg . . . 1352
9.3.2.43 pack_dms_SLQSSwiGetDyingGaspStatistics . . . 1352
9.3.2.44 pack_dms_SLQSSwiGetFirmwareCurr . . . 1352
9.3.2.45 pack_dms_SLQSSwiGetFwUpdateStatus . . . 1353
9.3.2.46 pack_dms_SLQSSwiGetHostDevInfo . . . 1353
9.3.2.47 pack_dms_SLQSSwiGetOSInfo . . . 1353
9.3.2.48 pack_dms_SLQSSwiGetSerialNoExt . . . 1354
9.3.2.49 pack_dms_SLQSSwiSetDyingGaspCfg . . . 1354
9.3.2.50 pack_dms_SLQSSwiSetHostDevInfo . . . 1354
9.3.2.51 pack_dms_SLQSSwiSetOSInfo . . . 1355
9.3.2.52 pack_dms_SLQSUIMGetState . . . 1355
9.3.2.53 pack_dms_UIMChangePIN . . . 1355

9.3.2.54 pack_dms_UIMGetControlKeyStatus . . . 1356
9.3.2.55 pack_dms_UIMGetICCID . . . 1356
9.3.2.56 pack_dms_UIMGetPINStatus . . . 1357
9.3.2.57 pack_dms_UIMSetControlKeyProtection . . . 1357
9.3.2.58 pack_dms_UIMSetPINProtection . . . 1357
9.3.2.59 pack_dms_UIMUnblockControlKey . . . 1358
9.3.2.60 pack_dms_UIMUnblockPIN . . . 1358
9.3.2.61 pack_dms_UIMVerifyPIN . . . 1358
9.3.2.62 pack_dms_ValidateSPC . . . 1359
9.3.2.63 unpack_dms_ActivateAutomatic . . . 1359
9.3.2.64 unpack_dms_GetActivationState . . . 1360
9.3.2.65 unpack_dms_GetBandCapability . . . 1360
9.3.2.66 unpack_dms_GetCrashAction . . . 1360
9.3.2.67 unpack_dms_GetCustFeature . . . 1361
9.3.2.68 unpack_dms_GetCustFeaturesV2 . . . 1361
9.3.2.69 unpack_dms_GetDeviceCap . . . 1361
9.3.2.70 unpack_dms_GetDeviceCapabilities . . . 1361
9.3.2.71 unpack_dms_GetDeviceHardwareRev . . . 1362
9.3.2.72 unpack_dms_GetDeviceMfr . . . 1362
9.3.2.73 unpack_dms_GetDeviceSerialNumbers . . . 1362
9.3.2.74 unpack_dms_GetFirmwareInfo . . . 1363
9.3.2.75 unpack_dms_GetFirmwareRevision . . . 1363
9.3.2.76 unpack_dms_GetFirmwareRevisions . . . 1363
9.3.2.77 unpack_dms_GetFSN . . . 1364
9.3.2.78 unpack_dms_GetHardwareRevision . . . 1364
9.3.2.79 unpack_dms_GetIMSI . . . 1364
9.3.2.80 unpack_dms_GetManufacturer . . . 1365
9.3.2.81 unpack_dms_GetModelID . . . 1365
9.3.2.82 unpack_dms_GetNetworkTime . . . 1365
9.3.2.83 unpack_dms_GetOfflineReason . . . 1366
9.3.2.84 unpack_dms_GetPower . . . 1366
9.3.2.85 unpack_dms_GetPRLVersion . . . 1366
9.3.2.86 unpack_dms_GetSerialNumbers . . . 1367
9.3.2.87 unpack_dms_GetUSBComp . . . 1367
9.3.2.88 unpack_dms_GetVoiceNumber . . . 1367
9.3.2.89 unpack_dms_ResetToFactoryDefaults . . . 1368
9.3.2.90 unpack_dms_SetActivationStatusCallback . . . 1368
9.3.2.91 unpack_dms_SetCrashAction . . . 1368
9.3.2.92 unpack_dms_SetCustFeature . . . 1369
9.3.2.93 unpack_dms_SetCustFeaturesV2 . . . 1369

9.3.2.94 unpack_dms_SetEventReport . . . 1369
9.3.2.95 unpack_dms_SetEventReport_ind . . . 1370
9.3.2.96 unpack_dms_SetFirmwarePreference . . . 1370
9.3.2.97 unpack_dms_SetPower . . . 1370
9.3.2.98 unpack_dms_SetUSBComp . . . 1371
9.3.2.99 unpack_dms_SLQSDmsSwiGetResetInfo . . . 1371
9.3.2.100 unpack_dms_SLQSDmsSwiGetResetInfo_Ind . . . 1371
9.3.2.101 unpack_dms_SLQSDmsSwiIndicationRegister . . . 1372
9.3.2.102 unpack_dms_SLQSGetBandCapability . . . 1372
9.3.2.103 unpack_dms_SLQSGetERIFile . . . 1373
9.3.2.104 unpack_dms_SLQSSwiClearDyingGaspStatistics . . . 1373
9.3.2.105 unpack_dms_SLQSSwiGetCrashInfo . . . 1373
9.3.2.106 unpack_dms_SLQSSwiGetDyingGaspCfg . . . 1374
9.3.2.107 unpack_dms_SLQSSwiGetDyingGaspStatistics . . . 1374
9.3.2.108 unpack_dms_SLQSSwiGetFirmwareCurr . . . 1374
9.3.2.109 unpack_dms_SLQSSwiGetFwUpdateStatus . . . 1375
9.3.2.110 unpack_dms_SLQSSwiGetHostDevInfo . . . 1375
9.3.2.111 unpack_dms_SLQSSwiGetOSInfo . . . 1375
9.3.2.112 unpack_dms_SLQSSwiGetSerialNoExt . . . 1376
9.3.2.113 unpack_dms_SLQSSwiSetDyingGaspCfg . . . 1376
9.3.2.114 unpack_dms_SLQSSwiSetHostDevInfo . . . 1376
9.3.2.115 unpack_dms_SLQSSwiSetOSInfo . . . 1377
9.3.2.116 unpack_dms_SLQSUIMGetState . . . 1377
9.3.2.117 unpack_dms_UIMChangePIN . . . 1377
9.3.2.118 unpack_dms_UIMGetControlKeyStatus . . . 1378
9.3.2.119 unpack_dms_UIMGetICCID . . . 1378
9.3.2.120 unpack_dms_UIMGetPINStatus . . . 1378
9.3.2.121 unpack_dms_UIMSetControlKeyProtection . . . 1379
9.3.2.122 unpack_dms_UIMSetPINProtection . . . 1379
9.3.2.123 unpack_dms_UIMUnblockControlKey . . . 1379
9.3.2.124 unpack_dms_UIMUnblockPIN . . . 1380
9.3.2.125 unpack_dms_UIMVerifyPIN . . . 1380
9.3.2.126 unpack_dms_ValidateSPC . . . 1380
9.4 fms.h File Reference . . . 1381
9.4.1 Macro Definition Documentation . . . 1382
9.4.1.1 FMS_FW_PRI_BUILD_MATCH_LEN . . . 1382
9.4.1.2 FMS_GOBI_LISTENTRIES_MAX . . . 1382
9.4.1.3 FMS_GOBI_MBN_BUILD_ID_STR_LEN . . . 1382
9.4.1.4 FMS_GOBI_MBN_IMG_ID_STR_LEN . . . 1382
9.4.1.5 FMS_IMAGE_ID_BUILD_ID_LEN . . . 1382

9.4.1.6 FMS_IMAGE_ID_IMG_ID_LEN . . . 1382
9.4.1.7 FMS_IMAGE_ID_MAX_ENTRIES . . . 1382
9.4.1.8 FMS_IMAGE_ID_PRI_IMGTYPE . . . 1382
9.4.1.9 FMS_MAX_IMAGE_ID_ELEMENT . . . 1382
9.4.1.10 FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE . . . 1382
9.4.2 Function Documentation . . . 1382
9.4.2.1 GetValidFwPriCombinations . . . 1382
9.4.2.2 pack_fms_GetImagesPreference . . . 1382
9.4.2.3 pack_fms_GetStoredImages . . . 1383
9.4.2.4 pack_fms_SetImagesPreference . . . 1383
9.4.2.5 unpack_fms_GetImagesPreference . . . 1383
9.4.2.6 unpack_fms_GetStoredImages . . . 1383
9.4.2.7 unpack_fms_SetImagesPreference . . . 1384
9.5 loc.h File Reference . . . 1384
9.5.1 Macro Definition Documentation . . . 1386
9.5.1.1 LOC_UINT8_MAX_STRING_SZ . . . 1386
9.5.1.2 LOCEVENTMASKBATCHFULLNOTIFICATION . . . 1386
9.5.1.3 LOCEVENTMASKENGINESTATE . . . 1386
9.5.1.4 LOCEVENTMASKFIXSESSIONSTATE . . . 1386
9.5.1.5 LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION . . . 1387
9.5.1.6 LOCEVENTMASKGEOFENCEBREACHNOTIFICATION . . . 1387
9.5.1.7 LOCEVENTMASKGEOFENCEGENALERT . . . 1387
9.5.1.8 LOCEVENTMASKGNSSMEASUREMENTREPORT . . . 1387
9.5.1.9 LOCEVENTMASKGNSSSVINFO . . . 1387
9.5.1.10 LOCEVENTMASKINJECTPOSITIONREQ . . . 1387
9.5.1.11 LOCEVENTMASKINJECTPREDICTEDORBITSREQ . . . 1387
9.5.1.12 LOCEVENTMASKINJECTTIMEREQ . . . 1387
9.5.1.13 LOCEVENTMASKINJECTWIFIAPDATAREQ . . . 1387
9.5.1.14 LOCEVENTMASKINVALIDVALUE . . . 1387
9.5.1.15 LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT . . . 1388
9.5.1.16 LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ . . . 1388
9.5.1.17 LOCEVENTMASKMOTIONDATACONTROL . . . 1388
9.5.1.18 LOCEVENTMASKNIGEOFENCENOTIFICATION . . . 1388
9.5.1.19 LOCEVENTMASKNINOTIFYVERIFYREQ . . . 1388
9.5.1.20 LOCEVENTMASKNMEA . . . 1388
9.5.1.21 LOCEVENTMASKPEDOMETERCONTROL . . . 1388
9.5.1.22 LOCEVENTMASKPOSITIONREPORT . . . 1388
9.5.1.23 LOCEVENTMASKSENSORSTREAMINGREADYSTATUS . . . 1388
9.5.1.24 LOCEVENTMASKSETSPISTREAMINGREPORT . . . 1388
9.5.1.25 LOCEVENTMASKTIMESYNCREQ . . . 1389

9.5.1.26 LOCEVENTMASKVEHICLEDATAREADYSTATUS . . . 1389
9.5.1.27 LOCEVENTMASKWIFIREQ . . . 1389
9.5.1.28 MAX_SENSOR_DATA_LEN . . . 1389
9.5.1.29 MAX_TEMP_DATA_LEN . . . 1389
9.5.2 Enumeration Type Documentation . . . 1389
9.5.2.1 anonymous enum . . . 1389
9.5.3 Function Documentation . . . 1389
9.5.3.1 pack_loc_DeleteAssistData . . . 1389
9.5.3.2 pack_loc_EventRegister . . . 1389
9.5.3.3 pack_loc_SetExtPowerState . . . 1390
9.5.3.4 pack_loc_SetOperationMode . . . 1390
9.5.3.5 pack_loc_SLQSLOCGetBestAvailPos . . . 1391
9.5.3.6 pack_loc_SLQSLOCInjectPosition . . . 1391
9.5.3.7 pack_loc_SLQSLOCInjectSensorData . . . 1391
9.5.3.8 pack_loc_SLQSLOCInjectUTCTime . . . 1392
9.5.3.9 pack_loc_SLQSLOCSetCradleMountConfig . . . 1392
9.5.3.10 pack_loc_Start . . . 1392
9.5.3.11 pack_loc_Stop . . . 1393
9.5.3.12 unpack_loc_BestAvailPos_Ind . . . 1393
9.5.3.13 unpack_loc_DeleteAssistData . . . 1393
9.5.3.14 unpack_loc_DeleteAssistData_Ind . . . 1394
9.5.3.15 unpack_loc_EngineState_Ind . . . 1394
9.5.3.16 unpack_loc_EventRegister . . . 1394
9.5.3.17 unpack_loc_GnssSvInfo_Ind . . . 1395
9.5.3.18 unpack_loc_PositionRpt_Ind . . . 1395
9.5.3.19 unpack_loc_SetExtPowerConfig_Ind . . . 1395
9.5.3.20 unpack_loc_SetExtPowerState . . . 1396
9.5.3.21 unpack_loc_SetOperationMode . . . 1396
9.5.3.22 unpack_loc_SetOperationMode_Ind . . . 1396
9.5.3.23 unpack_loc_SLQSLOCGetBestAvailPos . . . 1397
9.5.3.24 unpack_loc_SLQSLOCInjectPosition . . . 1397
9.5.3.25 unpack_loc_SLQSLOCInjectSensorData . . . 1397
9.5.3.26 unpack_loc_SLQSLOCInjectUTCTime . . . 1398
9.5.3.27 unpack_loc_SLQSLOCSetCradleMountConfig . . . 1398
9.5.3.28 unpack_loc_Start . . . 1398
9.5.3.29 unpack_loc_Stop . . . 1399
9.6 nas.h File Reference . . . 1399
9.6.1 Macro Definition Documentation . . . 1407
9.6.1.1 NAS_IMSI_M_S1_LENGTH . . . 1407
9.6.1.2 NAS_IMSI_M_S2_LENGTH . . . 1407

9.6.1.3 NAS_MAX_DESCRIPTION_LENGTH . . . 1407
9.6.1.4 NAS_MAX_NUM_NETWORKS . . . 1407
9.6.1.5 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1407
9.6.1.6 NAS_NAM_NAME_LENGTH . . . 1407
9.6.1.7 NAS_OTA_MESSAGE_MAX_BUF_SIZE . . . 1407
9.6.1.8 NAS_PLMN_LENGTH . . . 1407
9.6.1.9 NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST . . . 1407
9.6.2 Enumeration Type Documentation . . . 1407
9.6.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB . . . 1407
9.6.2.2 LITEQMI_NAS_LTE_CPHY_SCELL_STATE . . . 1407
9.6.2.3 NAS_LTE_CPHY_CA_BW_NRB_LITE . . . 1407
9.6.2.4 NAS_LTE_CPHY_SCELL_STATE_LITE . . . 1408
9.6.3 Function Documentation . . . 1408
9.6.3.1 pack_nas_GetACCOLC . . . 1408
9.6.3.2 pack_nas_GetANAAAAuthenticationStatus . . . 1408
9.6.3.3 pack_nas_GetCDMANetworkParameters . . . 1408
9.6.3.4 pack_nas_GetHomeNetwork . . . 1409
9.6.3.5 pack_nas_GetHomeNetwork3GPP2 . . . 1409
9.6.3.6 pack_nas_GetNetworkPreference . . . 1409
9.6.3.7 pack_nas_GetRFInfo . . . 1409
9.6.3.8 pack_nas_GetServingNetwork . . . 1410
9.6.3.9 pack_nas_GetServingNetworkCapabilities . . . 1410
9.6.3.10 pack_nas_GetSignalStrengths . . . 1410
9.6.3.11 pack_nas_InitiateDomainAttach . . . 1411
9.6.3.12 pack_nas_PerformNetworkScan . . . 1411
9.6.3.13 pack_nas_SetACCOLC . . . 1411
9.6.3.14 pack_nas_SetCDMANetworkParameters . . . 1412
9.6.3.15 pack_nas_SetLURejectCallback . . . 1412
9.6.3.16 pack_nas_SetNetworkPreference . . . 1412
9.6.3.17 pack_nas_SetRFInfoCallback . . . 1413
9.6.3.18 pack_nas_SLQSConfigSigInfo . . . 1413
9.6.3.19 pack_nas_SLQSGetErrorRate . . . 1413
9.6.3.20 pack_nas_SlqsGetLTECphyCAInfo . . . 1414
9.6.3.21 pack_nas_SLQSGetNetworkTime . . . 1414
9.6.3.22 pack_nas_SLQSGetOperatorNameData . . . 1414
9.6.3.23 pack_nas_SLQSGetPLMNName . . . 1414
9.6.3.24 pack_nas_SLQSGetServingSystem . . . 1415
9.6.3.25 pack_nas_SLQSGetSignalStrength . . . 1415
9.6.3.26 pack_nas_SLQSGetSysInfo . . . 1415
9.6.3.27 pack_nas_SLQSGetSysSelectionPref . . . 1416

9.6.3.28 pack_nas_SLQSInitiateNetworkRegistration . . . 1416
9.6.3.29 pack_nas_SLQSNasConfigSigInfo2 . . . 1416
9.6.3.30 pack_nas_SLQSNasGet3GPP2Subscription . . . 1417
9.6.3.31 pack_nas_SLQSNasGetCellLocationInfo . . . 1417
9.6.3.32 pack_nas_SLQSNasGetHDRColorCode . . . 1417
9.6.3.33 pack_nas_SLQSNasGetSigInfo . . . 1418
9.6.3.34 pack_nas_SLQSNasGetTxRxInfo . . . 1418
9.6.3.35 pack_nas_SLQSNasIndicationRegisterExt . . . 1418
9.6.3.36 pack_nas_SLQSNASSwiGetChannelLock . . . 1419
9.6.3.37 pack_nas_SLQSNasSwiIndicationRegister . . . 1419
9.6.3.38 pack_nas_SLQSNasSwiModemStatus . . . 1419
9.6.3.39 pack_nas_SLQSNASSwiSetChannelLock . . . 1420
9.6.3.40 pack_nas_SLQSSetBandPreference . . . 1420
9.6.3.41 pack_nas_SLQSSetSignalStrengthsCallback . . . 1420
9.6.3.42 pack_nas_SLQSSetSysSelectionPref . . . 1421
9.6.3.43 pack_nas_SLQSSwiGetHDRPersonality . . . 1421
9.6.3.44 pack_nas_SLQSSwiGetHDRProtSubtype . . . 1421
9.6.3.45 pack_nas_SLQSSwiGetHRPDStats . . . 1422
9.6.3.46 pack_nas_SLQSSwiGetLteCQI . . . 1422
9.6.3.47 pack_nas_SLQSSwiGetLteSccRxInfo . . . 1422
9.6.3.48 pack_nas_SLQSSwiNetworkDebug . . . 1422
9.6.3.49 pack_nas_SLQSSwiPSDetach . . . 1423
9.6.3.50 unpack_nas_GetACCOLC . . . 1423
9.6.3.51 unpack_nas_GetANAAAAuthenticationStatus . . . 1423
9.6.3.52 unpack_nas_GetCDMANetworkParameters . . . 1424
9.6.3.53 unpack_nas_GetHomeNetwork . . . 1424
9.6.3.54 unpack_nas_GetHomeNetwork3GPP2 . . . 1424
9.6.3.55 unpack_nas_GetNetworkPreference . . . 1425
9.6.3.56 unpack_nas_GetRFInfo . . . 1425
9.6.3.57 unpack_nas_GetServingNetwork . . . 1425
9.6.3.58 unpack_nas_GetServingNetworkCapabilities . . . 1425
9.6.3.59 unpack_nas_GetSignalStrengths . . . 1426
9.6.3.60 unpack_nas_InitiateDomainAttach . . . 1426
9.6.3.61 unpack_nas_PerformNetworkScan . . . 1426
9.6.3.62 unpack_nas_SetACCOLC . . . 1427
9.6.3.63 unpack_nas_SetCDMANetworkParameters . . . 1427
9.6.3.64 unpack_nas_SetDataCapabilitiesCallback_ind . . . 1427
9.6.3.65 unpack_nas_SetEventReportInd . . . 1428
9.6.3.66 unpack_nas_SetLURejectCallback . . . 1428
9.6.3.67 unpack_nas_SetNasLTECphyCaIndCallback_ind . . . 1428

9.6.3.68 unpack_nas_SetNetworkPreference . . . 1428
9.6.3.69 unpack_nas_SetRFInfoCallback . . . 1428
9.6.3.70 unpack_nas_SetRoamingIndicatorCallback_ind . . . 1429
9.6.3.71 unpack_nas_SetServingSystemCallback_ind . . . 1429
9.6.3.72 unpack_nas_SLQSConfigSigInfo . . . 1429
9.6.3.73 unpack_nas_SLQSGetErrorRate . . . 1429
9.6.3.74 unpack_nas_SlqsGetLTECphyCAInfo . . . 1430
9.6.3.75 unpack_nas_SLQSGetNetworkTime . . . 1430
9.6.3.76 unpack_nas_SLQSGetOperatorNameData . . . 1430
9.6.3.77 unpack_nas_SLQSGetPLMNName . . . 1430
9.6.3.78 unpack_nas_SLQSGetServingSystem . . . 1431
9.6.3.79 unpack_nas_SLQSGetSignalStrength . . . 1431
9.6.3.80 unpack_nas_SLQSGetSysInfo . . . 1431
9.6.3.81 unpack_nas_SLQSGetSysSelectionPref . . . 1432
9.6.3.82 unpack_nas_SLQSInitiateNetworkRegistration . . . 1432
9.6.3.83 unpack_nas_SLQSNasConfigSigInfo2 . . . 1432
9.6.3.84 unpack_nas_SLQSNasGet3GPP2Subscription . . . 1433
9.6.3.85 unpack_nas_SLQSNasGetCellLocationInfo . . . 1433
9.6.3.86 unpack_nas_SLQSNasGetHDRColorCode . . . 1433
9.6.3.87 unpack_nas_SLQSNasGetSigInfo . . . 1434
9.6.3.88 unpack_nas_SLQSNasGetTxRxInfo . . . 1434
9.6.3.89 unpack_nas_SLQSNasIndicationRegisterExt . . . 1434
9.6.3.90 unpack_nas_SLQSNasNetworkTimeCallBack_ind . . . 1435
9.6.3.91 unpack_nas_SLQSNasSigInfoCallback_ind . . . 1435
9.6.3.92 unpack_nas_SLQSNASSwiGetChannelLock . . . 1435
9.6.3.93 unpack_nas_SLQSNasSwiIndicationRegister . . . 1435
9.6.3.94 unpack_nas_SLQSNasSwiModemStatus . . . 1436
9.6.3.95 unpack_nas_SLQSNasSwiOTAMessageCallback_ind . . . 1436
9.6.3.96 unpack_nas_SLQSNASSwiSetChannelLock . . . 1436
9.6.3.97 unpack_nas_SLQSNasSysInfoCallback_ind . . . 1437
9.6.3.98 unpack_nas_SLQSNasTimerCallback_ind . . . 1437
9.6.3.99 unpack_nas_SLQSSetBandPreference . . . 1437
9.6.3.100 unpack_nas_SLQSSetSignalStrengthsCallback . . . 1438
9.6.3.101 unpack_nas_SLQSSetSysSelectionPref . . . 1438
9.6.3.102 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind . . . 1438
9.6.3.103 unpack_nas_SLQSSwiGetHDRPersonality . . . 1438
9.6.3.104 unpack_nas_SLQSSwiGetHDRProtSubtype . . . 1439
9.6.3.105 unpack_nas_SLQSSwiGetHRPDStats . . . 1439
9.6.3.106 unpack_nas_SLQSSwiGetLteCQI . . . 1439
9.6.3.107 unpack_nas_SLQSSwiGetLteSccRxInfo . . . 1440

9.6.3.108 unpack_nas_SLQSSwiNetworkDebug . . . 1440
9.6.3.109 unpack_nas_SLQSSwiPSDetach . . . 1440
9.7 pds.h File Reference . . . 1441
9.7.1 Function Documentation . . . 1441
9.7.1.1 pack_pds_GetPDSDefaults . . . 1441
9.7.1.2 pack_pds_GetPDSState . . . 1442
9.7.1.3 pack_pds_GetPortAutomaticTracking . . . 1442
9.7.1.4 pack_pds_SetPDSDefaults . . . 1442
9.7.1.5 pack_pds_SetPDSState . . . 1443
9.7.1.6 pack_pds_SetPortAutomaticTracking . . . 1443
9.7.1.7 unpack_pds_GetPDSDefaults . . . 1443
9.7.1.8 unpack_pds_GetPDSState . . . 1444
9.7.1.9 unpack_pds_GetPortAutomaticTracking . . . 1444
9.7.1.10 unpack_pds_SetPDSDefaults . . . 1444
9.7.1.11 unpack_pds_SetPDSState . . . 1445
9.7.1.12 unpack_pds_SetPortAutomaticTracking . . . 1445
9.8 qaCbkCatEventReportInd.h File Reference . . . 1445
9.8.1 Macro Definition Documentation . . . 1446
9.8.1.1 QMI_CAN_COMMON_EVENT_TLV_NUMBER . . . 1446
9.8.1.2 QMI_MAX_CAT_EVENT_DATA_LENGTH . . . 1446
9.8.2 Enumeration Type Documentation . . . 1446
9.8.2.1 eQMI_CAT_EVENT_REPORT_IND_TLV . . . 1446
9.8.2.2 eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH . . . 1447
9.8.3 Function Documentation . . . 1447
9.8.3.1 UpkQmiCbkCatEventReportInd . . . 1447
9.9 qaCbkSwiOmaDmEventReportInd.h File Reference . . . 1447
9.9.1 Macro Definition Documentation . . . 1448
9.9.1.1 QMI_SWIOMA_DM_CONFIG . . . 1448
9.9.1.2 QMI_SWIOMA_DM_FOTA . . . 1448
9.9.1.3 QMI_SWIOMA_DM_NOT . . . 1448
9.9.2 Enumeration Type Documentation . . . 1448
9.9.2.1 eQMI_SWIOMA_DM_EVENT_REPORT_IND . . . 1448
9.9.3 Function Documentation . . . 1448
9.9.3.1 UpkQmiCbkSwiOmaDmEventReportInd . . . 1448
9.9.3.2 UpkQmiCbkSwiOmaDmEventReportIndExt . . . 1448
9.10 qaGobiApiAudio.h File Reference . . . 1448
9.10.1 Detailed Description . . . 1449
9.10.2 Function Documentation . . . 1449
9.10.2.1 SLQSGetAudioPathConfig . . . 1449
9.10.2.2 SLQSGetAudioProfile . . . 1449

9.10.2.3 SLQSGetAudioVolTLBConfig . . . 1450
9.10.2.4 SLQSSetAudioPathConfig . . . 1450
9.10.2.5 SLQSSetAudioProfile . . . 1451
9.10.2.6 SLQSSetAudioVolTLBConfig . . . 1451
9.11 qaGobiApiCat.h File Reference . . . 1452
9.11.1 Detailed Description . . . 1452
9.11.2 Function Documentation . . . 1452
9.11.2.1 CATSendEnvelopeCommand . . . 1452
9.11.2.2 CATSendTerminalResponse . . . 1453
9.12 qaGobiApiCbk.h File Reference . . . 1453
9.12.1 Detailed Description . . . 1461
9.12.2 Macro Definition Documentation . . . 1461
9.12.2.1 CBK_DISABLE_EVENT . . . 1461
9.12.2.2 CBK_ENABLE_EVENT . . . 1461
9.12.2.3 CBK_NOCHANGE . . . 1461
9.12.2.4 DEREGISTER_EVENT . . . 1461
9.12.2.5 DEREGISTER_SRV . . . 1461
9.12.2.6 DHCP_MAX_NUM_OPTIONS . . . 1461
9.12.2.7 DHCP_OPTION_DATA_BUF_SIZE . . . 1461
9.12.2.8 EVENT_MASK_CARD . . . 1461
9.12.2.9 EVENT_MASK_DEREGISTER_ALL . . . 1461
9.12.2.10 EVENT_MASK_PHY_SLOT_STATUS . . . 1461
9.12.2.11 FIRST_INSTANCE . . . 1461
9.12.2.12 INVALID_INSTACNE . . . 1462
9.12.2.13 IPV4 . . . 1462
9.12.2.14 IPV4V6 . . . 1462
9.12.2.15 IPV6 . . . 1462
9.12.2.16 LOC_EVENT_MASK_ENG_STATE . . . 1462
9.12.2.17 LOC_EVENT_MASK_GNSS_SV_INFO . . . 1462
9.12.2.18 LOC_EVENT_MASK_INJECT_TIME . . . 1462
9.12.2.19 LOC_EVENT_MASK_SENSOR_STREAM . . . 1462
9.12.2.20 LOC_EVENT_MASK_TIME_SYNC . . . 1462
9.12.2.21 LOC_EVENT_POSITION_REPORT . . . 1462
9.12.2.22 MAX_MITIGATION_DEV_ID_LEN . . . 1462
9.12.2.23 MAX_NO_OF_APPLICATIONS . . . 1462
9.12.2.24 MAX_NO_OF_CALLS . . . 1462
9.12.2.25 MAX_NO_OF_FILES . . . 1462
9.12.2.26 MAX_NO_OF_SLOTS . . . 1462
9.12.2.27 MAX_NO_OF_UUSINFO . . . 1462
9.12.2.28 MAX_PATH_LENGTH . . . 1462

9.12.2.29 MAX_RADIO_INTERFACE_LIST . . . 1462
9.12.2.30 MAXUSSDLENGTH . . . 1462
9.12.2.31 NAS_SRV . . . 1462
9.12.2.32 NUM_OF_SET . . . 1462
9.12.2.33 PDS_SRV . . . 1462
9.12.2.34 QMI_ETWS_MAX_PAYLOAD_LENGTH . . . 1462
9.12.2.35 QMI_MAX_VOICE_NUMBER_LENGTH . . . 1462
9.12.2.36 QMI_WMS_MAX_PAYLOAD_LENGTH . . . 1462
9.12.2.37 REGISTER_EVENT . . . 1462
9.12.2.38 REGISTER_SRV . . . 1462
9.12.2.39 SECOND_INSTANCE . . . 1462
9.12.2.40 SIGSTRENGTH_THRESHOLD_ARR_SZ . . . 1463
9.12.2.41 THIRD_INSTANCE . . . 1463
9.12.2.42 USSD_DCS_8BIT . . . 1463
9.12.2.43 USSD_DCS_ASCII . . . 1463
9.12.2.44 USSD_DCS_UCS2 . . . 1463
9.12.2.45 VOICE_SRV . . . 1463
9.12.2.46 WDS_SRV . . . 1463
9.12.3 Typedef Documentation . . . 1463
9.12.3.1 accelAcceptReady . . . 1463
9.12.3.2 accelTempAcceptReady . . . 1463
9.12.3.3 eDevState . . . 1464
9.12.3.4 eSMSEventType . . . 1464
9.12.3.5 gpsTime . . . 1464
9.12.3.6 gyroAcceptReady . . . 1464
9.12.3.7 gyroTempAcceptReady . . . 1465
9.12.3.8 LteNasReleaseInfo . . . 1465
9.12.3.9 modemTempNotification . . . 1466
9.12.3.10 packetSrvStatus . . . 1466
9.12.3.11 precisionDilution . . . 1467
9.12.3.12 ResetInfoNotification . . . 1468
9.12.3.13 sensorDataUsage . . . 1468
9.12.3.14 sessionInformation . . . 1469
9.12.3.15 sessionInformationExt . . . 1469
9.12.3.16 SMSAsyncRawSend . . . 1469
9.12.3.17 SMSCAddressInfo . . . 1470
9.12.3.18 SMSEtwsMessageInfo . . . 1470
9.12.3.19 SMSEtwsPlmnInfo . . . 1471
9.12.3.20 SMSEventInfo . . . 1471
9.12.3.21 SMSMessageModeInfo . . . 1472

9.12.3.22 SMSMTMessageInfo . . . 1472
9.12.3.23 SMSOnIMSInfo . . . 1472
9.12.3.24 SMSTransferRouteMTMessageInfo . . . 1472
9.12.3.25 svUsedforFix . . . 1473
9.12.3.26 SwiAvmsEventInd . . . 1473
9.12.3.27 SwiOTAMsg . . . 1473
9.12.3.28 tFNActivationStatus . . . 1474
9.12.3.29 tFNAllCallStatus . . . 1474
9.12.3.30 tFNASwiLTECphyCaInfo . . . 1474
9.12.3.31 tFNASwiOTAMsg . . . 1475
9.12.3.32 tFNAsyncRawSend . . . 1475
9.12.3.33 tFNBandPreference . . . 1475
9.12.3.34 tFNBestAvailPos . . . 1477
9.12.3.35 tFNCATEvent . . . 1477
9.12.3.36 tFNCbkUimSlotStatusChangeInd . . . 1477
9.12.3.37 tFNDataCapabilities . . . 1478
9.12.3.38 tFNDataSysStatus . . . 1478
9.12.3.39 tFNDelAssistData . . . 1478
9.12.3.40 tFNDeviceStateChange . . . 1478
9.12.3.41 tFNDHCPv4ClientLeaseStatus . . . 1479
9.12.3.42 tFNDTMFEvent . . . 1479
9.12.3.43 tFNDUNCallInfo . . . 1479
9.12.3.44 tFNEventPosition . . . 1479
9.12.3.45 tFNFwDldCompletion . . . 1480
9.12.3.46 tFNGnssSvInfo . . . 1480
9.12.3.47 tFNHDRPersonaity . . . 1480
9.12.3.48 tFNImsaPdpStatus . . . 1480
9.12.3.49 tFNImsaRatStatus . . . 1481
9.12.3.50 tFNImsaRegStatus . . . 1481
9.12.3.51 tFNImsaSvcStatus . . . 1481
9.12.3.52 tFNImsRegMgrConfig . . . 1481
9.12.3.53 tFNImsSIPConfig . . . 1481
9.12.3.54 tFNImsSMSConfig . . . 1482
9.12.3.55 tFNImsUserConfig . . . 1482
9.12.3.56 tFNImsVoIPConfig . . . 1482
9.12.3.57 tFNInfoRec . . . 1482
9.12.3.58 tFNInjectPosition . . . 1483
9.12.3.59 tFNInjectSensorData . . . 1483
9.12.3.60 tFNInjectTimeStatus . . . 1483
9.12.3.61 tFNInjectUTCTime . . . 1483

9.12.3.62 tFNLUReject . . . 1483
9.12.3.63 tFNMemoryFull . . . 1485
9.12.3.64 tFNMessageWaiting . . . 1485
9.12.3.65 tFNMitiLvlRpt . . . 1485
9.12.3.66 tFNMobileIPStatus . . . 1485
9.12.3.67 tFNModemTempInfo . . . 1485
9.12.3.68 tFNNasTimer . . . 1486
9.12.3.69 tFNNet . . . 1486
9.12.3.70 tFNNetworkReject . . . 1486
9.12.3.71 tFNNetworkTime . . . 1486
9.12.3.72 tFNNewGPS . . . 1487
9.12.3.73 tFNNewNMEA . . . 1487
9.12.3.74 tFNNewRMTransferStatistics . . . 1488
9.12.3.75 tFNNewSMS . . . 1488
9.12.3.76 tFNOMADMState . . . 1488
9.12.3.77 tFNOpMode . . . 1489
9.12.3.78 tFNOTASPStatus . . . 1490
9.12.3.79 tFNPacketSrvState . . . 1490
9.12.3.80 tFNPDSState . . . 1490
9.12.3.81 tFNPower . . . 1491
9.12.3.82 tFNPrivacyChange . . . 1491
9.12.3.83 tFNQosNWStatus . . . 1491
9.12.3.84 tFNQosPriEvent . . . 1492
9.12.3.85 tFNQosStatus . . . 1492
9.12.3.86 tFNRankIndicator . . . 1493
9.12.3.87 tFNResetInfo . . . 1493
9.12.3.88 tFNRFInfo . . . 1493
9.12.3.89 tFNRoamingIndicator . . . 1494
9.12.3.90 tFNSDKTerminated . . . 1494
9.12.3.91 tFNSensorStreaming . . . 1494
9.12.3.92 tFNServingSystem . . . 1495
9.12.3.93 tFNSetCradleMount . . . 1495
9.12.3.94 tFNSetEngineState . . . 1495
9.12.3.95 tFNSetEventTimeSync . . . 1495
9.12.3.96 tFNSetExtPowerConfig . . . 1495
9.12.3.97 tFNSigInfo . . . 1496
9.12.3.98 tFNSignalStrength . . . 1496
9.12.3.99 tFNSLQSOMADMAlert . . . 1496
9.12.3.100 tFNSLQSQOSEvent . . . 1496
9.12.3.101 tFNSLQSSessionState . . . 1497

9.12.3.102tFNSLQSSignalStrengths . . . 1497
9.12.3.103tFNSLQSWDSEvent . . . 1497
9.12.3.104tFNSMSEvents . . . 1497
9.12.3.105tFNSUPSInfo . . . 1497
9.12.3.106tFNSUPSNotification . . . 1498
9.12.3.107tFNSwiAvmEventReport . . . 1498
9.12.3.108tFNSysInfo . . . 1498
9.12.3.109tFNSysSelectionPref . . . 1498
9.12.3.110tFNtransLayerInfo . . . 1498
9.12.3.111tFNtransNWRegInfo . . . 1499
9.12.3.112tFNUIMRefresh . . . 1499
9.12.3.113tFNUIMStatusChangeInfo . . . 1499
9.12.3.114tFNUSSDNotification . . . 1499
9.12.3.115tFNUSSDNoWaitIndication . . . 1500
9.12.3.116tFNUSSDRelease . . . 1500
9.12.3.117transLayerNotification . . . 1500
9.12.3.118transNWRegInfoNotification . . . 1500
9.12.4 Enumeration Type Documentation . . . 1501
9.12.4.1 device_state_enum . . . 1501
9.12.4.2 eQaQMIService . . . 1501
9.12.4.3 SMSEventType . . . 1501
9.12.4.4 TYPE_OF_SERVICE_DOMAIN . . . 1501
9.12.5 Function Documentation . . . 1502
9.12.5.1 iSetCATEventCallback . . . 1502
9.12.5.2 iSetSignalStrengthCallback . . . 1502
9.12.5.3 iSLQSSetDUNCallInfoCallback . . . 1502
9.12.5.4 iSLQSSetSignalStrengthsCallback . . . 1502
9.12.5.5 iSLQSSetWdsFirstInstEventCallback . . . 1502
9.12.5.6 iSLQSSetWdsSecondInstEventCallback . . . 1502
9.12.5.7 iSLQSSetWdsThirdInstEventCallback . . . 1502
9.12.5.8 iSLQSSetWdsXferStatsFirstInstCallback . . . 1502
9.12.5.9 iSLQSSetWdsXferStatsSecondInstCallback . . . 1502
9.12.5.10 SetActivationStatusCallback . . . 1502
9.12.5.11 SetCATEventCallback . . . 1502
9.12.5.12 SetDataCapabilitiesCallback . . . 1503
9.12.5.13 SetDeviceStateChangeCbk . . . 1504
9.12.5.14 SetFwDldCompletionCbk . . . 1504
9.12.5.15 SetGPSCallback . . . 1505
9.12.5.16 SetLocBestAvailPosCallback . . . 1505
9.12.5.17 SetLocCradleMountCallback . . . 1505

9.12.5.18 SetLocDeleteAssistDataCallback . . . 1506
9.12.5.19 SetLocEngineStateCallback . . . 1506
9.12.5.20 SetLocEventPositionCallback . . . 1506
9.12.5.21 SetLocEventTimeSyncCallback . . . 1506
9.12.5.22 SetLocGnssSvInfoCallback . . . 1507
9.12.5.23 SetLocInjectSensorDataCallback . . . 1507
9.12.5.24 SetLocInjectTimeCallback . . . 1507
9.12.5.25 SetLocOpModeCallback . . . 1508
9.12.5.26 SetLocSensorStreamingCallback . . . 1508
9.12.5.27 SetLocSetExtPowerConfigCallback . . . 1508
9.12.5.28 SetLURejectCallback . . . 1508
9.12.5.29 SetMobileIPStatusCallback . . . 1509
9.12.5.30 SetNasLTECphyCaIndCallback . . . 1509
9.12.5.31 SetNetChangeCbk . . . 1510
9.12.5.32 SetNewSMSCallback . . . 1510
9.12.5.33 SetNMEACallback . . . 1511
9.12.5.34 SetOMADMStateCallback . . . 1511
9.12.5.35 SetPDSStateCallback . . . 1511
9.12.5.36 SetPowerCallback . . . 1512
9.12.5.37 SetRankIndicatorCallback . . . 1512
9.12.5.38 SetRFInfoCallback . . . 1512
9.12.5.39 SetRMTransferStatisticsCallback . . . 1513
9.12.5.40 SetRoamingIndicatorCallback . . . 1513
9.12.5.41 SetSignalStrengthCallback . . . 1513
9.12.5.42 SetSLQSOMADMAlertCallback . . . 1514
9.12.5.43 SetSLQSOMADMAlertCallbackExt . . . 1514
9.12.5.44 SetUimSlotStatusChangeCallback . . . 1515
9.12.5.45 SetUSSDNotificationCallback . . . 1515
9.12.5.46 SetUSSDNoWaitIndicationCallback . . . 1515
9.12.5.47 SetUSSDReleaseCallback . . . 1516
9.12.5.48 SLQSNasNetworkRejectCallback . . . 1516
9.12.5.49 SLQSNasNetworkTimeCallBack . . . 1516
9.12.5.50 SLQSNasSigInfo2CallBack . . . 1517
9.12.5.51 SLQSNasSigInfoCallBack . . . 1517
9.12.5.52 SLQSNasSwiOTAMessageCallback . . . 1518
9.12.5.53 SLQSNasSysInfoCallBack . . . 1518
9.12.5.54 SLQSNasTimerCallback . . . 1519
9.12.5.55 SLQSSetBandPreferenceCbk . . . 1519
9.12.5.56 SLQSSetDataSystemStatusCallback . . . 1519
9.12.5.57 SLQSSetDHCPv4ClientLeaseStatusCallback . . . 1520

9.12.5.58 SLQSSetDUNCallInfoCallback . . . 1520
9.12.5.59 SLQSSetIMSAPdpStatusCallback . . . 1521
9.12.5.60 SLQSSetIMSARatStatusCallback . . . 1521
9.12.5.61 SLQSSetIMSARegStatusCallback . . . 1521
9.12.5.62 SLQSSetIMSASvcStatusCallback . . . 1522
9.12.5.63 SLQSSetIMSSMSConfigCallback . . . 1522
9.12.5.64 SLQSSetIMSUserConfigCallback . . . 1523
9.12.5.65 SLQSSetIMSVoIPConfigCallback . . . 1523
9.12.5.66 SLQSSetLocInjectPositionCallback . . . 1524
9.12.5.67 SLQSSetLocInjectUTCTimeCallback . . . 1524
9.12.5.68 SLQSSetModemTempCallback . . . 1524
9.12.5.69 SLQSSetPacketSrvStatusCallback . . . 1524
9.12.5.70 SLQSSetQosEventCallback . . . 1525
9.12.5.71 SLQSSetQosNWStatusCallback . . . 1525
9.12.5.72 SLQSSetQosPriEventCallback . . . 1526
9.12.5.73 SLQSSetQosStatusCallback . . . 1526
9.12.5.74 SLQSSetRegMgrConfigCallback . . . 1526
9.12.5.75 SLQSSetSDKTerminatedCallback . . . 1527
9.12.5.76 SLQSSetServingSystemCallback . . . 1527
9.12.5.77 SLQSSetSessionStateCallback . . . 1528
9.12.5.78 SLQSSetSignalStrengthsCallback . . . 1528
9.12.5.79 SLQSSetSIPConfigCallback . . . 1529
9.12.5.80 SLQSSetSMSEventCallback . . . 1529
9.12.5.81 SLQSSetSwiGetResetInfoCallback . . . 1529
9.12.5.82 SLQSSetSwiHDRPersCallback . . . 1530
9.12.5.83 SLQSSetSysSelectionPrefCallBack . . . 1530
9.12.5.84 SLQSSetTransLayerInfoCallback . . . 1530
9.12.5.85 SLQSSetTransNWRegInfoCallback . . . 1531
9.12.5.86 SLQSSetWdsEventCallback . . . 1531
9.12.5.87 SLQSSetWdsTransferStatisticCallback . . . 1532
9.12.5.88 SLQSSwiAVMSEventReportCallback . . . 1533
9.12.5.89 SLQSTmdMitigationLvlRptCallback . . . 1533
9.12.5.90 SLQSUIMSetRefreshCallBack . . . 1533
9.12.5.91 SLQSUIMSetStatusChangeCallBack . . . 1533
9.12.5.92 SLQSVoiceInfoRecCallback . . . 1534
9.12.5.93 SLQSVoiceSetAllCallStatusCallBack . . . 1534
9.12.5.94 SLQSVoiceSetDTMFEventCallBack . . . 1535
9.12.5.95 SLQSVoiceSetOTASPStatusCallBack . . . 1535
9.12.5.96 SLQSVoiceSetPrivacyChangeCallBack . . . 1536
9.12.5.97 SLQSVoiceSetSUPSCallBack . . . 1536

9.12.5.98 SLQSVoiceSetSUPSNotificationCallback . . . 1536
9.12.5.99 SLQSWmsAsyncRawSendCallBack . . . 1537
9.12.5.100 SLQSWmsMemoryFullCallBack . . . 1537
9.12.5.101 SLQSWmsMessageWaitingCallBack . . . 1538
9.13 qaGobiApiDcs.h File Reference . . . 1538
9.13.1 Detailed Description . . . 1539
9.13.2 Macro Definition Documentation . . . 1539
9.13.2.1 LEN . . . 1539
9.13.2.2 PORTNAM_LEN . . . 1539
9.13.3 Function Documentation . . . 1539
9.13.3.1 QCWWAN2kConnect . . . 1539
9.13.3.2 QCWWAN2kEnumerateDevices . . . 1540
9.13.3.3 QCWWAN2kGetConnectedDeviceID . . . 1540
9.13.3.4 QCWWANConnect . . . 1541
9.13.3.5 QCWWANDisconnect . . . 1541
9.13.3.6 QCWWANEnumerateDevices . . . 1541
9.13.3.7 SetSDKImagePath . . . 1542
9.13.3.8 SLQSGetDeviceMode . . . 1542
9.13.3.9 SLQSGetNetStatistic . . . 1543
9.13.3.10 SLQSGetUsbPortNames . . . 1543
9.13.3.11 SLQSKillSDKProcess . . . 1543
9.13.3.12 SLQSSetLoggingMask . . . 1544
9.13.3.13 SLQSStart . . . 1544
9.13.3.14 SLQSStart_AVAgent . . . 1545
9.13.3.15 SLQSStartSrv . . . 1545
9.14 qaGobiApiDms.h File Reference . . . 1546
9.14.1 Detailed Description . . . 1548
9.14.2 Macro Definition Documentation . . . 1548
9.14.2.1 MAX_CUST_ID_LEN . . . 1548
9.14.2.2 MAX_CUST_VALUE_LEN . . . 1548
9.14.2.3 MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1548
9.14.2.4 MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1548
9.14.2.5 MAX_FSN_LENGTH . . . 1548
9.14.3 Typedef Documentation . . . 1548
9.14.3.1 custFeaturesInfo . . . 1548
9.14.3.2 custFeaturesSetting . . . 1550
9.14.3.3 dmsCurrentPRLInfo . . . 1552
9.14.3.4 ERIFileparams . . . 1552
9.14.3.5 serialNumbersInfo . . . 1552
9.14.3.6 SLQSSwiGetHostDevInfoParams . . . 1553

9.14.3.7 SLQSSwiGetOSInfoParams . . . 1553
9.14.3.8 SLQSSwiGetSerialNoExtParams . . . 1554
9.14.3.9 SLQSSwiSetHostDevInfoParams . . . 1554
9.14.3.10 SLQSSwiSetOSInfoParams . . . 1555
9.14.4 Function Documentation . . . 1555
9.14.4.1 ActivateAutomatic . . . 1555
9.14.4.2 GetActivationState . . . 1556
9.14.4.3 GetDeviceCapabilities . . . 1556
9.14.4.4 GetFirmwareRevision . . . 1557
9.14.4.5 GetFirmwareRevisions . . . 1558
9.14.4.6 GetHardwareRevision . . . 1558
9.14.4.7 GetIMSI . . . 1559
9.14.4.8 GetManufacturer . . . 1559
9.14.4.9 GetModelID . . . 1560
9.14.4.10 GetNetworkTime . . . 1560
9.14.4.11 GetOfflineReason . . . 1561
9.14.4.12 GetPower . . . 1562
9.14.4.13 GetPRLVersion . . . 1562
9.14.4.14 GetSerialNumbers . . . 1563
9.14.4.15 GetVoiceNumber . . . 1563
9.14.4.16 ResetToFactoryDefaults . . . 1564
9.14.4.17 SetPower . . . 1564
9.14.4.18 SLQSDmsSwiGetResetInfo . . . 1565
9.14.4.19 SLQSDmsSwiIndicationRegister . . . 1565
9.14.4.20 SLQSGetBandCapabilities . . . 1565
9.14.4.21 SLQSGetBandCapability . . . 1566
9.14.4.22 SLQSGetCurrentPRLInfo . . . 1567
9.14.4.23 SLQSGetCustFeatures . . . 1567
9.14.4.24 SLQSGetCustFeaturesV2 . . . 1568
9.14.4.25 SLQSGetERIFile . . . 1568
9.14.4.26 SLQSGetSerialNumbers . . . 1569
9.14.4.27 SLQSSetCustFeatures . . . 1569
9.14.4.28 SLQSSetCustFeaturesV2 . . . 1569
9.14.4.29 SLQSSwiClearDyingGaspStatistics . . . 1570
9.14.4.30 SLQSSwiGetCrashAction . . . 1570
9.14.4.31 SLQSSwiGetCrashInfo . . . 1570
9.14.4.32 SLQSSwiGetDyingGaspCfg . . . 1571
9.14.4.33 SLQSSwiGetDyingGaspStatistics . . . 1571
9.14.4.34 SLQSSwiGetFSN . . . 1571
9.14.4.35 SLQSSwiGetFwUpdateStatus . . . 1572

9.14.4.36 SLQSSwiGetHostDevInfo . . . 1572
9.14.4.37 SLQSSwiGetOSInfo . . . 1573
9.14.4.38 SLQSSwiGetSerialNoExt . . . 1573
9.14.4.39 SLQSSwiGetUSBComp . . . 1574
9.14.4.40 SLQSSwiSetCrashAction . . . 1574
9.14.4.41 SLQSSwiSetDyingGaspCfg . . . 1575
9.14.4.42 SLQSSwiSetHostDevInfo . . . 1575
9.14.4.43 SLQSSwiSetOSInfo . . . 1575
9.14.4.44 SLQSSwiSetUSBComp . . . 1576
9.14.4.45 SLQSUIMGetState . . . 1576
9.14.4.46 UIMChangePIN . . . 1577
9.14.4.47 UIMGetControlKeyStatus . . . 1577
9.14.4.48 UIMGetICCID . . . 1578
9.14.4.49 UIMGetPINStatus . . . 1579
9.14.4.50 UIMSetControlKeyProtection . . . 1580
9.14.4.51 UIMSetPINProtection . . . 1580
9.14.4.52 UIMUnblockControlKey . . . 1581
9.14.4.53 UIMUnblockPIN . . . 1582
9.14.4.54 UIMVerifyPIN . . . 1583
9.14.4.55 ValidateSPC . . . 1584
9.15 qaGobiApiFms.h File Reference . . . 1584
9.15.1 Detailed Description . . . 1587
9.15.2 Macro Definition Documentation . . . 1588
9.15.2.1 BUILD_ID_LEN . . . 1588
9.15.2.2 BUILD_ID_MAX_LEN . . . 1588
9.15.2.3 DEVICE_OFFLINE . . . 1588
9.15.2.4 DEVICE_RESET . . . 1588
9.15.2.5 DEVICE_SHUTDOWN . . . 1588
9.15.2.6 FIRMWARE_UPDATE_FAIL . . . 1588
9.15.2.7 FIRMWARE_UPDATE_SUCCESS . . . 1588
9.15.2.8 FIRMWARE_UPGRADE_SUCCESS . . . 1588
9.15.2.9 G3K_FIRMWARE_DOWNLOAD . . . 1588
9.15.2.10 GOBI_LISTENTRIES_MAX . . . 1588
9.15.2.11 GOBI_MBN_BUILD_ID_STR_LEN . . . 1588
9.15.2.12 GOBI_MBN_IMG_ID_STR_LEN . . . 1588
9.15.2.13 GOBI_SET_IMG_PREF_RSPLEN . . . 1588
9.15.2.14 IMG_ID_LEN . . . 1588
9.15.2.15 IMGDETAILS_LEN . . . 1588
9.15.2.16 MAX_IMAGE_IDE_ELEMENTS . . . 1588
9.15.2.17 PRI_UPDATE_FAIL . . . 1588

9.15.2.18 SLQSFWINFO_APPVERSION_SZ . . . 1588
9.15.2.19 SLQSFWINFO_BOOTVERSION_SZ . . . 1588
9.15.2.20 SLQSFWINFO_CARRIER_SZ . . . 1588
9.15.2.21 SLQSFWINFO_CUR_CARR_NAME . . . 1588
9.15.2.22 SLQSFWINFO_CUR_CARR_REV . . . 1588
9.15.2.23 SLQSFWINFO_MODELID_SZ . . . 1588
9.15.2.24 SLQSFWINFO_PACKAGEID_SZ . . . 1588
9.15.2.25 SLQSFWINFO_PRIVERSION_SZ . . . 1588
9.15.2.26 SLQSFWINFO_SKU_SZ . . . 1588
9.15.2.27 SPKG_FIRMWARE_DOWNLOAD . . . 1588
9.15.2.28 UNIQUE_ID_LEN . . . 1589
9.15.3 Enumeration Type Documentation . . . 1589
9.15.3.1 eGobiDeviceSeries . . . 1589
9.15.3.2 eGobiImageCarrier . . . 1589
9.15.3.3 eGobiImageGPS . . . 1590
9.15.3.4 eGobiImageRegion . . . 1590
9.15.3.5 eGobiImageTech . . . 1591
9.15.4 Function Documentation . . . 1591
9.15.4.1 DeleteStoredImage . . . 1591
9.15.4.2 DownloadToSlot . . . 1591
9.15.4.3 eGetDeviceSeries . . . 1592
9.15.4.4 GetImagesPreference . . . 1592
9.15.4.5 GetImageStore . . . 1593
9.15.4.6 GetStoredImages . . . 1593
9.15.4.7 SetImagesPreference . . . 1594
9.15.4.8 SLQSDownloadFirmwareToSlot . . . 1594
9.15.4.9 SLQSGetBootVersionNumber . . . 1595
9.15.4.10 SLQSGetFirmwareInfo . . . 1596
9.15.4.11 SLQSGetImageInfo . . . 1596
9.15.4.12 SLQSGetImageInfo_9x15 . . . 1597
9.15.4.13 SLQSGetImageInfoMC77xx . . . 1597
9.15.4.14 SLQSGetImageInfoMC83xx . . . 1598
9.15.4.15 SLQSGetValidFwPriCombinations . . . 1599
9.15.4.16 SLQSIsSpkgFormatRequired . . . 1599
9.15.4.17 SLQSSetCrashStateCheckIgnore . . . 1599
9.15.4.18 SLQSSetSIMBasedImageSwitching . . . 1600
9.15.4.19 SLQSSetSpkgFormatRequired . . . 1600
9.15.4.20 SLQSSwiGetAllCarrierImages . . . 1600
9.15.4.21 SLQSSwiGetFirmwareCurr . . . 1601
9.15.4.22 SLQSUpgradeFirmware9x15 . . . 1601

9.15.4.23 upgrade_mc77xx_fw . . . 1602
9.15.4.24 UpgradeFirmware2k . . . 1602
9.16 qaGobiApiIms.h File Reference . . . 1603
9.16.1 Detailed Description . . . 1604
9.16.2 Function Documentation . . . 1604
9.16.2.1 SLQSGetIMSSMSConfig . . . 1604
9.16.2.2 SLQSGetIMSUserConfig . . . 1604
9.16.2.3 SLQSGetIMSVoIPConfig . . . 1605
9.16.2.4 SLQSGetRegMgrConfig . . . 1605
9.16.2.5 SLQSGetSIPConfig . . . 1606
9.16.2.6 SLQSImsConfigIndicationRegister . . . 1606
9.16.2.7 SLQSSetIMSSMSConfig . . . 1607
9.16.2.8 SLQSSetIMSUserConfig . . . 1607
9.16.2.9 SLQSSetIMSVoIPConfig . . . 1608
9.16.2.10 SLQSSetRegMgrConfig . . . 1608
9.16.2.11 SLQSSetSIPConfig . . . 1609
9.17 qaGobiApiImsa.h File Reference . . . 1609
9.17.1 Detailed Description . . . 1610
9.17.2 Function Documentation . . . 1610
9.17.2.1 SLQSGetIMSARegStatus . . . 1610
9.17.2.2 SLQSGetIMSAServiceStatus . . . 1610
9.17.2.3 SLQSGetIMSASupportedFields . . . 1611
9.17.2.4 SLQSGetIMSASupportedMsg . . . 1611
9.17.2.5 SLQSRegisterIMSAIndication . . . 1612
9.18 qaGobiApiLoc.h File Reference . . . 1612
9.18.1 Detailed Description . . . 1613
9.18.2 Macro Definition Documentation . . . 1613
9.18.2.1 MAX_SENSOR_DATA_LEN . . . 1613
9.18.2.2 MAX_TEMP_DATA_LEN . . . 1613
9.18.3 Function Documentation . . . 1613
9.18.3.1 SLQSLOCDelAssData . . . 1613
9.18.3.2 SLQSLOCEventRegister . . . 1614
9.18.3.3 SLQSLOCGetBestAvailPos . . . 1614
9.18.3.4 SLQSLOCInjectPosition . . . 1615
9.18.3.5 SLQSLOCInjectSensorData . . . 1615
9.18.3.6 SLQSLOCInjectUTCTime . . . 1616
9.18.3.7 SLQSLOCSetCradleMountConfig . . . 1616
9.18.3.8 SLQSLOCSetExtPowerState . . . 1616
9.18.3.9 SLQSLOCSetOpMode . . . 1617
9.18.3.10 SLQSLOCStart . . . 1617

9.18.3.11 SLQSLOCStop . . . 1618
9.18.3.12 SwiLocGetAutoStart . . . 1618
9.18.3.13 SwiLocSetAutoStart . . . 1619
9.19 qaGobiApiNas.h File Reference . . . 1619
9.19.1 Detailed Description . . . 1624
9.19.2 Macro Definition Documentation . . . 1624
9.19.2.1 IMSI_M_S1_LENGTH . . . 1624
9.19.2.2 IMSI_M_S2_LENGTH . . . 1624
9.19.2.3 MAX_DATA_SRV_CAPABILITIES . . . 1624
9.19.2.4 MAX_DESCRIPTION_LENGTH . . . 1624
9.19.2.5 MAX_PILOT_SETS . . . 1624
9.19.2.6 MAX_SERV_SYSTEM_RADIO_INTERFACES . . . 1624
9.19.2.7 NAM_NAME_LENGTH . . . 1624
9.19.2.8 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1625
9.19.2.9 NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE . . . 1625
9.19.2.10 NAS_SIG_INFO_MIN_dB_FLOAT_VALUE . . . 1625
9.19.2.11 NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE . . . 1625
9.19.2.12 PLMN_LENGTH . . . 1625
9.19.2.13 SLQS_SS_INFO_LIST_MAX_ELEMENTS . . . 1625
9.19.2.14 SLQS_SYSTEM_ID_SIZE . . . 1625
9.19.2.15 UATISIZE . . . 1625
9.19.3 Typedef Documentation . . . 1625
9.19.3.1 SlqsNas3GppNetworkRAT . . . 1625
9.19.3.2 slqsNetworkScanInfo . . . 1625
9.19.3.3 sysSelectPrefInfo . . . 1626
9.19.3.4 sysSelectPrefParams . . . 1630
9.19.4 Enumeration Type Documentation . . . 1636
9.19.4.1 _NAMS_RADIO_IF_TECHNOLOGY_ . . . 1636
9.19.4.2 eSYS_SRV_DOMAIN . . . 1636
9.19.4.3 NAS_LTE_CPHY_CA_BW_NRB . . . 1637
9.19.4.4 NAS_LTE_CPHY_SCELL_STATE . . . 1637
9.19.5 Function Documentation . . . 1637
9.19.5.1 GetACCOLC . . . 1637
9.19.5.2 GetANAAAAuthenticationStatus . . . 1637
9.19.5.3 GetCDMANetworkParameters . . . 1638
9.19.5.4 GetHomeNetwork . . . 1640
9.19.5.5 GetHomeNetwork3GPP2 . . . 1640
9.19.5.6 GetNetworkPreference . . . 1642
9.19.5.7 GetRFInfo . . . 1643
9.19.5.8 GetServingNetwork . . . 1643

9.19.5.9 GetServingNetworkCapabilities . . . 1644
9.19.5.10 GetSignalStrengths . . . 1645
9.19.5.11 InitiateDomainAttach . . . 1646
9.19.5.12 InitiateNetworkRegistration . . . 1646
9.19.5.13 PerformNetworkScan . . . 1647
9.19.5.14 SetACCOLC . . . 1647
9.19.5.15 SetCDMANetworkParameters . . . 1648
9.19.5.16 SetNetworkPreference . . . 1649
9.19.5.17 SLQSConfigSigInfo . . . 1650
9.19.5.18 SLQSGetErrorRate . . . 1651
9.19.5.19 SLQSGetNetworkTime . . . 1651
9.19.5.20 SLQSGetOperatorNameData . . . 1651
9.19.5.21 SLQSGetPLMNName . . . 1652
9.19.5.22 SLQSGetServingSystem . . . 1652
9.19.5.23 SLQSGetSignalStrength . . . 1653
9.19.5.24 SLQSGetSysSelectionPref . . . 1653
9.19.5.25 SLQSInitiateNetworkRegistration . . . 1654
9.19.5.26 SLQSNasConfigSigInfo2 . . . 1654
9.19.5.27 SLQSNasGet3GPP2Subscription . . . 1655
9.19.5.28 SLQSNasGetCellLocationInfo . . . 1655
9.19.5.29 SLQSNasGetHDRColorCode . . . 1656
9.19.5.30 SLQSNASGetLTECPHYCaInfo . . . 1656
9.19.5.31 SLQSNasGetSigInfo . . . 1656
9.19.5.32 SLQSNasGetSysInfo . . . 1657
9.19.5.33 SLQSNasGetTxRxInfo . . . 1657
9.19.5.34 SLQSNasIndicationRegister . . . 1658
9.19.5.35 SLQSNasIndicationRegisterExt . . . 1659
9.19.5.36 SLQSNasIndicationRegisterLTECphyCa . . . 1659
9.19.5.37 SLQSNASSwiGetChannelLock . . . 1659
9.19.5.38 SLQSNasSwiIndicationRegister . . . 1660
9.19.5.39 SLQSNasSwiModemStatus . . . 1660
9.19.5.40 SLQSNASSwiSetChannelLock . . . 1661
9.19.5.41 SLQSPerformNetworkScan . . . 1661
9.19.5.42 SLQSSetBandPreference . . . 1662
9.19.5.43 SLQSSetSysSelectionPref . . . 1663
9.19.5.44 SLQSSwiGetHDRPersonality . . . 1663
9.19.5.45 SLQSSwiGetHDRProtSubtype . . . 1664
9.19.5.46 SLQSSwiGetHRPDStats . . . 1664
9.19.5.47 SLQSSwiGetLteCQI . . . 1664
9.19.5.48 SLQSSwiGetLteSccRxInfo . . . 1665

9.19.5.49 SLQSSwiNetworkDebug . . . 1665
9.19.5.50 SLQSSwiPSDetach . . . 1666
9.20 qaGobiApiOmadm.h File Reference . . . 1666
9.20.1 Detailed Description . . . 1666
9.20.2 Function Documentation . . . 1666
9.20.2.1 OMADMCancelSession . . . 1666
9.20.2.2 OMADMGetPendingNIA . . . 1667
9.20.2.3 OMADMGetSessionInfo . . . 1667
9.20.2.4 OMADMStartSession . . . 1669
9.21 qaGobiApiPds.h File Reference . . . 1669
9.21.1 Detailed Description . . . 1670
9.21.2 Macro Definition Documentation . . . 1670
9.21.2.1 DEFAULTBYTEVALUE . . . 1670
9.21.2.2 DEFAULTLONGVALUE . . . 1670
9.21.2.3 DEFAULTWORDVALUE . . . 1670
9.21.3 Enumeration Type Documentation . . . 1670
9.21.3.1 anonymous enum . . . 1670
9.21.4 Function Documentation . . . 1671
9.21.4.1 ForceXTRADownload . . . 1671
9.21.4.2 GetPDSDefaults . . . 1671
9.21.4.3 GetPDSState . . . 1672
9.21.4.4 GetPortAutomaticTracking . . . 1672
9.21.4.5 GetServiceAutomaticTracking . . . 1673
9.21.4.6 GetXTRAAutomaticDownload . . . 1673
9.21.4.7 GetXTRANetwork . . . 1674
9.21.4.8 GetXTRAValidity . . . 1674
9.21.4.9 PDSInjectTimeReference . . . 1675
9.21.4.10 ResetPDSData . . . 1675
9.21.4.11 SetPDSDefaults . . . 1676
9.21.4.12 SetPDSState . . . 1677
9.21.4.13 SetPortAutomaticTracking . . . 1677
9.21.4.14 SetServiceAutomaticTracking . . . 1678
9.21.4.15 SetXTRAAutomaticDownload . . . 1678
9.21.4.16 SetXTRANetwork . . . 1679
9.21.4.17 SLQSGetAGPSConfig . . . 1679
9.21.4.18 SLQSGetGPSStateInfo . . . 1680
9.21.4.19 SLQSPDSDeterminePosition . . . 1680
9.21.4.20 SLQSPDSInjectAbsoluteTimeReference . . . 1681
9.21.4.21 SLQSPDSInjectPositionData . . . 1681
9.21.4.22 SLQSSetAGPSConfig . . . 1682

9.21.4.23 SLQSSetPositionMethodState . . . 1682
9.21.4.24 StartPDSTrackingSessionExt . . . 1683
9.21.4.25 StopPDSTrackingSession . . . 1684
9.22 qaGobiApiQos.h File Reference . . . 1684
9.22.1 Detailed Description . . . 1685
9.22.2 Macro Definition Documentation . . . 1685
9.22.2.1 MAX_QOS_FILTER_TLV . . . 1685
9.22.2.2 MAX_QOS_SPEC_PER_APN . . . 1685
9.22.3 Function Documentation . . . 1685
9.22.3.1 SLQSQosGetFlowStatus . . . 1685
9.22.3.2 SLQSQosGetGranted . . . 1685
9.22.3.3 SLQSQosGetNetworkStatus . . . 1686
9.22.3.4 SLQSQosGetNWProf . . . 1686
9.22.3.5 SLQSQosModify . . . 1687
9.22.3.6 SLQSQosRel . . . 1687
9.22.3.7 SLQSQosReq . . . 1688
9.22.3.8 SLQSQosReset . . . 1688
9.22.3.9 SLQSQosResume . . . 1689
9.22.3.10 SLQSQosSuspend . . . 1689
9.22.3.11 SLQSQosSwiReadApnExtraParams . . . 1690
9.22.3.12 SLQSQosSwiReadDataStats . . . 1690
9.23 qaGobiApiRms.h File Reference . . . 1690
9.23.1 Detailed Description . . . 1691
9.23.2 Function Documentation . . . 1691
9.23.2.1 GetSMSWake . . . 1691
9.23.2.2 SetSMSWake . . . 1691
9.24 qaGobiApiSar.h File Reference . . . 1692
9.24.1 Detailed Description . . . 1692
9.24.2 Enumeration Type Documentation . . . 1692
9.24.2.1 eQMISARRFState . . . 1692
9.24.3 Function Documentation . . . 1693
9.24.3.1 SLQSGetRfSarState . . . 1693
9.24.3.2 SLQSSetRfSarState . . . 1694
9.25 qaGobiApiSms.h File Reference . . . 1694
9.25.1 Detailed Description . . . 1696
9.25.2 Macro Definition Documentation . . . 1696
9.25.2.1 ABSOLUTE_VALIDITY . . . 1696
9.25.2.2 CONFIG_LEN . . . 1696
9.25.2.3 MAX_SMS_ROUTES . . . 1696
9.25.2.4 NUM_OF_SET . . . 1696

9.25.2.5 TIME_DATE_BUF . . . 1696
9.25.2.6 TIME_STAMP_BUF . . . 1696
9.25.3 Typedef Documentation . . . 1696
9.25.3.1 getIndicationRegResp . . . 1697
9.25.3.2 getTransLayerInfoResp . . . 1697
9.25.3.3 getTransNWRegInfoResp . . . 1698
9.25.3.4 qaQmi3GPP2BroadcastCfgInfo . . . 1698
9.25.3.5 qaQmi3GPPBroadcastCfgInfo . . . 1698
9.25.3.6 setIndicationRegReq . . . 1699
9.25.3.7 transLayerInfo . . . 1699
9.25.4 Function Documentation . . . 1700
9.25.4.1 GetSMSCAddress . . . 1700
9.25.4.2 SaveSMS . . . 1700
9.25.4.3 SendSMS . . . 1701
9.25.4.4 SetSMSCAddress . . . 1702
9.25.4.5 SLQSCDMADecodeMTTextMsg . . . 1703
9.25.4.6 SLQSCDMAEncodeMOTextMsg . . . 1703
9.25.4.7 SLQSDeleteSMS . . . 1703
9.25.4.8 SLQSGetIndicationRegister . . . 1704
9.25.4.9 SLQSGetMessageWaiting . . . 1705
9.25.4.10 SLQSGetSMS . . . 1705
9.25.4.11 SLQSGetSmsBroadcastConfig . . . 1706
9.25.4.12 SLQSGetSMSList . . . 1707
9.25.4.13 SLQSGetTransLayerInfo . . . 1708
9.25.4.14 SLQSGetTransNWRegInfo . . . 1708
9.25.4.15 SLQSModifySMSStatus . . . 1708
9.25.4.16 SLQSSendAsyncSMS . . . 1709
9.25.4.17 SLQSSendLongSMS . . . 1710
9.25.4.18 SLQSSendSMS . . . 1710
9.25.4.19 SLQSSetIndicationRegister . . . 1711
9.25.4.20 SLQSSetSmsBroadcastActivation . . . 1711
9.25.4.21 SLQSSetSmsBroadcastConfig . . . 1712
9.25.4.22 SLQSSetSmsStorage . . . 1712
9.25.4.23 SLQSSmsGetMaxStorageSize . . . 1713
9.25.4.24 SLQSSmsGetMessageProtocol . . . 1713
9.25.4.25 SLQSSmsSetRoutes . . . 1714
9.25.4.26 SLQSSwiGetSMSStorage . . . 1714
9.25.4.27 SLQSWCDMADecodeLongTextMsg . . . 1715
9.25.4.28 SLQSWCDMADecodeMTTextMsg . . . 1715
9.25.4.29 SLQSWCDMAEncodeMOTextMsg . . . 1716

9.26 qaGobiApiSwi.h File Reference . . . 1716
9.26.1 Detailed Description . . . 1716
9.26.2 Function Documentation . . . 1716
9.26.2.1 SLQSGetPidof . . . 1716
9.26.2.2 SLQSGetSdkVersion . . . 1717
9.26.2.3 SLQSSendRawQMI . . . 1717
9.27 qaGobiApiSwiAudio.h File Reference . . . 1717
9.27.1 Detailed Description . . . 1718
9.27.2 Macro Definition Documentation . . . 1718
9.27.2.1 MAX_LEN_IFACE_TABLE . . . 1718
9.27.3 Function Documentation . . . 1718
9.27.3.1 SLQSGetM2MAudioProfile . . . 1718
9.27.3.2 SLQSGetM2MAudioVolume . . . 1719
9.27.3.3 SLQSGetM2MAVMute . . . 1719
9.27.3.4 SLQSGetM2MSpkrGain . . . 1720
9.27.3.5 SLQSSetM2MAudioAVCFG . . . 1720
9.27.3.6 SLQSSetM2MAudioLPBK . . . 1720
9.27.3.7 SLQSSetM2MAudioNVDef . . . 1721
9.27.3.8 SLQSSetM2MAudioProfile . . . 1721
9.27.3.9 SLQSSetM2MAudioVolume . . . 1721
9.27.3.10 SLQSSetM2MAVMute . . . 1722
9.27.3.11 SLQSSetM2MSpkrGain . . . 1722
9.28 qaGobiApiSwiAvms.h File Reference . . . 1723
9.28.1 Detailed Description . . . 1725
9.28.2 Macro Definition Documentation . . . 1725
9.28.2.1 MAX_AVMS_RETRY_TIMER_NUMBER . . . 1725
9.28.2.2 MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH . . . 1725
9.28.2.3 MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH . . . 1725
9.28.2.4 MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH . . . 1725
9.28.2.5 MAX_SWI_AVMS_SET_SETTING_APN_LENGTH . . . 1725
9.28.2.6 MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS . . . 1725
9.28.2.7 MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH . . . 1725
9.28.2.8 MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH . . . 1725
9.28.2.9 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH . . . 1725
9.28.2.10 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH . . . 1725
9.28.2.11 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH . . . 1725
9.28.2.12 QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH . . . 1725
9.28.3 Typedef Documentation . . . 1725

9.28.3.1 QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp . . . 1725
9.28.3.2 QmiSwiAvmsEventReportConfigResp . . . 1726
9.28.3.3 QmiSwiAvmsEventReportConnectionRequestResp . . . 1726
9.28.3.4 QmiSwiAvmsEventReportDataSessionStatusResp . . . 1727
9.28.3.5 QmiSwiAvmsEventReportHTTPStatusResp . . . 1727
9.28.3.6 QmiSwiAvmsEventReportNotificationResp . . . 1728
9.28.3.7 QmiSwiAvmsEventReportPackageIDResp . . . 1729
9.28.3.8 QmiSwiAvmsEventReportRegStatusResp . . . 1729
9.28.3.9 QmiSwiAvmsEventReportSessionTypeResp . . . 1730
9.28.3.10 QmiSwiAvmsEventReportWAMSParamChangeResp . . . 1730
9.28.3.11 QmiSwiAvmsSetSettingsAPNInfo . . . 1730
9.28.3.12 QmiSwiAvmsSetSettingsConnectionRetryTimers . . . 1730
9.28.3.13 QmiSwiAvmsSetSettingsPeriodInfo . . . 1730
9.28.3.14 SLQSAVMSSetSettingsReq . . . 1730
9.28.3.15 SLQSAVMSSettings . . . 1731
9.28.3.16 SLQSAVMSSettingsAPNInfo . . . 1731
9.28.3.17 SLQSAVMSSettingsConnectionRetryTimers . . . 1731
9.28.3.18 SLQSAVMSSettingsPeriodsInfo . . . 1732
9.28.4 Function Documentation . . . 1732
9.28.4.1 SLQSAVMSGetSessionInfo . . . 1732
9.28.4.2 SLQSAVMSGetSettings . . . 1732
9.28.4.3 SLQSAVMSSelection . . . 1732
9.28.4.4 SLQSAVMSSetEventReport . . . 1733
9.28.4.5 SLQSAVMSSetSettings . . . 1733
9.28.4.6 SLQSAVMSSetSettingsIgnoreAutoReboot . . . 1734
9.28.4.7 SLQSAVMSStartSession . . . 1734
9.28.4.8 SLQSAVMSStopSession . . . 1735
9.29 qaGobiApiSwiOmadms.h File Reference . . . 1735
9.29.1 Detailed Description . . . 1736
9.29.2 Typedef Documentation . . . 1736
9.29.2.1 SLQSOMADMSessionInfo . . . 1736
9.29.2.2 SLQSOMADMSettings . . . 1738
9.29.2.3 SLQSOMADMSettingsReqParams . . . 1739
9.29.2.4 SLQSOMADMSettingsReqParams3 . . . 1740
9.29.3 Function Documentation . . . 1741
9.29.3.1 SLQSOMADMCancelSession . . . 1741
9.29.3.2 SLQSOMADMGetSessionInfo . . . 1741
9.29.3.3 SLQSOMADMGetSettings . . . 1742
9.29.3.4 SLQSOMADMGetSettings2 . . . 1742
9.29.3.5 SLQSOMADMSendSelection . . . 1743

9.29.3.6 SLQSOMADMSendSelection2 . . . 1743
9.29.3.7 SLQSOMADMSetSettings . . . 1744
9.29.3.8 SLQSOMADMSetSettings2 . . . 1745
9.29.3.9 SLQSOMADMSetSettings3 . . . 1745
9.29.3.10 SLQSOMADMStartSession . . . 1745
9.29.3.11 SLQSOMADMStartSession2 . . . 1746
9.30 qaGobiApiTableBandClasses.h File Reference . . . 1746
9.30.1 Detailed Description . . . 1746
9.30.2 Band Classes (Value - Description) . . . 1746
9.30.2.1 LTE Bands . . . 1748
9.31 qaGobiApiTableCallControlReturnReasons.h File Reference . . . 1749
9.31.1 Detailed Description . . . 1749
9.31.2 Call Control Result Reasons (Value - Name - Description) . . . 1749
9.32 qaGobiApiTableCallEndReasons.h File Reference . . . 1750
9.32.1 Detailed Description . . . 1750
9.32.2 Call end reason codes (Code - Reason) . . . 1750
9.32.2.1 Technology-agnostic call end reasons . . . 1750
9.32.2.2 EVDO CDMA 1xEV-DO . . . 1751
9.32.2.3 WCDMA/GSM call end reasons . . . 1751
9.32.2.4 EVDO CDMA 1xEV-DO . . . 1753
9.32.2.5 call end reason type . . . 1754
9.32.2.6 Mobile IP call end reasons (Type=1) . . . 1754
9.32.2.7 Internal call end reasons (Type=2) . . . 1756
9.32.2.8 Call Manager defined call end reasons (Type=3) . . . 1757
9.32.2.9 3GPP specification defined call end reasons (Type=6) . . . 1762
9.32.2.10 PPP call end reasons (Type=7) . . . 1764
9.32.2.11 EHRPD call end reasons (Type=8) . . . 1764
9.32.2.12 IPV6 call end reasons (Type=9) . . . 1765
9.33 qaGobiApiTableCarrierCodes.h File Reference . . . 1765
9.33.1 Detailed Description . . . 1765
9.33.2 Carrier Codes (Number - Carrier) . . . 1765
9.34 qaGobiApiTableCodingScheme.h File Reference . . . 1767
9.34.1 Detailed Description . . . 1767
9.34.2 Call Control Result Reasons (Value - Name - Description) . . . 1767
9.34.2.1 Use of bits 3..0\n\n . . . 1767
9.34.3 Coding Group Bits 7..4(0001) . . . 1768
9.34.3.1 use of bits 3..0 . . . 1768
9.34.4 Coding Group Bits 7..4(0010) . . . 1768
9.34.4.1 use of bits 3..0 . . . 1768
9.34.5 Coding Group Bits 7..4(0011) . . . 1768

9.34.5.1 use of bits 3..0 . . . 1768
9.34.6 Coding Group Bits 7..4(01xx) . . . 1769
9.34.6.1 use of bits 3..0 . . . 1769
9.34.7 Coding Group Bits 7..4(1001) . . . 1769
9.34.7.1 Reserved coding groups . . . 1769
9.34.8 Coding Group Bits 7..4(1010..1101) . . . 1769
9.34.8.1 Reserved coding groups . . . 1769
9.34.9 Coding Group Bits 7..4(1110) . . . 1769
9.34.9.1 Defined by the WAP Forum . . . 1769
9.34.10 Coding Group Bits 7..4 (1111) . . . 1769
9.34.10.1 Data coding / message handling . . . 1770
9.34.11 Macro Definition Documentation . . . 1770
9.34.11.1 __GOBI_API_CODING_SCHEME_H__ . . . 1770
9.35 qaGobiApiTableGpsCapabilityCodes.h File Reference . . . 1770
9.35.1 Detailed Description . . . 1770
9.35.2 GPS capability (Value - Capability) . . . 1770
9.36 qaGobiApiTablePowerModes.h File Reference . . . 1770
9.36.1 Detailed Description . . . 1770
9.36.2 Power Modes (Value - Description) . . . 1771
9.37 qaGobiApiTableRadioInterfaces.h File Reference . . . 1771
9.37.1 Detailed Description . . . 1771
9.37.2 Radio interface . . . 1771
9.37.2.1 Technology (Value - Radio Interface Technology) . . . 1771
9.38 qaGobiApiTableRegionCodes.h File Reference . . . 1772
9.38.1 Detailed Description . . . 1772
9.38.2 Region Codes (Code - Region) . . . 1772
9.39 qaGobiApiTableServiceOptions.h File Reference . . . 1772
9.39.1 Detailed Description . . . 1772
9.39.2 Service Option codes (Code - Reason) . . . 1772
9.39.2.1 Description . . . 1772
9.40 qaGobiApiTableSupServiceInfoClasses.h File Reference . . . 1774
9.40.1 Detailed Description . . . 1774
9.40.2 Supplementary Service Information Classes (Value - Service Class) . . . 1775
9.41 qaGobiApiTableSwiAudio.h File Reference . . . 1775
9.41.1 Detailed Description . . . 1775
9.41.2 ACDB Device (Device ID - description) . . . 1775
9.41.3 Physical Interface (Device ID - description - Interface parameters) . . . 1775
9.42 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference . . . 1775
9.42.1 Detailed Description . . . 1776
9.42.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage) . . . 1776

9.43 qaGobiApiTableVoiceCallEndReasons.h File Reference . . . 1777
9.43.1 Detailed Description . . . 1777
9.43.2 Voice Call and supplementary services end reason codes (Code - Reason) . . . 1777
9.43.2.1 General . . . 1777
9.43.2.2 service Errors . . . 1779
9.43.2.3 control cause values . . . 1780
9.43.2.4 reject causes . . . 1781
9.43.2.5 reject causes . . . 1782
9.43.2.6 reject causes . . . 1782
9.43.2.7 stratum reject causes . . . 1782
9.43.2.8 reject causes . . . 1783
9.43.2.9 IP end reasons . . . 1783
9.44 qaGobiApiTmd.h File Reference . . . 1783
9.44.1 Detailed Description . . . 1784
9.44.2 Macro Definition Documentation . . . 1784
9.44.2.1 MAX_MITIGATION_DEV_ID_LEN . . . 1784
9.44.2.2 MAX_MITIGATION_DEV_LIST_LEN . . . 1784
9.44.3 Function Documentation . . . 1784
9.44.3.1 SLQSTmdDeRegNotMitigationLvl . . . 1784
9.44.3.2 SLQSTmdGetMitigationDevList . . . 1784
9.44.3.3 SLQSTmdGetMitigationLvl . . . 1785
9.44.3.4 SLQSTmdRegNotMitigationLvl . . . 1785
9.45 qaGobiApiUim.h File Reference . . . 1785
9.45.1 Detailed Description . . . 1787
9.45.2 Macro Definition Documentation . . . 1788
9.45.2.1 MAX_ACTIVE_PERS_FEATURES . . . 1788
9.45.2.2 MAX_CONTENT_LENGTH . . . 1788
9.45.2.3 MAX_DESCRIPTION_LENGTH . . . 1788
9.45.2.4 MAX_ICCID_LENGTH . . . 1788
9.45.2.5 MAX_NO_OF_APPLICATIONS . . . 1788
9.45.2.6 MAX_NO_OF_SLOTS . . . 1788
9.45.2.7 MAX_PATH_LENGTH . . . 1788
9.45.2.8 MAX_PUK_LENGTH . . . 1788
9.45.2.9 MAX_SLOTS_STATUS . . . 1788
9.45.3 Function Documentation . . . 1788
9.45.3.1 SLQSUIMAuthenticate . . . 1788
9.45.3.2 SLQSUIMChangePin . . . 1788
9.45.3.3 SLQSUIMDepersonalization . . . 1789
9.45.3.4 SLQSUIMEventRegister . . . 1790
9.45.3.5 SLQSUIMGetCardStatus . . . 1790

9.45.3.6 SLQSUIMGetConfiguration . . . 1791
9.45.3.7 SLQSUIMGetFileAttributes . . . 1791
9.45.3.8 SLQSUIMGetSlotsStatus . . . 1792
9.45.3.9 SLQSUIMPowerDown . . . 1792
9.45.3.10 SLQSUIMPowerUp . . . 1792
9.45.3.11 SLQSUIMReadTransparent . . . 1793
9.45.3.12 SLQSUIMRefreshComplete . . . 1793
9.45.3.13 SLQSUIMRefreshGetLastEvent . . . 1794
9.45.3.14 SLQSUIMRefreshOK . . . 1795
9.45.3.15 SLQSUIMRefreshRegister . . . 1795
9.45.3.16 SLQSUIMReset . . . 1796
9.45.3.17 SLQSUIMSetPinProtection . . . 1796
9.45.3.18 SLQSUIMSwitchSlot . . . 1797
9.45.3.19 SLQSUIMUnblockPin . . . 1797
9.45.3.20 SLQSUIMVerifyPin . . . 1798
9.46 qaGobiApiVoice.h File Reference . . . 1798
9.46.1 Detailed Description . . . 1801
9.46.2 Macro Definition Documentation . . . 1801
9.46.2.1 MAX_CALL_NO_LEN . . . 1801
9.46.2.2 MAX_DESCRIPTION_LENGTH . . . 1801
9.46.2.3 MAX_NO_OF_CALLS . . . 1801
9.46.2.4 MAXUSSDLENGTH . . . 1801
9.46.2.5 PASSWORD_LENGTH . . . 1801
9.46.3 Enumeration Type Documentation . . . 1801
9.46.3.1 serviceClassInformation . . . 1801
9.46.4 Function Documentation . . . 1802
9.46.4.1 AnswerUSSD . . . 1802
9.46.4.2 CancelUSSD . . . 1802
9.46.4.3 OriginateUSSD . . . 1803
9.46.4.4 SLQSOriginateUSSD . . . 1803
9.46.4.5 SLQSVoiceALSSelectLine . . . 1803
9.46.4.6 SLQSVoiceALSSetLineSwitching . . . 1804
9.46.4.7 SLQSVoiceAnswerCall . . . 1804
9.46.4.8 SLQSVoiceBindSubscription . . . 1805
9.46.4.9 SLQSVoiceBurstDTMF . . . 1805
9.46.4.10 SLQSVoiceDialCall . . . 1806
9.46.4.11 SLQSVoiceEndCall . . . 1806
9.46.4.12 SLQSVoiceGetAllCallInfo . . . 1807
9.46.4.13 SLQSVoiceGetCallBarring . . . 1807
9.46.4.14 SLQSVoiceGetCallForwardingStatus . . . 1808

9.46.4.15 SLQSVoiceGetCallInfo . . . 1808
9.46.4.16 SLQSVoiceGetCallWaiting . . . 1809
9.46.4.17 SLQSVoiceGetCLIP . . . 1809
9.46.4.18 SLQSVoiceGetCLIR . . . 1810
9.46.4.19 SLQSVoiceGetCNAP . . . 1811
9.46.4.20 SLQSVoiceGetCOLP . . . 1811
9.46.4.21 SLQSVoiceGetCOLR . . . 1812
9.46.4.22 SLQSVoiceGetConfig . . . 1812
9.46.4.23 SLQSVoiceIndicationRegister . . . 1813
9.46.4.24 SLQSVoiceManageCalls . . . 1813
9.46.4.25 SLQSVoiceOrigUSSDNoWait . . . 1814
9.46.4.26 SLQSVoiceSendFlash . . . 1814
9.46.4.27 SLQSVoiceSetCallBarringPassword . . . 1815
9.46.4.28 SLQSVoiceSetConfig . . . 1815
9.46.4.29 SLQSVoiceSetPreferredPrivacy . . . 1816
9.46.4.30 SLQSVoiceSetSUPSService . . . 1816
9.46.4.31 SLQSVoiceStartContDTMF . . . 1817
9.46.4.32 SLQSVoiceStopContDTMF . . . 1817
9.47 qaGobiApiWds.h File Reference . . . 1818
9.47.1 Detailed Description . . . 1821
9.47.2 Macro Definition Documentation . . . 1822
9.47.2.1 IPV6_ADDRESS_ARRAY_SIZE . . . 1822
9.47.3 Typedef Documentation . . . 1822
9.47.3.1 GetProfileSettingIn . . . 1822
9.47.3.2 GetProfileSettingOut . . . 1822
9.47.3.3 QmiProfileInfo . . . 1822
9.47.3.4 QmiWDSDataBearers . . . 1822
9.47.3.5 QmiWDSDataBearerTechnology . . . 1823
9.47.3.6 slqs3GPPConfigItem . . . 1824
9.47.4 Enumeration Type Documentation . . . 1825
9.47.4.1 qmiDataBearerMasks . . . 1825
9.47.5 Function Documentation . . . 1825
9.47.5.1 GetAutoconnect . . . 1825
9.47.5.2 GetByteTotals . . . 1825
9.47.5.3 GetConnectionRate . . . 1826
9.47.5.4 GetDataBearerTechnology . . . 1827
9.47.5.5 GetDefaultProfile . . . 1828
9.47.5.6 GetDefaultProfileLTE . . . 1829
9.47.5.7 GetDefaultProfileNum . . . 1831
9.47.5.8 GetDormancyState . . . 1831

9.47.5.9 GetIPAddressLTE . . . 1832
9.47.5.10 GetLastMobileIPError . . . 1832
9.47.5.11 GetMobileIP . . . 1833
9.47.5.12 GetMobileIPProfile . . . 1833
9.47.5.13 GetPacketStatistics . . . 1835
9.47.5.14 GetPacketStatus . . . 1835
9.47.5.15 GetSessionDuration . . . 1836
9.47.5.16 GetSessionState . . . 1836
9.47.5.17 iGetByteTotals . . . 1837
9.47.5.18 iGetConnectionRate . . . 1837
9.47.5.19 iGetPacketStatistics . . . 1837
9.47.5.20 iSLQSMISetIPFamilyPreference . . . 1837
9.47.5.21 RMSetTransferStatistics . . . 1837
9.47.5.22 SetActiveMobileIPProfile . . . 1837
9.47.5.23 SetAutoconnect . . . 1838
9.47.5.24 SetDefaultProfile . . . 1838
9.47.5.25 SetDefaultProfileLTE . . . 1840
9.47.5.26 SetDefaultProfileLTEV2 . . . 1841
9.47.5.27 SetDefaultProfileNum . . . 1842
9.47.5.28 SetMobileIP . . . 1843
9.47.5.29 SetMobileIPParameters . . . 1843
9.47.5.30 SetMobileIPProfile . . . 1844
9.47.5.31 SLQSAutoConnect . . . 1845
9.47.5.32 SLQSCreateProfile . . . 1846
9.47.5.33 SLQSDeleteProfile . . . 1846
9.47.5.34 SLQSGet3GPPConfigItem . . . 1847
9.47.5.35 SLQSGetByteTotals . . . 1847
9.47.5.36 SLQSGetConnectionRate . . . 1848
9.47.5.37 SLQSGetCurrDataSystemStat . . . 1848
9.47.5.38 SLQSGetCurrentChannelRate . . . 1849
9.47.5.39 SLQSGetDataBearerTechnology . . . 1849
9.47.5.40 SLQSGetDataBearerTechnologyExt . . . 1850
9.47.5.41 SLQSGetDUNCallInfo . . . 1850
9.47.5.42 SLQSGetPacketStatistics . . . 1851
9.47.5.43 SLQSGetProfile . . . 1851
9.47.5.44 SLQSGetProfileSettings . . . 1853
9.47.5.45 SLQSGetRuntimeSettings . . . 1853
9.47.5.46 SLQSGetSessionState . . . 1853
9.47.5.47 SLQSModifyProfile . . . 1854
9.47.5.48 SLQSResetPacketStatics . . . 1855

9.47.5.49 SLQSSet3GPPConfigItem . . . 1855
9.47.5.50 SLQSSetProfile . . . 1855
9.47.5.51 SLQSSGetDHCPv4ClientConfig . . . 1857
9.47.5.52 SLQSSGetLoopback . . . 1857
9.47.5.53 SLQSSSetDHCPv4ClientConfig . . . 1857
9.47.5.54 SLQSSSetLoopback . . . 1858
9.47.5.55 SLQSStartStopDataSession . . . 1858
9.47.5.56 SLQSWdsGoActive . . . 1859
9.47.5.57 SLQSWdsGoDormant . . . 1859
9.47.5.58 SLQSWdsSetEventReport . . . 1859
9.47.5.59 SLQSWdsSwiPDPRuntimeSettings . . . 1860
9.47.5.60 WDS_IsGobiDevice . . . 1860
9.48 qaNasGetRFBandInfo.h File Reference . . . 1860
9.48.1 Enumeration Type Documentation . . . 1861
9.48.1.1 eQMI_NAS_GET_RF_INFO_RESP . . . 1861
9.48.2 Function Documentation . . . 1861
9.48.2.1 PkQmiNasGetRFBandInfo . . . 1861
9.48.2.2 UpkQmiNasGetRFBandInfo . . . 1861
9.49 qaNasPerformNetworkScan.h File Reference . . . 1861
9.49.1 Macro Definition Documentation . . . 1862
9.49.1.1 FORBIDDEN_INDEX . . . 1862
9.49.1.2 INDEX_ZERO . . . 1862
9.49.1.3 MAX_DESCRIPTION_LENGTH . . . 1862
9.49.1.4 PREFERRED_INDEX . . . 1862
9.49.1.5 QMI_NAS_MAX_INSTANCES . . . 1862
9.49.1.6 QMI_NAS_NETSTATUS_MASK . . . 1862
9.49.1.7 ROAMING_INDEX . . . 1862
9.49.2 Enumeration Type Documentation . . . 1862
9.49.2.1 eQMI_NAS_PERFORM_NETWORK_SCAN_RESP . . . 1862
9.49.3 Function Documentation . . . 1862
9.49.3.1 PkQmiNasPerformNetworkScan . . . 1862
9.49.3.2 UpkQmiNasPerformNetworkScan . . . 1862
9.50 qmerrno.h File Reference . . . 1862
9.50.1 Enumeration Type Documentation . . . 1864
9.50.1.1 eQCWWANError . . . 1864
9.50.1.2 qm_wds_ds_profile_extended_err_codes . . . 1869
9.51 qos.h File Reference . . . 1869
9.51.1 Macro Definition Documentation . . . 1871
9.51.1.1 LITEQMI_MAX_QOS_FILTERS . . . 1871
9.51.1.2 LITEQMI_MAX_QOS_FLOW_PER_APN_STATS . . . 1871

9.51.1.3 LITEQMI_MAX_QOS_FLOWS . . . 1871
9.51.2 Function Documentation . . . 1871
9.51.2.1 pack_qos_SLQSQosGetNetworkStatus . . . 1871
9.51.2.2 pack_qos_SLQSQosSwiReadApnExtraParams . . . 1871
9.51.2.3 pack_qos_SLQSQosSwiReadDataStats . . . 1872
9.51.2.4 pack_qos_SLQSSetQosEventCallback . . . 1873
9.51.2.5 unpack_qos_SLQSQosGetNetworkStatus . . . 1873
9.51.2.6 unpack_qos_SLQSQosSwiReadApnExtraParams . . . 1874
9.51.2.7 unpack_qos_SLQSQosSwiReadDataStats . . . 1874
9.51.2.8 unpack_qos_SLQSSetQosEventCallback . . . 1875
9.51.2.9 unpack_qos_SLQSSetQosEventCallback_ind . . . 1875
9.51.2.10 unpack_qos_SLQSSetQosNWStatusCallback_ind . . . 1876
9.51.2.11 unpack_qos_SLQSSetQosPriEventCallback_ind . . . 1876
9.51.2.12 unpack_qos_SLQSSetQosStatusCallback_ind . . . 1877
9.52 rms.h File Reference . . . 1877
9.52.1 Function Documentation . . . 1878
9.52.1.1 pack_rms_GetSMSWake . . . 1878
9.52.1.2 pack_rms_SetSMSWake . . . 1878
9.52.1.3 unpack_rms_GetSMSWake . . . 1878
9.52.1.4 unpack_rms_SetSMSWake . . . 1879
9.53 sar.h File Reference . . . 1879
9.53.1 Function Documentation . . . 1879
9.53.1.1 pack_sar_SLQSGetRfSarState . . . 1879
9.53.1.2 pack_sar_SLQSSetRfSarState . . . 1880
9.53.1.3 unpack_sar_SLQSGetRfSarState . . . 1880
9.53.1.4 unpack_sar_SLQSSetRfSarState . . . 1881
9.54 sms.h File Reference . . . 1881
9.54.1 Macro Definition Documentation . . . 1884
9.54.1.1 MAX_CDMA_ENC_MO_TXT_MSG_SIZE . . . 1884
9.54.1.2 MAX_MS_TRANSFER_ROUTE_MSG . . . 1884
9.54.1.3 MAX_MSC_ADDRESS_SIZE . . . 1884
9.54.1.4 MAX_MSE_TWS_MSG . . . 1884
9.54.1.5 MAX_SMS_LIST_SIZE . . . 1884
9.54.1.6 MAX_SMS_MESSAGE_SIZE . . . 1884
9.54.1.7 SMS_CONFIG_LEN . . . 1884
9.54.1.8 SMS_MAX_SMS_ROUTES . . . 1884
9.54.1.9 SMS_NUM_OF_SET . . . 1884
9.54.1.10 SMSC_TYPE_LEN . . . 1884
9.54.2 Typedef Documentation . . . 1885
9.54.2.1 sMSCAddressInfo . . . 1885

9.54.2.2 sMSEtwsMessageInfo . . . 1885
9.54.2.3 sMSEtwsPlmnInfo . . . 1885
9.54.2.4 sMSMessageModeInfo . . . 1885
9.54.2.5 sMSMTMessageInfo . . . 1885
9.54.2.6 sMSOnIMSInfo . . . 1885
9.54.2.7 sMSTransferRouteMTMessageInfo . . . 1886
9.54.3 Enumeration Type Documentation . . . 1886
9.54.3.1 eqmiCbkSetStatus . . . 1886
9.54.4 Function Documentation . . . 1886
9.54.4.1 pack_sms_GetSMSCAddress . . . 1886
9.54.4.2 pack_sms_SaveSMS . . . 1887
9.54.4.3 pack_sms_SendSMS . . . 1887
9.54.4.4 pack_sms_SetNewSMSCallback . . . 1887
9.54.4.5 pack_sms_SetSMSCAddress . . . 1888
9.54.4.6 pack_sms_SLQSDeleteSMS . . . 1888
9.54.4.7 pack_sms_SLQSGetIndicationRegister . . . 1888
9.54.4.8 pack_sms_SLQSGetMessageWaiting . . . 1889
9.54.4.9 pack_sms_SLQSGetSMS . . . 1889
9.54.4.10 pack_sms_SLQSGetSmsBroadcastConfig . . . 1890
9.54.4.11 pack_sms_SLQSGetSMSList . . . 1890
9.54.4.12 pack_sms_SLQSGetTransLayerInfo . . . 1890
9.54.4.13 pack_sms_SLQSGetTransNWRegInfo . . . 1891
9.54.4.14 pack_sms_SLQSModifySMSStatus . . . 1891
9.54.4.15 pack_sms_SLQSSendAsyncSMS . . . 1891
9.54.4.16 pack_sms_SLQSSetIndicationRegister . . . 1892
9.54.4.17 pack_sms_SLQSSetSmsBroadcastActivation . . . 1892
9.54.4.18 pack_sms_SLQSSetSmsBroadcastConfig . . . 1892
9.54.4.19 pack_sms_SLQSSetSmsStorage . . . 1893
9.54.4.20 pack_sms_SLQSSmsGetMaxStorageSize . . . 1893
9.54.4.21 pack_sms_SLQSSmsGetMessageProtocol . . . 1894
9.54.4.22 pack_sms_SLQSSmsSetRoutes . . . 1894
9.54.4.23 pack_sms_SLQSSwiGetSMSStorage . . . 1894
9.54.4.24 unpack_sms_GetSMSCAddress . . . 1895
9.54.4.25 unpack_sms_SaveSMS . . . 1895
9.54.4.26 unpack_sms_SendSMS . . . 1895
9.54.4.27 unpack_sms_SetNewSMSCallback . . . 1896
9.54.4.28 unpack_sms_SetNewSMSCallback_ind . . . 1896
9.54.4.29 unpack_sms_SetSMSCAddress . . . 1896
9.54.4.30 unpack_sms_SLQSDeleteSMS . . . 1897
9.54.4.31 unpack_sms_SLQSGetIndicationRegister . . . 1897

9.54.4.32 unpack_sms_SLQSGetMessageWaiting . . . 1897
9.54.4.33 unpack_sms_SLQSGetSMS . . . 1898
9.54.4.34 unpack_sms_SLQSGetSmsBroadcastConfig . . . 1898
9.54.4.35 unpack_sms_SLQSGetSMSList . . . 1898
9.54.4.36 unpack_sms_SLQSGetTransLayerInfo . . . 1899
9.54.4.37 unpack_sms_SLQSGetTransNWRegInfo . . . 1899
9.54.4.38 unpack_sms_SLQSModifySMSStatus . . . 1899
9.54.4.39 unpack_sms_SLQSSendAsyncSMS . . . 1900
9.54.4.40 unpack_sms_SLQSSetIndicationRegister . . . 1900
9.54.4.41 unpack_sms_SLQSSetSmsBroadcastActivation . . . 1900
9.54.4.42 unpack_sms_SLQSSetSmsBroadcastConfig . . . 1901
9.54.4.43 unpack_sms_SLQSSetSmsStorage . . . 1901
9.54.4.44 unpack_sms_SLQSSmsGetMaxStorageSize . . . 1901
9.54.4.45 unpack_sms_SLQSSmsGetMessageProtocol . . . 1902
9.54.4.46 unpack_sms_SLQSSmsSetRoutes . . . 1902
9.54.4.47 unpack_sms_SLQSSwiGetSMSStorage . . . 1902
9.54.4.48 unpack_sms_SLQSWmsMemoryFullCallBack_ind . . . 1903
9.55 SwiDataTypes.h File Reference . . . 1903
9.55.1 Detailed Description . . . 1904
9.55.2 Macro Definition Documentation . . . 1904
9.55.2.1 QMI_NO_LTE_FW_SUPPORT . . . 1904
9.55.2.2 QMI_TLV_PLACEHOLDER . . . 1904
9.55.2.3 SWI_API . . . 1904
9.55.2.4 UNUSEDPARAM . . . 1904
9.55.3 Typedef Documentation . . . 1904
9.55.3.1 BOOL . . . 1904
9.55.3.2 BYTE . . . 1904
9.55.3.3 CHAR . . . 1904
9.55.3.4 FLOAT . . . 1904
9.55.3.5 INT32 . . . 1904
9.55.3.6 INT8 . . . 1904
9.55.3.7 LPCSTR . . . 1904
9.55.3.8 SHORT . . . 1904
9.55.3.9 ULONG . . . 1904
9.55.3.10 ULONGLONG . . . 1904
9.55.3.11 USHORT . . . 1904
9.55.3.12 WORD . . . 1904
9.56 swiloc.h File Reference . . . 1904
9.56.1 Function Documentation . . . 1905
9.56.1.1 pack_swiloc_SwiLocGetAutoStart . . . 1905

9.56.1.2 pack_swiloc_SwiLocSetAutoStart . . . 1905
9.56.1.3 unpack_swiloc_SwiLocGetAutoStart . . . 1905
9.56.1.4 unpack_swiloc_SwiLocSetAutoStart . . . 1906
9.57 swioma.h File Reference . . . 1906
9.57.1 Macro Definition Documentation . . . 1907
9.57.1.1 LITEQMI_MAX_SWIOMA_STR_LEN . . . 1907
9.57.2 Function Documentation . . . 1907
9.57.2.1 pack_swioma_SLQSOMADMAlertCallback . . . 1907
9.57.2.2 pack_swioma_SLQSOMADMCancelSession . . . 1908
9.57.2.3 pack_swioma_SLQSOMADMGetSessionInfo . . . 1908
9.57.2.4 pack_swioma_SLQSOMADMGetSettings . . . 1909
9.57.2.5 pack_swioma_SLQSOMADMSendSelection . . . 1910
9.57.2.6 pack_swioma_SLQSOMADMSetSettings . . . 1910
9.57.2.7 pack_swioma_SLQSOMADMStartSession . . . 1911
9.57.2.8 unpack_swioma_SLQSOMADMAlertCallback . . . 1911
9.57.2.9 unpack_swioma_SLQSOMADMAlertCallback_ind . . . 1912
9.57.2.10 unpack_swioma_SLQSOMADMCancelSession . . . 1912
9.57.2.11 unpack_swioma_SLQSOMADMGetSessionInfo . . . 1913
9.57.2.12 unpack_swioma_SLQSOMADMGetSettings . . . 1913
9.57.2.13 unpack_swioma_SLQSOMADMSendSelection . . . 1914
9.57.2.14 unpack_swioma_SLQSOMADMSetSettings . . . 1914
9.57.2.15 unpack_swioma_SLQSOMADMStartSession . . . 1915
9.58 SWIWWANCMAPI.h File Reference . . . 1915
9.59 uim.h File Reference . . . 1915
9.59.1 Macro Definition Documentation . . . 1918
9.59.1.1 MAX_DESCRIPTION_LENGTH . . . 1918
9.59.1.2 MAX_ICCID_LENGTH . . . 1918
9.59.1.3 MAX_NO_OF_APPLICATIONS . . . 1918
9.59.1.4 MAX_NO_OF_SLOTS . . . 1918
9.59.1.5 MAX_SLOTS_STATUS . . . 1918
9.59.1.6 UIM_MAX_ACTIVE_PERS_FEATURES . . . 1918
9.59.1.7 UIM_MAX_CONTENT_LENGTH . . . 1918
9.59.1.8 UIM_MAX_DESCRIPTION_LENGTH . . . 1918
9.59.1.9 UIM_MAX_NO_OF_APPLICATIONS . . . 1918
9.59.1.10 UIM_MAX_NO_OF_SLOTS . . . 1918
9.59.1.11 UIM_UINT8_MAX_STRING_SZ . . . 1918
9.59.2 Function Documentation . . . 1918
9.59.2.1 pack_uim_ChangePin . . . 1918
9.59.2.2 pack_uim_GetCardStatus . . . 1919
9.59.2.3 pack_uim_ReadTransparent . . . 1919

9.59.2.4 pack_uim_SetPinProtection . . . 1919
9.59.2.5 pack_uim_SLQSUIMAuthenticate . . . 1920
9.59.2.6 pack_uim_SLQSUIMDepersonalization . . . 1920
9.59.2.7 pack_uim_SLQSUIMEventRegister . . . 1921
9.59.2.8 pack_uim_SLQSUIMGetConfiguration . . . 1921
9.59.2.9 pack_uim_SLQSUIMGetFileAttributes . . . 1921
9.59.2.10 pack_uim_SLQSUIMGetSlotsStatus . . . 1922
9.59.2.11 pack_uim_SLQSUIMPowerDown . . . 1922
9.59.2.12 pack_uim_SLQSUIMPowerUp . . . 1922
9.59.2.13 pack_uim_SLQSUIMRefreshComplete . . . 1923
9.59.2.14 pack_uim_SLQSUIMRefreshGetLastEvent . . . 1923
9.59.2.15 pack_uim_SLQSUIMRefreshOK . . . 1923
9.59.2.16 pack_uim_SLQSUIMRefreshRegister . . . 1924
9.59.2.17 pack_uim_SLQSUIMReset . . . 1924
9.59.2.18 pack_uim_SLQSUIMSwitchSlot . . . 1925
9.59.2.19 pack_uim_UnblockPin . . . 1925
9.59.2.20 pack_uim_VerifyPin . . . 1925
9.59.2.21 unpack_uim_ChangePin . . . 1926
9.59.2.22 unpack_uim_GetCardStatus . . . 1926
9.59.2.23 unpack_uim_ReadTransparent . . . 1926
9.59.2.24 unpack_uim_SetPinProtection . . . 1927
9.59.2.25 unpack_uim_SetUimSlotStatusChangeCallback_ind . . . 1927
9.59.2.26 unpack_uim_SLQSUIMAuthenticate . . . 1927
9.59.2.27 unpack_uim_SLQSUIMDepersonalization . . . 1928
9.59.2.28 unpack_uim_SLQSUIMEventRegister . . . 1928
9.59.2.29 unpack_uim_SLQSUIMGetConfiguration . . . 1929
9.59.2.30 unpack_uim_SLQSUIMGetFileAttributes . . . 1929
9.59.2.31 unpack_uim_SLQSUIMGetSlotsStatus . . . 1929
9.59.2.32 unpack_uim_SLQSUIMPowerDown . . . 1930
9.59.2.33 unpack_uim_SLQSUIMPowerUp . . . 1930
9.59.2.34 unpack_uim_SLQSUIMRefreshComplete . . . 1930
9.59.2.35 unpack_uim_SLQSUIMRefreshGetLastEvent . . . 1931
9.59.2.36 unpack_uim_SLQSUIMRefreshOK . . . 1931
9.59.2.37 unpack_uim_SLQSUIMRefreshRegister . . . 1931
9.59.2.38 unpack_uim_SLQSUIMReset . . . 1932
9.59.2.39 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind . . . 1932
9.59.2.40 unpack_uim_SLQSUIMSwitchSlot . . . 1932
9.59.2.41 unpack_uim_UnblockPin . . . 1933
9.59.2.42 unpack_uim_VerifyPin . . . 1933
9.60 voice.h File Reference . . . 1933

9.60.1 Macro Definition Documentation . . . 1937
9.60.1.1 BARRING_PASSWORD_LENGTH . . . 1937
9.60.1.2 MAX_VOICE_CALL_NO_LEN . . . 1937
9.60.1.3 MAX_VOICE_DESCRIPTION_LENGTH . . . 1937
9.60.1.4 MAXVOICEUSSDLENGTH . . . 1937
9.60.1.5 VOICE_MAX_NO_OF_CALLS . . . 1937
9.60.2 Function Documentation . . . 1938
9.60.2.1 pack_voice_AnswerUSSD . . . 1938
9.60.2.2 pack_voice_CancelUSSD . . . 1938
9.60.2.3 pack_voice_OriginateUSSD . . . 1938
9.60.2.4 pack_voice_SLQSOriginateUSSD . . . 1939
9.60.2.5 pack_voice_SLQSVoiceALSSelectLine . . . 1939
9.60.2.6 pack_voice_SLQSVoiceALSSetLineSwitching . . . 1939
9.60.2.7 pack_voice_SLQSVoiceAnswerCall . . . 1940
9.60.2.8 pack_voice_SLQSVoiceBindSubscription . . . 1940
9.60.2.9 pack_voice_SLQSVoiceBurstDTMF . . . 1941
9.60.2.10 pack_voice_SLQSVoiceDialCall . . . 1941
9.60.2.11 pack_voice_SLQSVoiceEndCall . . . 1941
9.60.2.12 pack_voice_SLQSVoiceGetAllCallInfo . . . 1942
9.60.2.13 pack_voice_SLQSVoiceGetCallBarring . . . 1942
9.60.2.14 pack_voice_SLQSVoiceGetCallForwardingStatus . . . 1942
9.60.2.15 pack_voice_SLQSVoiceGetCallInfo . . . 1943
9.60.2.16 pack_voice_SLQSVoiceGetCallWaiting . . . 1943
9.60.2.17 pack_voice_SLQSVoiceGetCLIP . . . 1944
9.60.2.18 pack_voice_SLQSVoiceGetCLIR . . . 1944
9.60.2.19 pack_voice_SLQSVoiceGetCNAP . . . 1944
9.60.2.20 pack_voice_SLQSVoiceGetCOLP . . . 1945
9.60.2.21 pack_voice_SLQSVoiceGetCOLR . . . 1945
9.60.2.22 pack_voice_SLQSVoiceGetConfig . . . 1945
9.60.2.23 pack_voice_SLQSVoiceIndicationRegister . . . 1946
9.60.2.24 pack_voice_SLQSVoiceManageCalls . . . 1946
9.60.2.25 pack_voice_SLQSVoiceOrigUSSDNoWait . . . 1946
9.60.2.26 pack_voice_SLQSVoiceSendFlash . . . 1947
9.60.2.27 pack_voice_SLQSVoiceSetCallBarringPassword . . . 1947
9.60.2.28 pack_voice_SLQSVoiceSetConfig . . . 1947
9.60.2.29 pack_voice_SLQSVoiceSetPreferredPrivacy . . . 1948
9.60.2.30 pack_voice_SLQSVoiceSetSUPSService . . . 1948
9.60.2.31 pack_voice_SLQSVoiceStartContDTMF . . . 1949
9.60.2.32 pack_voice_SLQSVoiceStopContDTMF . . . 1949
9.60.2.33 unpack_voice_AnswerUSSD . . . 1949

9.60.2.34 unpack_voice_CancelUSSD . . . 1950
9.60.2.35 unpack_voice_OriginateUSSD . . . 1950
9.60.2.36 unpack_voice_SLQSOriginateUSSD . . . 1950
9.60.2.37 unpack_voice_SLQSVoiceALSSelectLine . . . 1951
9.60.2.38 unpack_voice_SLQSVoiceALSSetLineSwitching . . . 1951
9.60.2.39 unpack_voice_SLQSVoiceAnswerCall . . . 1951
9.60.2.40 unpack_voice_SLQSVoiceBindSubscription . . . 1952
9.60.2.41 unpack_voice_SLQSVoiceBurstDTMF . . . 1952
9.60.2.42 unpack_voice_SLQSVoiceDialCall . . . 1952
9.60.2.43 unpack_voice_SLQSVoiceEndCall . . . 1953
9.60.2.44 unpack_voice_SLQSVoiceGetAllCallInfo . . . 1953
9.60.2.45 unpack_voice_SLQSVoiceGetCallBarring . . . 1953
9.60.2.46 unpack_voice_SLQSVoiceGetCallForwardingStatus . . . 1954
9.60.2.47 unpack_voice_SLQSVoiceGetCallInfo . . . 1954
9.60.2.48 unpack_voice_SLQSVoiceGetCallWaiting . . . 1954
9.60.2.49 unpack_voice_SLQSVoiceGetCLIP . . . 1955
9.60.2.50 unpack_voice_SLQSVoiceGetCLIR . . . 1955
9.60.2.51 unpack_voice_SLQSVoiceGetCNAP . . . 1955
9.60.2.52 unpack_voice_SLQSVoiceGetCOLP . . . 1956
9.60.2.53 unpack_voice_SLQSVoiceGetCOLR . . . 1956
9.60.2.54 unpack_voice_SLQSVoiceGetConfig . . . 1956
9.60.2.55 unpack_voice_SLQSVoiceIndicationRegister . . . 1957
9.60.2.56 unpack_voice_SLQSVoiceManageCalls . . . 1957
9.60.2.57 unpack_voice_SLQSVoiceOrigUSSDNoWait . . . 1957
9.60.2.58 unpack_voice_SLQSVoiceSendFlash . . . 1958
9.60.2.59 unpack_voice_SLQSVoiceSetCallBarringPassword . . . 1958
9.60.2.60 unpack_voice_SLQSVoiceSetConfig . . . 1958
9.60.2.61 unpack_voice_SLQSVoiceSetPreferredPrivacy . . . 1959
9.60.2.62 unpack_voice_SLQSVoiceSetSUPSService . . . 1959
9.60.2.63 unpack_voice_SLQSVoiceStartContDTMF . . . 1959
9.60.2.64 unpack_voice_SLQSVoiceStopContDTMF . . . 1960
9.61 wds.h File Reference . . . 1960
9.61.1 Macro Definition Documentation . . . 1966
9.61.1.1 BYT_STAT_STAT_MASK . . . 1966
9.61.1.2 IPV6_ADDRESS_ARRAY_SIZE . . . 1966
9.61.1.3 MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE . . . 1966
9.61.1.4 PACK_WDS_IPV4 . . . 1966
9.61.1.5 PACK_WDS_IPV6 . . . 1966
9.61.1.6 WDS_DHCP_MAX_NUM_OPTIONS . . . 1966
9.61.1.7 WDS_DHCP_OPTION_DATA_BUF_SIZE . . . 1966

9.61.2 Typedef Documentation . . . 1966
9.61.2.1 UnpackQmiProfileInfo . . . 1966
9.61.3 Function Documentation . . . 1966
9.61.3.1 pack_wds_DHCPv4ClientLeaseChange . . . 1966
9.61.3.2 pack_wds_GetAutoconnect . . . 1966
9.61.3.3 pack_wds_GetByteTotals . . . 1967
9.61.3.4 pack_wds_GetConnectionRate . . . 1967
9.61.3.5 pack_wds_GetDataBearerTechnology . . . 1968
9.61.3.6 pack_wds_GetDefaultProfile . . . 1968
9.61.3.7 pack_wds_GetDefaultProfileNum . . . 1968
9.61.3.8 pack_wds_GetDormancyState . . . 1969
9.61.3.9 pack_wds_GetLastMobileIPError . . . 1969
9.61.3.10 pack_wds_GetMobileIP . . . 1970
9.61.3.11 pack_wds_GetMobileIPProfile . . . 1970
9.61.3.12 pack_wds_GetPacketStatistics . . . 1970
9.61.3.13 pack_wds_GetPacketStatus . . . 1971
9.61.3.14 pack_wds_GetSessionDuration . . . 1971
9.61.3.15 pack_wds_GetSessionState . . . 1972
9.61.3.16 pack_wds_RMSetTransferStatistics . . . 1972
9.61.3.17 pack_wds_SetAutoconnect . . . 1972
9.61.3.18 pack_wds_SetDefaultProfile . . . 1973
9.61.3.19 pack_wds_SetDefaultProfileNum . . . 1973
9.61.3.20 pack_wds_SetMobileIP . . . 1973
9.61.3.21 pack_wds_SetMobileIPParameters . . . 1974
9.61.3.22 pack_wds_SetMobileIPProfile . . . 1974
9.61.3.23 pack_wds_SLQSCreateProfile . . . 1975
9.61.3.24 pack_wds_SLQSDeleteProfile . . . 1975
9.61.3.25 pack_wds_SLQSGet3GPPConfigItem . . . 1976
9.61.3.26 pack_wds_SLQSGetCurrDataSystemStat . . . 1976
9.61.3.27 pack_wds_SLQSGetCurrentChannelRate . . . 1976
9.61.3.28 pack_wds_SLQSGetDataBearerTechnology . . . 1977
9.61.3.29 pack_wds_SLQSGetDUNCallInfo . . . 1977
9.61.3.30 pack_wds_SLQSGetProfileSettings . . . 1978
9.61.3.31 pack_wds_SLQSGetRuntimeSettings . . . 1978
9.61.3.32 pack_wds_SLQSModifyProfile . . . 1979
9.61.3.33 pack_wds_SLQSResetPacketStatics . . . 1979
9.61.3.34 pack_wds_SLQSSet3GPPConfigItem . . . 1979
9.61.3.35 pack_wds_SLQSSetIPFamilyPreference . . . 1980
9.61.3.36 pack_wds_SLQSSetWdsEventCallback . . . 1980
9.61.3.37 pack_wds_SLQSSGetDHCPv4ClientConfig . . . 1981

9.61.3.38 pack_wds_SLQSSGetLoopback . . . 1981
9.61.3.39 pack_wds_SLQSSSetDHCPv4ClientConfig . . . 1981
9.61.3.40 pack_wds_SLQSSSetLoopback . . . 1982
9.61.3.41 pack_wds_SLQSStartDataSession . . . 1982
9.61.3.42 pack_wds_SLQSStopDataSession . . . 1983
9.61.3.43 pack_wds_SLQSWdsGoActive . . . 1983
9.61.3.44 pack_wds_SLQSWdsGoDormant . . . 1984
9.61.3.45 pack_wds_SLQSWdsSetEventReport . . . 1984
9.61.3.46 pack_wds_SLQSWdsSwiPDPRuntimeSettings . . . 1984
9.61.3.47 unpack_wds_DHCPv4ClientLease_ind . . . 1985
9.61.3.48 unpack_wds_DHCPv4ClientLeaseChange . . . 1985
9.61.3.49 unpack_wds_GetAutoconnect . . . 1985
9.61.3.50 unpack_wds_GetByteTotals . . . 1986
9.61.3.51 unpack_wds_GetConnectionRate . . . 1986
9.61.3.52 unpack_wds_GetDataBearerTechnology . . . 1986
9.61.3.53 unpack_wds_GetDefaultProfile . . . 1987
9.61.3.54 unpack_wds_GetDefaultProfileNum . . . 1987
9.61.3.55 unpack_wds_GetDormancyState . . . 1987
9.61.3.56 unpack_wds_GetLastMobileIPError . . . 1988
9.61.3.57 unpack_wds_GetMobileIP . . . 1988
9.61.3.58 unpack_wds_GetMobileIPProfile . . . 1988
9.61.3.59 unpack_wds_GetPacketStatistics . . . 1989
9.61.3.60 unpack_wds_GetPacketStatus . . . 1989
9.61.3.61 unpack_wds_GetSessionDuration . . . 1990
9.61.3.62 unpack_wds_GetSessionState . . . 1990
9.61.3.63 unpack_wds_RMSetTransferStatistics . . . 1990
9.61.3.64 unpack_wds_RMTransferStatistics_ind . . . 1991
9.61.3.65 unpack_wds_SetAutoconnect . . . 1991
9.61.3.66 unpack_wds_SetDefaultProfile . . . 1991
9.61.3.67 unpack_wds_SetDefaultProfileNum . . . 1992
9.61.3.68 unpack_wds_SetMobileIP . . . 1992
9.61.3.69 unpack_wds_SetMobileIPParameters . . . 1992
9.61.3.70 unpack_wds_SetMobileIPProfile . . . 1992
9.61.3.71 unpack_wds_SLQSCreateProfile . . . 1993
9.61.3.72 unpack_wds_SLQSDeleteProfile . . . 1993
9.61.3.73 unpack_wds_SLQSGet3GPPConfigItem . . . 1993
9.61.3.74 unpack_wds_SLQSGetCurrDataSystemStat . . . 1994
9.61.3.75 unpack_wds_SLQSGetCurrentChannelRate . . . 1994
9.61.3.76 unpack_wds_SLQSGetDataBearerTechnology . . . 1995
9.61.3.77 unpack_wds_SLQSGetDUNCallInfo . . . 1995

9.61.3.78 unpack_wds_SLQSGetProfileSettings . . . 1995
9.61.3.79 unpack_wds_SLQSGetRuntimeSettings . . . 1996
9.61.3.80 unpack_wds_SLQSModifyProfile . . . 1996
9.61.3.81 unpack_wds_SLQSResetPacketStatics . . . 1996
9.61.3.82 unpack_wds_SLQSSet3GPPConfigItem . . . 1996
9.61.3.83 unpack_wds_SLQSSetIPFamilyPreference . . . 1997
9.61.3.84 unpack_wds_SLQSSetPacketSrvStatusCallback . . . 1997
9.61.3.85 unpack_wds_SLQSSetWdsEventCallback . . . 1997
9.61.3.86 unpack_wds_SLQSSetWdsEventCallback_ind . . . 1998
9.61.3.87 unpack_wds_SLQSSGetDHCPv4ClientConfig . . . 1998
9.61.3.88 unpack_wds_SLQSSGetLoopback . . . 1998
9.61.3.89 unpack_wds_SLQSSSetDHCPv4ClientConfig . . . 1999
9.61.3.90 unpack_wds_SLQSSSetLoopback . . . 1999
9.61.3.91 unpack_wds_SLQSStartDataSession . . . 1999
9.61.3.92 unpack_wds_SLQSStopDataSession . . . 2000
9.61.3.93 unpack_wds_SLQSWdsGoActive . . . 2000
9.61.3.94 unpack_wds_SLQSWdsGoDormant . . . 2000
9.61.3.95 unpack_wds_SLQSWdsSetEventReport . . . 2001
9.61.3.96 unpack_wds_SLQSWdsSwiPDPRuntimeSettings . . . 2001

Index **2002**

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) . . . 37
Wireless Data Service (WDS) . . . 38
Device Management Service (DMS) . . . 39
Network Access Service (NAS) . . . 40
CallBacK registration (CBK) . . . 41
Short Message Service (SMS) . . . 42
Position Determination Service (PDS) . . . 43
Card Application Toolkit (CAT) . . . 44
Remote Management Service (RMS) . . . 45
Firmware Management Service (FMS) . . . 46
Open Mobile Alliance Service (OMA) . . . 47
Specific Absorption Rate (SAR) . . . 48
SWI Open Mobile Alliance Service (SWIOMA) . . . 49
Voice Service (VOICE) . . . 50
Non-service specific APIs (SWI) . . . 51
User Identity Module Service (UIM) . . . 52
Audio Service (AUDIO) . . . 53
Quality of Service (QOS) . . . 54
IMS Service (IMS) . . . 55
SWI Audio Service(SWIAUDIO) . . . 56
Location Service(LOC) . . . 57
Thermal Mitigation Device(TMD) . . . 58
AVMS (SWIAVMS) . . . 59

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Tables 61

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

_getIndicationRegResp . . . 63
_GetProfileSettingIn . . . 64
_GetProfileSettingOut . . . 64
_getResetInfoNotification . . . 65
_getTransLayerInfoResp . . . 66
_getTransNWRegInfoResp . . . 66
_MitigationDevInfo . . . 67
_modemTempNotification . . . 67
_packetSrvStatus . . . 68
_qaQmi3GPP2BroadcastCfgInfo . . . 70
_qaQmi3GPPBroadcastCfgInfo . . . 71
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp . . . 71
_QmiSwiAvmsEventReportConfigResp . . . 73
_QmiSwiAvmsEventReportConnectionRequestResp . . . 74
_QmiSwiAvmsEventReportDataSessionStatusResp . . . 75
_QmiSwiAvmsEventReportHTTPStatusResp . . . 75
_QmiSwiAvmsEventReportNotificationResp . . . 77
_QmiSwiAvmsEventReportPackageIDResp . . . 78
_QmiSwiAvmsEventReportRegStatusResp . . . 78
_QmiSwiAvmsEventReportSessionTypeResp . . . 78
_QmiSwiAvmsEventReportWAMSParamChangeResp . . . 79
_QmiSwiAvmsSetSettingsAPNInfo . . . 79
_QmiSwiAvmsSetSettingsConnectionRetryTimers . . . 80
_QmiSwiAvmsSetSettingsPeriodInfo . . . 80
_setIndicationRegReq . . . 80
_slqs3GPPConfigItem . . . 81
_SLQSAVMSSetSettingsReqPara . . . 83
_SLQSAVMSSettings . . . 85
_SLQSAVMSSettingsAPNInfo . . . 86
_SLQSAVMSSettingsConnectionRetryTimers . . . 86
_SLQSAVMSSettingsPeriodsInfo . . . 86
SlqsNas3GppNetworkRAT . . . 87
_slqsNetworkScanInfo . . . 87
_SLQSOMADMSessionInfo . . . 89
_SLQSOMADMSettings . . . 91
_SLQSOMADMSettingsReqParams . . . 93
_SLQSOMADMSettingsReqParams3 . . . 94
_SLQSSwiGetHostDevInfoParams . . . 95

_SLQSSwiGetOSInfoParams . . . 96
_SLQSSwiGetSerialNoExtParams . . . 97
_SLQSSwiSetHostDevInfoParams . . . 97
_SLQSSwiSetOSInfoParams . . . 98
_SwiAvmsEventInd . . . 99
_sysSelectPrefInfo . . . 100
_sysSelectPrefParams . . . 105
_transLayerinfo . . . 111
_transLayerInfoNotification . . . 111
_transNWRegInfoNotification . . . 112
accelAcceptReady_s . . . 113
accelTempAcceptReady_s . . . 113
acqOrderPref . . . 114
ActPilotPNElement . . . 115
AddCDMASysInfo . . . 115
AddSysInfo . . . 116
airTimer . . . 116
allCallsAlphaIDInfo . . . 117
allCallsDiagInfo . . . 117
allCallsUUSInfo . . . 118
alphaIDInfo . . . 118
altitudeSrcInfo . . . 119
altSrcInfo_t . . . 120
appStats . . . 121
appStatus . . . 124
arrAlertingPattern . . . 127
arrAlertingType . . . 128
arrAlphaID . . . 129
arrCalledPartyNum . . . 129
arrCallEndReason . . . 130
arrCallInfo . . . 131
arrConnectPartyNum . . . 131
arrDiagInfo . . . 132
arrRedirPartyNum . . . 132
arrRemotePartyName . . . 133
arrRemotePartyNum . . . 133
arrSvcOption . . . 134
arrUUSInfo . . . 135
authenticateResult . . . 135
authenticationData . . . 136
BandCapabilityResp . . . 137
BdsSV . . . 140
BdsSVInfo . . . 140
BroadcastConfig . . . 141
burstDTMFInfo . . . 142
CallBarringSysInfo . . . 142
callBarStatus . . . 143
calledPartyInfo . . . 144
calledPartySubAdd . . . 146
callerIDInfo . . . 147
callFwdTypeAndPlan . . . 147
callFWExtInfo . . . 148
callFWInfo . . . 150
callInfo . . . 151
callingPartyInfo . . . 153
cardResult . . . 154
cardStatus . . . 155
CarrierImage_t . . . 156

CatAlPhaIdentifierTlv . . . 157
CatCommonEventTlv . . . 158
CatEndProactiveSessionTlv . . . 158
CATEventDataType . . . 159
CatEventIDDataTlv . . . 159
CatEventListTlv . . . 159
CatRefreshTlv . . . 160
ccSUPSType . . . 160
CDMABroadcastConfig . . . 161
CDMAChannel . . . 162
CDMAECIOThresh . . . 163
CDMAInfo . . . 163
cdmaMsgDecodingParams . . . 164
cdmaMsgEncodingParams . . . 167
CDMARSSIThresh . . . 169
CDMASSInfo . . . 169
cdmaSSInfo . . . 170
CDMASysInfo . . . 170
CDMASysInfoExt . . . 174
CellDb . . . 174
cellParams . . . 175
changeUIMPIN . . . 176
channelRate . . . 177
ChannelRate . . . 177
CLIPResp . . . 178
CLIRResp . . . 179
ClkInfo . . . 180
CNAPResp . . . 181
COLPResp . . . 181
COLRResp . . . 182
CommInfo . . . 183
ConnectionStatus . . . 184
connectionStatus . . . 185
connectNumInfo . . . 185
CrashInfo . . . 187
crashInfoParams . . . 188
CrashInfoParams . . . 189
crashInformation . . . 189
CreateProfileIn . . . 190
CreateProfileOut . . . 191
CsgId . . . 192
CSGID . . . 192
CUGInfo . . . 193
curAMRConfig . . . 194
CurrDataSysStat . . . 195
currentCatEvent . . . 196
CurrentImgList . . . 196
currentPLMN . . . 197
CurrImageInfo . . . 198
CurrNetworkInfo . . . 199
currNetworkInfo . . . 201
custFeaturesInfo . . . 201
custFeaturesSetting . . . 204
custSettingInfo . . . 205
custSettingList . . . 206
dataBearers . . . 207
DataBearerTech . . . 207
DataBearerTechExt . . . 209

dataBearerTechnology . . . 210
dataRate . . . 211
dataSrvCapabilities . . . 212
DataStatusDetail . . . 213
DataULongLongTlv . . . 215
DataULongTlv . . . 215
DcsUsbPortNames . . . 215
delAssistDataStatus . . . 215
depersonalizationInformation . . . 216
detailSvcInfo . . . 217
DeviceConfigDetail . . . 219
DHCPOption . . . 220
DHCPOptionList . . . 220
diagInfo . . . 221
dirNum . . . 221
dms_ActivationStatusTlv . . . 222
dms_OperatingModeTlv . . . 223
dmsCurrentPRLInfo . . . 223
DMScustSettingInfo . . . 224
DMScustSettingList . . . 225
DMSgetCustomFeatureV2 . . . 225
DMSgetCustomInput . . . 226
dmsIndicationRegisterReq . . . 226
dmsSwiGetResetInfo . . . 227
Domain . . . 228
DomainNameList . . . 228
DRCParams . . . 229
DTMFInfo . . . 229
DTMFLengths . . . 230
DUNCallInfoInd . . . 231
dunchannelRate . . . 232
ecioListElement . . . 232
ECIOThresh . . . 233
ECTNum . . . 234
encryptedPIN1 . . . 234
eriDataparams . . . 235
ERIFileparams . . . 235
errorRateListElement . . . 236
eTWSPLMNInfoTlv . . . 237
extDispRecInfo . . . 237
FactorySequenceNumber . . . 238
fileAttributes . . . 238
fileInfo . . . 241
FirmwareUpdatStat . . . 242
FMSImageElement . . . 244
FMSImageIdElement . . . 244
FMSImageIDEntries . . . 245
FMSImageList . . . 246
FMSPrefImageList . . . 247
fwinfo_s . . . 247
GERANInfo . . . 248
geranInstInfo . . . 250
getAllCallInformation . . . 251
getAllCallRmtPtyName . . . 251
getAllCallRmtPtyNum . . . 252
GetAudioPathConfigReq . . . 252
GetAudioPathConfigResp . . . 253
GetAudioProfileReq . . . 255

GetAudioProfileResp . . . 255
GetAudioVolTLBConfigReq . . . 257
GetAudioVolTLBConfigResp . . . 257
getCallFWExtInfo . . . 258
getCallFWInfo . . . 258
getCustomFeatureV2 . . . 259
getCustomInput . . . 260
getDUNCallInfoReq . . . 260
getDUNCallInfoResp . . . 262
getDyingGaspCfg . . . 264
getDyingGaspStatistics . . . 265
GetErrRateResp . . . 266
GetHRPDStatsResp . . . 266
GetIMSSMSConfigParams . . . 267
GetIMSUserConfigParams . . . 268
GetIMSVoIPConfigResp . . . 269
GetInstIDResp . . . 271
GetM2MAudioProfileReq . . . 272
GetM2MAudioProfileResp . . . 272
GetM2MAudioVolumeReq . . . 273
GetM2MAudioVolumeResp . . . 274
GetM2MAVMuteReq . . . 274
GetM2MAVMuteResp . . . 275
GetM2MSpkrGainReq . . . 275
GetM2MSpkrGainResp . . . 276
getMsgWaitingInfo . . . 276
GetNetworkTimeResp . . . 277
GetRegMgrConfigParams . . . 277
GetSessionIDResp . . . 278
GetSIPConfigResp . . . 278
GnssData . . . 280
gnssSvInfoNotification . . . 281
GPRSQoS . . . 282
GPRSRequestedQoS . . . 283
GPSStateInfo . . . 284
gpsTime_s . . . 288
gsmCellInfo . . . 288
GSMRSSIThresh . . . 290
GSMSrvStatusInfo . . . 290
GSMSysInfo . . . 291
gyroAcceptReady_s . . . 294
gyroTempAcceptReady_s . . . 295
HDRECIOThresh . . . 295
HDRIOThresh . . . 296
HDRPersonalityInd . . . 296
HDRPersonalityResp . . . 297
HDRProtSubtypResp . . . 297
HDRRSSIThresh . . . 298
HDRSINRThresh . . . 299
HDRSINRThreshold . . . 299
HDRSSInfo . . . 300
hdrSSInfo . . . 301
HDRSysInfo . . . 302
homeSIDNID . . . 304
hotSwapStatus . . . 304
image_info_t . . . 305
ImageElement . . . 305
ImageIdElement . . . 306

ImageIDEntries . . . 307
ImageList . . . 308
IMSAIndRegisterInfo . . . 308
imsaPdpStatusInfo . . . 309
imsaRatStatusInfo . . . 310
IMSARegistrationStatus . . . 311
imsaRegStatusInfo . . . 312
IMSAServiceStatus . . . 312
IMSASupportedFieldsResp . . . 314
IMSASupportedMsgInfo . . . 315
imsaSvcStatusInfo . . . 315
imsCfgIndRegisterInfo . . . 316
imsRegMgrConfigInfo . . . 318
imsSIPConfigInfo . . . 318
imsSMSConfigInfo . . . 319
imsUserConfigInfo . . . 320
imsVoIPConfigInfo . . . 321
IndFieldsList . . . 323
infoInterFreq . . . 324
IOThresh . . . 325
IPv4Addr . . . 325
IPv6Addr . . . 326
IPV6AddressInfo . . . 326
ipv6AddressInfo . . . 327
IPV6GWAddressInfo . . . 327
IPv6TrafCls . . . 328
LibPackGPRSRequestedQoS . . . 328
LibpackProfile3GPP . . . 329
LibpackProfile3GPP2 . . . 335
LibPackprofile_3GPP . . . 340
LibPackprofile_3GPP2 . . . 346
LibPackQosClassID . . . 351
LibPackTFTIDParams . . . 352
LibPackUMTSQoS . . . 354
LibPackUMTSReqQoSSigInd . . . 356
lineCtrlInfo . . . 357
loc_BdsSV . . . 358
loc_BdsSVInfo . . . 358
loc_CellDb . . . 359
loc_ClkInfo . . . 359
loc_GnssData . . . 361
loc_gpsTime . . . 362
loc_LocApplicationInfo . . . 363
loc_precisionDilution . . . 364
loc_satelliteInfo . . . 365
loc_sensorDataUsage . . . 367
loc_SV . . . 368
loc_SVInfo . . . 369
loc_svUsedforFix . . . 369
LocApplicationInfo . . . 370
LocDelAssDataReq . . . 371
LOCEventRegisterReqResp . . . 372
LOCExtPowerStateReqResp . . . 374
LocInjectPositionReq . . . 374
LocInjectSensorDataReq . . . 378
LocSetCradleMountReq . . . 379
LOCStartReq . . . 380
LOCStopReq . . . 382

LteCQIParm . . . 382
lteEARFCN . . . 383
lteGsmCellInfo . . . 384
LTEInfo . . . 385
LTEInfoInterfreq . . . 387
LTEInfoIntrafreq . . . 387
LTEInfoNeighboringGSM . . . 390
LTEInfoNeighboringWCDMA . . . 390
LteNasReleaseInfo_s . . . 391
ltePCI . . . 392
lteRsrpinformation . . . 392
LTERSRPThresh . . . 393
LTERSRQThresh . . . 393
LTERSSIThresh . . . 394
LteSccRxInfoResp . . . 394
LTESigRptCfg . . . 395
LTESigRptConfig . . . 395
lteSnrinformation . . . 396
LTESNRThresh . . . 397
LTESNRThreshold . . . 397
LTESSInfo . . . 398
lteSSInfo . . . 399
LTESysInfo . . . 399
lteWcdmaCellInfo . . . 402
messageModeTlv . . . 403
messageWaitingInfoContent . . . 403
minBasedIMSI . . . 404
mitigationDevList . . . 405
MNRInfo . . . 405
ModifyProfileIn . . . 406
ModifyProfileOut . . . 407
msgWaitingInfo . . . 407
namName . . . 408
nas_acqOrderPref . . . 408
nas_ActPilotPNElement . . . 409
nas_AddCDMASysInfo . . . 409
nas_AddSysInfo . . . 410
nas_CallBarringSysInfo . . . 411
nas_callBarStatus . . . 411
nas_CDMAChannel . . . 412
nas_CDMAECIOThresh . . . 413
nas_CDMAInfo . . . 413
nas_CDMARSSIThresh . . . 414
nas_CDMASysInfo . . . 415
nas_CDMASysInfoExt . . . 419
nas_cellParams . . . 419
nas_CommInfo . . . 420
nas_CSGID . . . 422
nas_currentPLMN . . . 423
nas_dataSrvCapabilities . . . 424
nas_DataStatusDetail . . . 424
nas_detailSvcInfo . . . 427
nas_DeviceConfigDetail . . . 428
nas_dirNum . . . 429
nas_DRCParams . . . 429
nas_ecioListElement . . . 430
nas_ECIOThresh . . . 431
nas_errorRateListElement . . . 431

nas_GERANInfo . . . 432
nas_geranInstInfo . . . 434
nas_gsmCellInfo . . . 435
nas_GSMRSSIThresh . . . 436
nas_GSMSrvStatusInfo . . . 436
nas_GSMSysInfo . . . 437
nas_HDRECIOThresh . . . 440
nas_HDRIOThresh . . . 440
nas_HDRRSSIThresh . . . 441
nas_HDRSINRThresh . . . 442
nas_HDRSINRThreshold . . . 442
nas_HDRSysInfo . . . 443
nas_homeSIDNID . . . 445
nas_infoInterFreq . . . 446
nas_IOThresh . . . 447
nas_lteEARFCN . . . 447
nas_lteGsmCellInfo . . . 448
nas_LTEInfo . . . 449
nas_LTEInfoInterfreq . . . 452
nas_LTEInfoIntrafreq . . . 452
nas_LTEInfoNeighboringGSM . . . 454
nas_LTEInfoNeighboringWCDMA . . . 455
nas_ltePCI . . . 456
nas_lteRsrpinformation . . . 456
nas_LTERSRPThresh . . . 457
nas_LTERSRQThresh . . . 457
nas_LTERSSIThresh . . . 458
nas_LTESigRptCfg . . . 458
nas_LTESigRptConfig . . . 459
nas_lteSnrinformation . . . 460
nas_LTESNRThresh . . . 461
nas_LTESNRThreshold . . . 461
nas_LTESysInfo . . . 462
nas_lteWcdmaCellInfo . . . 464
nas_minBasedIMSI . . . 465
nas_MNRInfo . . . 466
nas_namName . . . 467
nas_netSelectionPref . . . 467
nas_NetworkStat1x . . . 468
nas_NetworkStatEVDO . . . 470
nas_nmrCellInfo . . . 472
nas_operatorNameString . . . 473
nas_OperatorPLMNData . . . 473
nas_operatorPLMNList . . . 474
nas_PhyCaAggPcellInfo . . . 475
nas_PhyCaAggScellDlBw . . . 476
nas_PhyCaAggScellIndex . . . 476
nas_PhyCaAggScellIndType . . . 477
nas_PhyCaAggScellInfo . . . 477
nas_PilotSetData . . . 480
nas_PilotSetParams . . . 480
nas_PLMNNetworkName . . . 481
nas_PLMNNetworkNameData . . . 482
nas_protocolSubtypeElement . . . 483
nas_qaQmi3Gpp2TimeZone . . . 485
nas_QmiNas3GppNetworkInfo . . . 485
nas_QmiNas3GppNetworkRAT . . . 486
nas_QmisNasPcsDigit . . . 487

nas_RejectReasonTlv . . . 488
nas_RFInfoTlv . . . 488
nas_roamIndList . . . 489
nas_RSRPThresh . . . 489
nas_rsrqInformation . . . 490
nas_RSRQThresh . . . 491
nas_RSSIThresh . . . 491
nas_rxInfo . . . 492
nas_RxSigInfo . . . 493
nas_rxSignalStrengthListElement . . . 494
nas_SccRxInfo . . . 495
nas_serviceProviderName . . . 496
nas_servSystem . . . 496
nas_sidNid . . . 498
nas_SignalStrengthTlv . . . 498
nas_SLQSSignalStrengthsIndReq . . . 499
nas_SLQSSignalStrengthsInformation . . . 500
nas_SLQSSignalStrengthsTlv . . . 501
nas_SrvStatusInfo . . . 501
nas_sysInfoCommon . . . 502
nas_TDSCDMAECIOThresh . . . 504
nas_TDSCDMARSCPThresh . . . 505
nas_TDSCDMARSSIThresh . . . 505
nas_TDSCDMASINRCONFThresh . . . 506
nas_TDSCDMASINRThresh . . . 506
nas_timeInfo . . . 507
nas_trueIMSI . . . 508
nas_txInfo . . . 509
nas_UMTSInfo . . . 510
nas_UMTSinstInfo . . . 512
nas_umtsLTENbrCell . . . 512
nas_UniversalTime . . . 513
nas_wcdmaCellInfo . . . 514
nas_WCDMAECIOThresh . . . 515
nas_WCDMAInfoLTENeighborCell . . . 516
nas_WCDMARSSIThresh . . . 517
nas_WCDMASysInfo . . . 517
nas_wcdmaUARFCN . . . 521
NASBandPreferenceTlv . . . 521
nasCellLocationInfoResp . . . 521
NASEmergencyModeTlv . . . 523
nasGet3GPP2SubscriptionInfoReq . . . 523
nasGet3GPP2SubscriptionInfoResp . . . 523
nasGetHDRColorCodeResp . . . 524
nasGetLTECphyCa . . . 525
NasGetLTECphyCaInfo . . . 525
nasGetLTECphyCaResp . . . 526
nasGetSigInfoResp . . . 526
nasGetSysInfoResp . . . 527
nasGetTxRxInfoReq . . . 530
nasGetTxRxInfoResp . . . 530
NASGWAcqOrderPrefTlv . . . 531
nasIndicationRegisterReq . . . 531
nasInitNetworkReg . . . 534
NASLTEBandPreferenceTlv . . . 535
NASLteNasReleaseInfoTlv . . . 536
NASModePreferenceTlv . . . 536
NASNetSelPreferenceTlv . . . 536

nasNetworkReject . . . 537
nasNetworkTime . . . 539
nasOperatorNameResp . . . 540
NASOTAMessageTlv . . . 541
NASPhyCaAggPcellInfo . . . 541
NASPhyCaAggScellDlBw . . . 542
NASPhyCaAggScellIndex . . . 543
NASPhyCaAggScellIndType . . . 543
NASPhyCaAggScellInfo . . . 544
nasPLMNNameReq . . . 545
nasPLMNNameResp . . . 546
NASPRLPreferenceTlv . . . 548
NASQmiCbkNasSwiOTAMessageInd . . . 548
NASQmiCbkNasSystemSelPrefInd . . . 549
NASRoamPreferenceTlv . . . 549
NASServDomainPrefTlv . . . 550
NASServingSystemInfo . . . 550
nasSigInfo . . . 551
nasSwiGetChannelLockResp . . . 552
NasSwiIndReg . . . 553
nasSwiSetChannelLockReq . . . 554
nasSysInfo . . . 555
NASTimeInfoTlv . . . 558
nasTimers . . . 558
netSelectionPref . . . 559
NetStats . . . 560
NetworkDebugResp . . . 561
NetworkStat1x . . . 562
NetworkStatEVDO . . . 564
newMTMessageTlv . . . 565
newPwdData . . . 566
nmrCellInfo . . . 566
NSSAudioCtrl . . . 568
NWProfile . . . 568
omaDmConfigTlv . . . 569
omaDmConfigTlvExt . . . 570
omaDmFotaTlv . . . 571
omaDmFotaTlvExt . . . 573
omaDmNotificationsTlv . . . 575
operatorNameString . . . 575
OperatorPLMNData . . . 576
operatorPLMNList . . . 577
pack_dms_ActivateAutomatic_t . . . 577
pack_dms_GetCustFeaturesV2_t . . . 578
pack_dms_ResetToFactoryDefaults_t . . . 578
pack_dms_SetActivationStatusCallback_t . . . 579
pack_dms_SetCrashAction_t . . . 579
pack_dms_SetCustFeature_t . . . 579
pack_dms_SetCustFeaturesV2_t . . . 580
pack_dms_SetEventReport_t . . . 581
pack_dms_SetPower_t . . . 581
pack_dms_SetUSBComp_t . . . 581
pack_dms_SLQSDmsSwiIndicationRegister_t . . . 582
pack_dms_SLQSSwiGetCrashInfo_t . . . 582
pack_dms_SLQSSwiSetDyingGaspCfg_t . . . 582
pack_dms_SLQSSwiSetHostDevInfo_t . . . 583
pack_dms_SLQSSwiSetOSInfo_t . . . 584
pack_dms_UIMChangePIN_t . . . 584

pack_dms_UIMGetControlKeyStatus_t . . . 585
pack_dms_UIMGetICCID_t . . . 585
pack_dms_UIMSetControlKeyProtection_t . . . 586
pack_dms_UIMSetPINProtection_t . . . 586
pack_dms_UIMUnblockControlKey_t . . . 587
pack_dms_UIMUnblockPIN_t . . . 588
pack_dms_UIMVerifyPIN_t . . . 588
pack_fms_GetImagesPreference_t . . . 589
pack_fms_GetStoredImages_t . . . 589
pack_fms_SetImagesPreference_t . . . 590
pack_loc_Delete_Assist_Data_t . . . 590
pack_loc_EventRegister_t . . . 591
pack_loc_SetExtPowerState_t . . . 594
pack_loc_SetOperationMode_t . . . 594
pack_loc_SLQSLOCGetBestAvailPos_t . . . 595
pack_loc_SLQSLOCInjectPosition_t . . . 595
pack_loc_SLQSLOCInjectSensorData_t . . . 600
pack_loc_SLQSLOCInjectUTCTime_t . . . 602
pack_loc_SLQSLOCSetCradleMountConfig_t . . . 603
pack_loc_Start_t . . . 603
pack_loc_Stop_t . . . 605
pack_nas_InitiateDomainAttach_t . . . 605
pack_nas_SetACCOLC_t . . . 606
pack_nas_SetCDMANetworkParameters_t . . . 606
pack_nas_SetNetworkPreference_t . . . 608
pack_nas_SLQSConfigSigInfo_t . . . 609
pack_nas_SLQSGetPLMNName_t . . . 610
pack_nas_SLQSInitiateNetworkRegistration_t . . . 611
pack_nas_SLQSNasConfigSigInfo2_t . . . 612
pack_nas_SLQSNasGet3GPP2Subscription_t . . . 616
pack_nas_SLQSNasGetTxRxInfo_t . . . 617
pack_nas_SLQSNasIndicationRegisterExt_t . . . 617
pack_nas_SLQSNasSwiIndicationRegister_t . . . 620
pack_nas_SLQSNASSwiSetChannelLock_t . . . 621
pack_nas_SLQSSetBandPreference_t . . . 622
pack_nas_SLQSSetSignalStrengthsCallback_t . . . 624
pack_nas_SLQSSetSysSelectionPref_t . . . 624
pack_nas_SLQSSwiPSDetach_t . . . 629
pack_pds_SetPDSDefaults_t . . . 629
pack_pds_SetPDSState_t . . . 630
pack_pds_SetPortAutomaticTracking_t . . . 630
pack_qmi_t . . . 631
pack_qos_SLQSQosSwiReadApnExtraParams_t . . . 631
pack_qos_SLQSQosSwiReadDataStats_t . . . 632
pack_qos_SLQSSetQosEventCallback_t . . . 632
pack_rms_SetSMSWake_t . . . 633
pack_sar_SLQSSetRfSarState_t . . . 633
pack_sms_SaveSMS_t . . . 634
pack_sms_SendSMS_t . . . 635
pack_sms_SetNewSMSCallback_t . . . 635
pack_sms_SetSMSCAddress_t . . . 636
pack_sms_SLQSDeleteSMS_t . . . 636
pack_sms_SLQSGetSMS_t . . . 637
pack_sms_SLQSGetSmsBroadcastConfig_t . . . 638
pack_sms_SLQSGetSMSList_t . . . 638
pack_sms_SLQSModifySMSStatus_t . . . 639
pack_sms_SLQSSendAsyncSMS_t . . . 640
pack_sms_SLQSSetIndicationRegister_t . . . 640

pack_sms_SLQSSetSmsBroadcastActivation_t . . . 641
pack_sms_SLQSSetSmsBroadcastConfig_t . . . 641
pack_sms_SLQSSetSmsStorage_t . . . 642
pack_sms_SLQSSmsGetMaxStorageSize_t . . . 642
pack_sms_SLQSSmsSetRoutes_t . . . 643
pack_swiloc_SwiLocSetAutoStart_t . . . 643
pack_swioma_SLQSOMADMCancelSession_t . . . 645
pack_swioma_SLQSOMADMGetSessionInfo_t . . . 645
pack_swioma_SLQSOMADMSendSelection_t . . . 646
pack_swioma_SLQSOMADMSetSettings_t . . . 646
pack_swioma_SLQSOMADMStartSession_t . . . 647
pack_uim_ChangePin_t . . . 648
pack_uim_ReadTransparent_t . . . 649
pack_uim_SetPinProtection_t . . . 650
pack_uim_SLQSUIMAuthenticate_t . . . 651
pack_uim_SLQSUIMDepersonalization_t . . . 652
pack_uim_SLQSUIMEventRegister_t . . . 652
pack_uim_SLQSUIMGetConfiguration_t . . . 652
pack_uim_SLQSUIMGetFileAttributes_t . . . 653
pack_uim_SLQSUIMPowerDown_t . . . 654
pack_uim_SLQSUIMPowerUp_t . . . 654
pack_uim_SLQSUIMRefreshComplete_t . . . 655
pack_uim_SLQSUIMRefreshGetLastEvent_t . . . 655
pack_uim_SLQSUIMRefreshOK_t . . . 656
pack_uim_SLQSUIMRefreshRegister_t . . . 656
pack_uim_SLQSUIMSwitchSlot_t . . . 657
pack_uim_UnblockPin_t . . . 658
pack_uim_VerifyPin_t . . . 659
pack_voice_AnswerUSSD_t . . . 660
pack_voice_OriginateUSSD_t . . . 660
pack_voice_SLQSOriginateUSSD_t . . . 660
pack_voice_SLQSVoiceALSSelectLine_t . . . 661
pack_voice_SLQSVoiceALSSetLineSwitching_t . . . 661
pack_voice_SLQSVoiceAnswerCall_t . . . 662
pack_voice_SLQSVoiceBindSubscription_t . . . 662
pack_voice_SLQSVoiceBurstDTMF_t . . . 663
pack_voice_SLQSVoiceDialCall_t . . . 663
pack_voice_SLQSVoiceEndCall_t . . . 665
pack_voice_SLQSVoiceGetCallBarring_t . . . 665
pack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 666
pack_voice_SLQSVoiceGetCallInfo_t . . . 667
pack_voice_SLQSVoiceGetCallWaiting_t . . . 667
pack_voice_SLQSVoiceGetConfig_t . . . 668
pack_voice_SLQSVoiceIndicationRegister_t . . . 670
pack_voice_SLQSVoiceManageCalls_t . . . 670
pack_voice_SLQSVoiceOrigUSSDNoWait_t . . . 671
pack_voice_SLQSVoiceSendFlash_t . . . 672
pack_voice_SLQSVoiceSetCallBarringPassword_t . . . 672
pack_voice_SLQSVoiceSetConfig_t . . . 673
pack_voice_SLQSVoiceSetPreferredPrivacy_t . . . 675
pack_voice_SLQSVoiceSetSUPSService_t . . . 675
pack_voice_SLQSVoiceStartContDTMF_t . . . 677
pack_voice_SLQSVoiceStopContDTMF_t . . . 678
pack_wds_DHCPv4ClientLeaseChange_t . . . 678
pack_wds_GetDefaultProfile_t . . . 679
pack_wds_GetDefaultProfileNum_t . . . 679
pack_wds_GetDormancyState_t . . . 679
pack_wds_GetLastMobileIPError_t . . . 679

pack_wds_GetMobileIP_t . . . 680
pack_wds_GetMobileIPProfile_t . . . 680
pack_wds_GetPacketStatistics_t . . . 680
pack_wds_GetPacketStatus_t . . . 680
pack_wds_GetSessionDuration_t . . . 681
pack_wds_RMSetTransferStatistics_t . . . 681
pack_wds_SetAutoconnect_t . . . 681
pack_wds_SetDefaultProfile_t . . . 681
pack_wds_SetDefaultProfileNum_t . . . 682
pack_wds_SetMobileIP_t . . . 683
pack_wds_SetMobileIPParameters_t . . . 683
pack_wds_SetMobileIPProfile_t . . . 685
pack_wds_SLQSCreateProfile_t . . . 686
pack_wds_SLQSDeleteProfile_t . . . 686
pack_wds_SLQSGetCurrDataSystemStat_t . . . 687
pack_wds_SLQSGetDataBearerTechnology_t . . . 687
pack_wds_SLQSGetDUNCallInfo_t . . . 687
pack_wds_SLQSGetProfileSettings_t . . . 688
pack_wds_SLQSGetRuntimeSettings_t . . . 688
pack_wds_SLQSModifyProfile_t . . . 689
pack_wds_SLQSSet3GPPConfigItem_t . . . 690
pack_wds_SLQSSetIPFamilyPreference_t . . . 692
pack_wds_SLQSSetWdsEventCallback_t . . . 692
pack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 693
pack_wds_SLQSSSetDHCPv4ClientConfig_t . . . 693
pack_wds_SLQSSSetLoopback_t . . . 694
pack_wds_SLQSStartDataSession_t . . . 694
pack_wds_SLQSStopDataSession_t . . . 695
pack_wds_SLQSWdsSetEventReport_t . . . 696
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 698
PackCreateProfileOut . . . 698
packgetDyingGaspCfg . . . 698
packgetDyingGaspStatistics . . . 699
PCMparams . . . 699
PCSCFFQDNAddress . . . 700
PCSCFFQDNAddressList . . . 700
PCSCFIPv4ServerAddressList . . . 701
PDSPositionData . . . 701
PDSPosMethodStateReq . . . 703
peerNumberInfo . . . 704
personalizationStatus . . . 706
PhyCaAggPcellInfo . . . 707
PhyCaAggScellDlBw . . . 708
PhyCaAggScellIndex . . . 708
PhyCaAggScellIndType . . . 709
PhyCaAggScellInfo . . . 709
PilotSetData . . . 712
PilotSetParams . . . 713
pktErrRate . . . 713
PlmnID . . . 714
PLMNNetworkName . . . 714
PLMNNetworkNameData . . . 715
Port . . . 717
precisionDilution_s . . . 717
PrefImageList . . . 718
prefVoiceSO . . . 718
Profile3GPP . . . 720
Profile3GPP2 . . . 726

ProfileIdentifier . . . 731
protocolSubtypeElement . . . 732
PSDetachReq . . . 733
qaQmi3Gpp2TimeZone . . . 734
qaQmiInterfaceInfo . . . 735
qaQmiServingSystemParam . . . 735
QmiCbkCatEventStatusReportInd . . . 739
QmiCbkLocBestAvailPosInd . . . 739
QmiCbkLocCradleMountInd . . . 745
QmiCbkLocEngineStateInd . . . 746
QmiCbkLocEventTimeSyncInd . . . 746
QmiCbkLocInjectPositionInd . . . 747
QmiCbkLocInjectSensorDataInd . . . 748
QmiCbkLocInjectTimeInd . . . 749
QmiCbkLocInjectUTCTimeInd . . . 749
QmiCbkLocPositionReportInd . . . 750
QmiCbkLocSensorStreamingInd . . . 756
QmiCbkLocSetExtPowerConfigInd . . . 756
QmiCbkNasLTECphyCaInfo . . . 757
QmiCbkSwiOmaDmEventStatusReportInd . . . 758
QmiCbkSwiOmaDmEventStatusReportIndExt . . . 758
QmiCbkTmdMitiLvlRptInd . . . 758
QmiCbkWdsStatisticsIndState . . . 759
qmifwinfo_s . . . 760
QmiNas3GppNetworkInfo . . . 761
QmiNasGetRFBandInfoResp . . . 762
QmiNasPerformNetworkScanResp . . . 762
qmiSmsMessageList . . . 763
qmiWDSDataBearerTechnology . . . 763
QmiWdsIpAddressInfo . . . 764
qmiWdsRunTimeSettings . . . 764
QosClassID . . . 768
QosEventInfo . . . 769
QosFlowInfo . . . 771
QosFlowInfoState . . . 772
QosMap . . . 772
RankIndicatorInd . . . 773
readResult . . . 773
readTransparentInfo . . . 774
redirNumInfo . . . 774
registerRefresh . . . 776
remainingRetries . . . 777
remotePartyName . . . 777
remotePartyNum . . . 778
ReqFieldsList . . . 779
RespFieldsList . . . 780
RFBandInfoElements . . . 780
rmTrasnferStaticsReq . . . 781
roamIndList . . . 781
RoamingInfo . . . 782
roamTimer . . . 782
RSRPThresh . . . 783
rsrqInformation . . . 784
RSRQThresh . . . 784
RSSIThresh . . . 785
RXAGCList . . . 786
RXAVCList . . . 787
rxInfo . . . 787

RXPCMIIRFltr . . . 789
RxSigInfo . . . 790
rxSignalStrengthListElement . . . 791
sApnExtraParams . . . 792
satelliteInfo . . . 793
SccRxInfo . . . 796
sensorData . . . 796
sensorData_t . . . 798
sensorDataUsage_s . . . 800
serialNumbersInfo . . . 800
serviceProviderName . . . 802
ServingSystemInfo . . . 802
servSystem . . . 804
sessionInfo . . . 805
sessionInfoExt . . . 806
sessionInfoTlv . . . 806
sessionInfoTlvExt . . . 807
SetAudioPathConfigReq . . . 807
SetAudioProfileReq . . . 809
SetAudioVolTLBConfigReq . . . 810
SetAudioVolTLBConfigResp . . . 811
setCustomSettingV2 . . . 812
setDyingGaspCfg . . . 812
SetIMSSMSConfigReq . . . 813
SetIMSSMSConfigResp . . . 813
SetIMSUserConfigReq . . . 814
SetIMSUserConfigResp . . . 814
SetIMSVoIPConfigReq . . . 815
SetIMSVoIPConfigResp . . . 817
SetM2MAudioAVCFGReq . . . 818
SetM2MAudioLPBKReq . . . 819
SetM2MAudioProfileReq . . . 819
SetM2MAudioVolumeReq . . . 820
SetM2MAVMuteReq . . . 821
SetM2MSpkrGainReq . . . 822
setPINProtection . . . 822
SetRegMgrConfigReq . . . 823
SetRegMgrConfigResp . . . 824
setSignalStrengthInfo . . . 824
SetSIPConfigReq . . . 829
SetSIPConfigResp . . . 830
sGetDeviceSeriesResult . . . 831
sidNid . . . 831
sigInfo . . . 832
signalInfo . . . 833
SignalStrengthDataType . . . 834
slot_t . . . 834
slotInf . . . 835
slotInfo . . . 837
slots_t . . . 838
slqsautoconnect . . . 838
SLQSDeleteProfileParams . . . 839
slqsfwinfo_s . . . 840
SlqsNas3GppNetworkInfo . . . 841
SlqsNasPcsDigit . . . 842
slqssendasyncsmsparams_s . . . 843
slqssendsmsparams_s . . . 845
slqsSessionStateInfo . . . 846

slqsSignalStrengthInfo . . . 847
SLQSSignalStrengthsIndReq . . . 850
SLQSSignalStrengthsInformation . . . 852
slqsWdsEventInfo . . . 853
sms_BroadcastConfig . . . 855
sms_CDMABroadcastConfig . . . 855
sms_getIndicationReg . . . 856
sms_getMsgWaitingInfo . . . 857
sms_getTransLayerInfo . . . 857
sms_getTransNWRegInfo . . . 858
sms_maxStorageSizeReq . . . 859
sms_maxStorageSizeResp . . . 859
sms_messageWaitingInfoContent . . . 860
sms_msgProtocolResp . . . 861
sms_qaQmi3GPP2BroadcastCfgInfo . . . 861
sms_qaQmi3GPPBroadcastCfgInfo . . . 862
sms_routeEntry . . . 862
sms_sendAsyncsmsParams . . . 863
sms_setIndicationReg . . . 866
sms_setRoutesReq . . . 866
sms_transLayerInfo . . . 867
SMSAsyncRawSend_s . . . 868
sMSCAddress . . . 869
SMSCAddress . . . 870
sMSCAddressTlv . . . 870
sMSEtwsMessage . . . 871
SMSEtwsMessage . . . 871
sMSEtwsMessageTlv . . . 872
SMSEtwsPlmn . . . 872
sMSEtwsPlmn . . . 873
SMSEventInfo_s . . . 873
smsMaxStorageSizeReq . . . 875
smsMaxStorageSizeResp . . . 875
SMSMemoryInfo . . . 876
SMSMessageMode . . . 876
sMSMessageMode . . . 877
smsMsgprotocolResp . . . 877
SMSMTMessage . . . 878
sMSMTMessage . . . 878
SMSOnIMS . . . 879
sMSOnIMS . . . 879
sMSOnIMSTlv . . . 879
smsRouteEntry . . . 880
smsSetRoutesReq . . . 881
SMSTransferRouteMTMessage . . . 882
sMSTransferRouteMTMessage . . . 882
sQosFlowStat . . . 883
sQosStat . . . 884
SrvStatusInfo . . . 885
ssdatasession_params . . . 886
SupportedMsgList . . . 889
SUPSInfo . . . 889
SV . . . 890
SVInfo . . . 891
svUsedforFix_s . . . 892
SWI_STRUCT_CarrierImage . . . 892
SwiLocGetAutoStartResp . . . 893
SwiLocSetAutoStartReq . . . 895

swiModemStatusResp . . . 897
SwiOTAMsg_s . . . 897
swiPDPRuntimeSettingsReq . . . 898
swiPDPRuntimeSettingsResp . . . 899
swiQosFilter . . . 901
swiQosFlow . . . 904
swiQosGranted . . . 907
swiQosIds . . . 907
swiQosModifyReq . . . 908
swiQosReq . . . 908
swiRMTrasnferStaticsReq . . . 909
sysInfoCommon . . . 910
t_gpsTime . . . 912
t_sensor . . . 912
t_Sv . . . 913
TDSCDMAECIOThresh . . . 913
TDSCDMARSCPThresh . . . 913
TDSCDMARSSIThresh . . . 914
tdscdmaSigInfoExt . . . 914
TDSCDMASigInfoExt . . . 915
TDSCDMASINRCONFThresh . . . 916
TDSCDMASINRThresh . . . 916
tempData_t . . . 917
tempratureData . . . 918
TFTIDParams . . . 919
timeInfo . . . 921
TmdDeRegNotMitigationLvlReq . . . 922
TmdGetMitigationDevListResp . . . 923
TmdGetMitigationLvlReq . . . 924
TmdGetMitigationLvlResp . . . 924
TmdMitigationLvlIndReq . . . 925
TmdRegNotMitigationLvlReq . . . 925
tokenBucket . . . 926
Tos . . . 926
transferRouteMessageTlv . . . 927
TransferStatInd . . . 927
transferStatInd . . . 928
TransferStatsDataType . . . 928
TrStatInd . . . 928
trueIMSI . . . 929
TXAGCList . . . 930
txInfo . . . 931
TXPCMIIRFltr . . . 932
uim_appStatus . . . 933
uim_authenticateResult . . . 936
uim_authenticationData . . . 937
uim_cardResult . . . 938
uim_cardStatus . . . 939
uim_changeUIMPIN . . . 940
uim_depersonalizationInformation . . . 941
uim_encryptedPIN1 . . . 942
uim_fileAttributes . . . 943
uim_fileInfo . . . 946
uim_hotSwapStatus . . . 947
uim_personalizationStatus . . . 947
uim_readResult . . . 948
uim_readTransparentInfo . . . 949
uim_refreshevent . . . 949

uim_registerRefresh . . . 951
uim_remainingRetries . . . 952
uim_sessionInformation . . . 953
uim_setPINProtection . . . 953
uim_slotInfo . . . 954
uim_UIMSessionInformation . . . 956
uim_unblockUIMPIN . . . 957
uim_verifyUIMPIN . . . 958
UIMAuthenticateReq . . . 958
UIMAuthenticateResp . . . 959
UIMChangePinReq . . . 960
UIMDepersonalizationReq . . . 961
UIMDepersonalizationResp . . . 961
UIMEventRegisterReqResp . . . 961
UIMGetCardStatusResp . . . 962
UIMGetConfigurationReq . . . 962
UIMGetConfigurationResp . . . 963
UIMGetFileAttributesReq . . . 964
UIMGetFileAttributesResp . . . 964
UIMGetSlotsStatusResp . . . 965
UIMPinResp . . . 966
UIMPowerDownReq . . . 966
UIMPowerUpReq . . . 967
UIMReadTransparentReq . . . 967
UIMReadTransparentResp . . . 968
UIMRefreshCompleteReq . . . 969
UIMRefreshEvent . . . 970
UIMRefreshGetLastEventReq . . . 971
UIMRefreshGetLastEventResp . . . 972
UIMRefreshOKReq . . . 972
UIMRefreshRegisterReq . . . 973
UIMSessionInformation . . . 973
UIMSetPinProtectionReq . . . 974
UIMSlotsStatus . . . 975
UIMSlotStatus . . . 975
UIMSlotStatusChangeInfo . . . 977
UIMStatusChangeInfo . . . 977
UIMSwitchSlotReq . . . 977
UIMUnblockPinReq . . . 978
UIMVerifyPinReq . . . 979
UMTSInfo . . . 980
UMTSinstInfo . . . 982
umtsLTENbrCell . . . 982
UMTSMinQoS . . . 983
UMTSQoS . . . 986
UMTSReqQoSSigInd . . . 988
unblockUIMPIN . . . 989
UniversalTime . . . 990
unpack_dms_GetActivationState_t . . . 991
unpack_dms_GetBandCapability_t . . . 992
unpack_dms_GetCrashAction_t . . . 992
unpack_dms_GetCustFeature_t . . . 992
unpack_dms_GetCustFeaturesV2_t . . . 993
unpack_dms_GetDeviceCap_t . . . 994
unpack_dms_GetDeviceCapabilities_t . . . 994
unpack_dms_GetDeviceHardwareRev_t . . . 995
unpack_dms_GetDeviceMfr_t . . . 995
unpack_dms_GetDeviceSerialNumbers_t . . . 995

unpack_dms_GetFirmwareInfo_t . . . 996
unpack_dms_GetFirmwareRevision_t . . . 997
unpack_dms_GetFirmwareRevisions_t . . . 997
unpack_dms_GetFSN_t . . . 998
unpack_dms_GetHardwareRevision_t . . . 998
unpack_dms_GetIMSI_t . . . 999
unpack_dms_GetManufacturer_t . . . 999
unpack_dms_GetModelID_t . . . 999
unpack_dms_GetNetworkTime_t . . . 1000
unpack_dms_GetOfflineReason_t . . . 1001
unpack_dms_GetPower_t . . . 1001
unpack_dms_GetPRLVersion_t . . . 1002
unpack_dms_GetSerialNumbers_t . . . 1002
unpack_dms_GetUSBComp_t . . . 1003
unpack_dms_GetVoiceNumber_t . . . 1003
unpack_dms_ResetToFactoryDefaults_t . . . 1003
unpack_dms_SetActivationStatusCallback_t . . . 1004
unpack_dms_SetCrashAction_t . . . 1004
unpack_dms_SetCustFeature_t . . . 1005
unpack_dms_SetCustFeaturesV2_t . . . 1005
unpack_dms_SetEventReport_ind_t . . . 1005
unpack_dms_SetEventReport_t . . . 1006
unpack_dms_SetFirmwarePreference_t . . . 1006
unpack_dms_SetPower_t . . . 1006
unpack_dms_SetUSBComp_t . . . 1006
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t . . . 1007
unpack_dms_SLQSDmsSwiGetResetInfo_t . . . 1008
unpack_dms_SLQSDmsSwiIndicationRegister_t . . . 1008
unpack_dms_SLQSGetBandCapability_t . . . 1009
unpack_dms_SLQSGetERIFile_t . . . 1012
unpack_dms_SLQSSwiClearDyingGaspStatistics_t . . . 1012
unpack_dms_SLQSSwiGetCrashInfo_t . . . 1013
unpack_dms_SLQSSwiGetDyingGaspCfg_t . . . 1013
unpack_dms_SLQSSwiGetDyingGaspStatistics_t . . . 1014
unpack_dms_SLQSSwiGetFirmwareCurr_t . . . 1014
unpack_dms_SLQSSwiGetFwUpdateStatus_t . . . 1015
unpack_dms_SLQSSwiGetHostDevInfo_t . . . 1017
unpack_dms_SLQSSwiGetOSInfo_t . . . 1017
unpack_dms_SLQSSwiGetSerialNoExt_t . . . 1018
unpack_dms_SLQSSwiSetDyingGaspCfg_t . . . 1018
unpack_dms_SLQSSwiSetHostDevInfo_t . . . 1019
unpack_dms_SLQSSwiSetOSInfo_t . . . 1019
unpack_dms_SLQSUIMGetState_t . . . 1020
unpack_dms_UIMGetControlKeyStatus_t . . . 1020
unpack_dms_UIMGetICCID_t . . . 1021
unpack_dms_UIMGetPINStatus_t . . . 1022
unpack_dms_UIMSetControlKeyProtection_t . . . 1023
unpack_dms_UIMSetPINProtection_t . . . 1024
unpack_dms_UIMUnblockControlKey_t . . . 1024
unpack_fms_GetImagesPreference_t . . . 1025
unpack_fms_GetStoredImages_t . . . 1026
unpack_fms_SetImagesPreference_t . . . 1026
unpack_loc_BestAvailPos_Ind_t . . . 1027
unpack_loc_Delete_Assist_Data_t . . . 1032
unpack_loc_DeleteAssistData_Ind_t . . . 1033
unpack_loc_EngineState_Ind_t . . . 1034
unpack_loc_EventRegister_t . . . 1034
unpack_loc_GnssSvInfo_Ind_t . . . 1035

unpack_loc_PositionRpt_Ind_t . . . 1035
unpack_loc_SetExtPowerConfig_Ind_t . . . 1042
unpack_loc_SetExtPowerState_t . . . 1042
unpack_loc_SetOperationMode_Ind_t . . . 1043
unpack_loc_SetOperationMode_t . . . 1043
unpack_loc_SLQSLOCGetBestAvailPos_t . . . 1044
unpack_loc_Start_t . . . 1044
unpack_loc_Stop_t . . . 1044
unpack_nas_GetCDMANetworkParameters_t . . . 1045
unpack_nas_GetHomeNetwork3GPP2_t . . . 1046
unpack_nas_GetHomeNetwork_t . . . 1048
unpack_nas_GetNetworkPreference_t . . . 1048
unpack_nas_GetRFInfo_t . . . 1049
unpack_nas_GetServingNetwork_t . . . 1050
unpack_nas_GetServingNetworkCapabilities_t . . . 1051
unpack_nas_GetSignalStrengths_t . . . 1051
unpack_nas_PerformNetworkScan_t . . . 1052
unpack_nas_SetDataCapabilitiesCallback_ind_t . . . 1053
unpack_nas_SetEventReportInd_t . . . 1053
unpack_nas_SetNasLTECphyCaIndCallback_ind_t . . . 1053
unpack_nas_SetNetworkPreference_t . . . 1054
unpack_nas_SetRoamingIndicatorCallback_ind_t . . . 1055
unpack_nas_SetServingSystemCallback_ind_t . . . 1056
unpack_nas_SLQSGetErrorRate_t . . . 1056
unpack_nas_SlqsGetLTECphyCAInfo_t . . . 1057
unpack_nas_SLQSGetNetworkTime_t . . . 1058
unpack_nas_SLQSGetOperatorNameData_t . . . 1058
unpack_nas_SLQSGetPLMNName_t . . . 1059
unpack_nas_SLQSGetServingSystem_t . . . 1060
unpack_nas_SLQSGetSignalStrength_t . . . 1062
unpack_nas_SLQSGetSysInfo_t . . . 1063
unpack_nas_SLQSGetSysSelectionPref_t . . . 1066
unpack_nas_SLQSNasGet3GPP2Subscription_t . . . 1069
unpack_nas_SLQSNasGetCellLocationInfo_t . . . 1070
unpack_nas_SLQSNasGetHDRColorCode_t . . . 1072
unpack_nas_SLQSNasGetSigInfo_t . . . 1072
unpack_nas_SLQSNasGetTxRxInfo_t . . . 1073
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t . . . 1073
unpack_nas_SLQSNasSigInfoCallback_ind_t . . . 1074
unpack_nas_SLQSNASSwiGetChannelLock_t . . . 1075
unpack_nas_SLQSNasSwiModemStatus_t . . . 1076
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t . . . 1076
unpack_nas_SLQSNasTimerCallback_ind_t . . . 1077
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t . . . 1077
unpack_nas_SLQSSwiGetHDRPersonality_t . . . 1078
unpack_nas_SLQSSwiGetHDRProtSubtype_t . . . 1078
unpack_nas_SLQSSwiGetHRPDStats_t . . . 1079
unpack_nas_SLQSSwiGetLteCQI_t . . . 1080
unpack_nas_SLQSSwiGetLteSccRxInfo_t . . . 1080
unpack_nas_SLQSSwiNetworkDebug_t . . . 1081
unpack_nas_SLQSSysInfoCallback_ind_t . . . 1082
unpack_omaDmConfigTlv_t . . . 1084
unpack_omaDmFotaTlv_t . . . 1085
unpack_omaDmNotificationsTlv_t . . . 1087
unpack_pds_GetPDSDefaults_t . . . 1088
unpack_pds_GetPDSState_t . . . 1088
unpack_pds_GetPortAutomaticTracking_t . . . 1089
unpack_qmi_t . . . 1089

unpack_qos_dataRate_t . . . 1090
unpack_qos_IPv4Addr_t . . . 1090
unpack_qos_IPv6Addr_t . . . 1091
unpack_qos_IPv6TrafCls_t . . . 1091
unpack_qos_pktErrRate_t . . . 1092
unpack_qos_Port_t . . . 1092
unpack_qos_QosFlowInfo_t . . . 1093
unpack_qos_QosFlowInfoState_t . . . 1094
unpack_qos_SLQSQosGetNetworkStatus_t . . . 1095
unpack_qos_SLQSQosSwiReadApnExtraParams_t . . . 1095
unpack_qos_SLQSQosSwiReadDataStats_t . . . 1097
unpack_qos_SLQSSetQosEventCallback_ind_t . . . 1098
unpack_qos_SLQSSetQosNWStatusCallback_ind_t . . . 1098
unpack_qos_SLQSSetQosPriEventCallback_ind_t . . . 1099
unpack_qos_SLQSSetQosStatusCallback_ind_t . . . 1099
unpack_qos_swiQosFilter_t . . . 1101
unpack_qos_swiQosFlow_t . . . 1104
unpack_qos_tokenBucket_t . . . 1108
unpack_qos_Tos_t . . . 1109
unpack_QosFlowStat_t . . . 1109
unpack_rms_GetSMSWake_t . . . 1110
unpack_RMTransferStatistics_ind_t . . . 1110
unpack_sar_SLQSGetRfSarState_t . . . 1111
unpack_sms_GetSMSCAddress_t . . . 1112
unpack_sms_SaveSMS_t . . . 1113
unpack_sms_SendSMS_t . . . 1113
unpack_sms_SetNewSMSCallback_ind_t . . . 1113
unpack_sms_SetNewSMSCallback_t . . . 1115
unpack_sms_SLQSDeleteSMS_t . . . 1115
unpack_sms_SLQSGetIndicationRegister_t . . . 1115
unpack_sms_SLQSGetMessageWaiting_t . . . 1115
unpack_sms_SLQSGetSMS_t . . . 1116
unpack_sms_SLQSGetSmsBroadcastConfig_t . . . 1116
unpack_sms_SLQSGetSMSList_t . . . 1117
unpack_sms_SLQSGetTransLayerInfo_t . . . 1117
unpack_sms_SLQSGetTransNWRegInfo_t . . . 1118
unpack_sms_SLQSModifySMSStatus_t . . . 1118
unpack_sms_SLQSSmsGetMaxStorageSize_t . . . 1118
unpack_sms_SLQSSmsGetMessageProtocol_t . . . 1119
unpack_sms_SLQSSwiGetSMSStorage_t . . . 1119
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t . . . 1120
unpack_swiloc_SwiLocGetAutoStart_t . . . 1120
unpack_swioma_SLQSOMADMAlertCallback_ind_t . . . 1122
unpack_swioma_SLQSOMADMGetSessionInfo_t . . . 1123
unpack_swioma_SLQSOMADMGetSettings_t . . . 1125
unpack_swioma_SLQSOMADMStartSession_t . . . 1127
unpack_uim_ChangePin_t . . . 1127
unpack_uim_GetCardStatus_t . . . 1128
unpack_uim_ReadTransparent_t . . . 1129
unpack_uim_SetPinProtection_t . . . 1130
unpack_uim_SetUimSlotStatusChangeCallback_ind_t . . . 1130
unpack_uim_SLQSUIMAuthenticate_t . . . 1131
unpack_uim_SLQSUIMDepersonalization_t . . . 1131
unpack_uim_SLQSUIMEventRegister_t . . . 1132
unpack_uim_SLQSUIMGetConfiguration_t . . . 1132
unpack_uim_SLQSUIMGetFileAttributes_t . . . 1133
unpack_uim_SLQSUIMGetSlotsStatus_t . . . 1134
unpack_uim_SLQSUIMRefreshGetLastEvent_t . . . 1134

unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t . . . 1135
unpack_uim_UnblockPin_t . . . 1135
unpack_uim_VerifyPin_t . . . 1136
unpack_voice_SLQSOriginateUSSD_t . . . 1136
unpack_voice_SLQSVoiceAnswerCall_t . . . 1137
unpack_voice_SLQSVoiceBurstDTMF_t . . . 1137
unpack_voice_SLQSVoiceDialCall_t . . . 1138
unpack_voice_SLQSVoiceEndCall_t . . . 1139
unpack_voice_SLQSVoiceGetAllCallInfo_t . . . 1139
unpack_voice_SLQSVoiceGetCallBarring_t . . . 1142
unpack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 1143
unpack_voice_SLQSVoiceGetCallInfo_t . . . 1144
unpack_voice_SLQSVoiceGetCallWaiting_t . . . 1147
unpack_voice_SLQSVoiceGetCLIP_t . . . 1149
unpack_voice_SLQSVoiceGetCLIR_t . . . 1150
unpack_voice_SLQSVoiceGetCNAP_t . . . 1151
unpack_voice_SLQSVoiceGetCOLP_t . . . 1152
unpack_voice_SLQSVoiceGetCOLR_t . . . 1154
unpack_voice_SLQSVoiceGetConfig_t . . . 1155
unpack_voice_SLQSVoiceManageCalls_t . . . 1157
unpack_voice_SLQSVoiceSendFlash_t . . . 1157
unpack_voice_SLQSVoiceSetCallBarringPassword_t . . . 1157
unpack_voice_SLQSVoiceSetConfig_t . . . 1159
unpack_voice_SLQSVoiceSetSUPSService_t . . . 1160
unpack_voice_SLQSVoiceStartContDTMF_t . . . 1161
unpack_voice_SLQSVoiceStopContDTMF_t . . . 1162
unpack_wds_DHCPv4ClientLease_ind_t . . . 1162
unpack_wds_GetAutoconnect_t . . . 1163
unpack_wds_GetByteTotals_t . . . 1163
unpack_wds_GetConnectionRate_t . . . 1164
unpack_wds_GetDataBearerTechnology_t . . . 1164
unpack_wds_GetDefaultProfile_t . . . 1165
unpack_wds_GetDefaultProfileNum_t . . . 1166
unpack_wds_GetDormancyState_t . . . 1167
unpack_wds_GetLastMobileIPError_t . . . 1167
unpack_wds_GetMobileIP_t . . . 1167
unpack_wds_GetMobileIPProfile_t . . . 1168
unpack_wds_GetPacketStatistics_t . . . 1169
unpack_wds_GetPacketStatus_t . . . 1170
unpack_wds_GetSessionDuration_t . . . 1171
unpack_wds_GetSessionState_t . . . 1172
unpack_wds_RMSetTransferStatistics_t . . . 1172
unpack_wds_SetMobileIPProfile_t . . . 1172
unpack_wds_SLQSCreateProfile_t . . . 1172
unpack_wds_SLQSDeleteProfile_t . . . 1173
unpack_wds_SLQSGet3GPPConfigItem_t . . . 1173
unpack_wds_SLQSGetCurrDataSystemStat_t . . . 1174
unpack_wds_SLQSGetCurrentChannelRate_t . . . 1175
unpack_wds_SLQSGetDataBearerTechnology_t . . . 1176
unpack_wds_SLQSGetDUNCallInfo_t . . . 1176
unpack_wds_SLQSGetProfileSettings_t . . . 1177
unpack_wds_SLQSGetRuntimeSettings_t . . . 1178
unpack_wds_SLQSModifyProfile_t . . . 1180
unpack_wds_SLQSSetIPFamilyPreference_t . . . 1180
unpack_wds_SLQSSetPacketSrvStatusCallback_t . . . 1180
unpack_wds_SLQSSetWdsEventCallback_ind_t . . . 1181
unpack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 1183
unpack_wds_SLQSSGetLoopback_t . . . 1184

unpack_wds_SLQSStartDataSession_t . . . 1184
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 1185
UnPackGetProfileSettingOut . . . 1186
unpackWdsProfileParam . . . 1187
USBCompConfig . . . 1187
USBCompParams . . . 1188
USSDNoWaitIndicationInfo . . . 1190
USSDRespFNetwork . . . 1191
USSInfo . . . 1191
USSResp . . . 1192
UUSInfo . . . 1193
verifyUIMPIN . . . 1194
voice_airTimer . . . 1194
voice_allCallsAlphaIDInfo . . . 1195
voice_allCallsDiagInfo . . . 1195
voice_allCallsUUSInfo . . . 1196
voice_alphaIDInfo . . . 1196
voice_arrAlertingPattern . . . 1197
voice_arrAlertingType . . . 1198
voice_arrAlphaID . . . 1199
voice_arrCalledPartyNum . . . 1199
voice_arrCallEndReason . . . 1200
voice_arrCallInfo . . . 1200
voice_arrConnectPartyNum . . . 1201
voice_arrDiagInfo . . . 1202
voice_arrRedirPartyNum . . . 1202
voice_arrRemotePartyName . . . 1203
voice_arrRemotePartyNum . . . 1203
voice_arrSvcOption . . . 1204
voice_arrUUSInfo . . . 1204
voice_burstDTMFInfo . . . 1205
voice_calledPartySubAdd . . . 1206
voice_callFwdTypeAndPlan . . . 1206
voice_callFWExtInfo . . . 1207
voice_callFWInfo . . . 1209
voice_callInfo . . . 1210
voice_ccSUPSType . . . 1212
voice_CLIPResp . . . 1212
voice_CLIRResp . . . 1213
voice_CNAPResp . . . 1214
voice_COLPResp . . . 1215
voice_COLRResp . . . 1215
voice_connectNumInfo . . . 1216
voice_CUGInfo . . . 1218
voice_curAMRConfig . . . 1218
voice_diagInfo . . . 1219
voice_DTMFLengths . . . 1220
voice_getAllCallInformation . . . 1220
voice_getAllCallRmtPtyName . . . 1221
voice_getAllCallRmtPtyNum . . . 1221
voice_getCallFWExtInfo . . . 1222
voice_getCallFWInfo . . . 1223
voice_peerNumberInfo . . . 1223
voice_prefVoiceSO . . . 1225
voice_remotePartyName . . . 1227
voice_remotePartyNum . . . 1228
voice_roamTimer . . . 1228
voice_USSInfo . . . 1229

voice_UUSInfo . . . 1230
voiceALSSelectLineInfo . . . 1231
voiceALSSetLineSwitchInfo . . . 1231
voiceAnswerCall . . . 1232
voiceBindSubscriptionInfo . . . 1232
voiceBurstDTMFInfo . . . 1232
voiceCallInfoReq . . . 1233
voiceCallInfoResp . . . 1233
voiceCallRequestParams . . . 1236
voiceCallResponseParams . . . 1238
voiceContDTMFinfo . . . 1239
voiceDTMFEventInfo . . . 1240
voiceFlashInfo . . . 1240
voiceGetAllCallInfo . . . 1241
voiceGetCallBarringReq . . . 1244
voiceGetCallBarringResp . . . 1244
voiceGetCallFWReq . . . 1246
voiceGetCallFWResp . . . 1246
voiceGetCallWaitInfo . . . 1248
voiceGetCLIPResp . . . 1249
voiceGetCLIRResp . . . 1250
voiceGetCNAPResp . . . 1252
voiceGetCOLPResp . . . 1253
voiceGetCOLRResp . . . 1254
voiceGetConfigReq . . . 1255
voiceGetConfigResp . . . 1257
voiceIndicationRegisterInfo . . . 1259
voiceInfoRec . . . 1260
voiceManageCallsReq . . . 1262
voiceManageCallsResp . . . 1263
voiceOrigUSSDNoWaitInfo . . . 1264
voiceOTASPStatusInfo . . . 1264
voicePrivacyInfo . . . 1265
voiceSetAllCallStatusCbkInfo . . . 1266
voiceSetCallBarringPwdInfo . . . 1268
voiceSetCallBarringPwdResp . . . 1269
voiceSetConfigReq . . . 1270
voiceSetConfigResp . . . 1272
voiceSetPrefPrivacy . . . 1273
voiceSetSUPSServiceReq . . . 1274
voiceSetSUPSServiceResp . . . 1276
voiceStopContDTMFinfo . . . 1277
voiceSUPSInfo . . . 1278
voiceSUPSNotification . . . 1280
wcdmaCellInfo . . . 1282
WCDMAECIOThresh . . . 1283
WCDMAInfoLTENeighborCell . . . 1283
wcdmaLongMsgDecodingParams . . . 1284
wcdmaMsgDecodingParams . . . 1286
wcdmaMsgEncodingParams . . . 1287
WCDMARSSIThresh . . . 1288
WCDMASysInfo . . . 1289
wcdmaUARFCN . . . 1292
wds_currNetworkInfo . . . 1293
wds_DataULongLongTlv . . . 1294
wds_DataULongTlv . . . 1294
wds_DHCPLeaseOptTlv . . . 1295
wds_DHCPLeaseStateTlv . . . 1295

wds_DHCPOpt . . . 1295
wds_DHCPProfileIdTlv . . . 1296
wds_DHCPv4HWConfig . . . 1296
wds_DHCPv4Option . . . 1297
wds_DHCPv4OptionList . . . 1297
wds_DHCPv4ProfileId . . . 1298
wds_Domain . . . 1298
wds_DomainNameList . . . 1299
wds_GPRSQoS . . . 1299
wds_IPv4AdTlv . . . 1300
wds_IPV6AddressInfo . . . 1300
wds_IPV6GWAddressInfo . . . 1301
wds_PCSCFFQDNAddress . . . 1301
wds_PCSCFFQDNAddressList . . . 1302
wds_PCSCFIPv4ServerAddressList . . . 1302
wds_ProfileIdentifier . . . 1303
wds_profileInfo . . . 1303
wds_TrStatInd . . . 1304
wds_UMTSMinQoS . . . 1304
WdsByteTotals . . . 1307
WdsByteTotalsElmnts . . . 1308
WdsClientLeaseChange . . . 1308
WdsConnectionRate . . . 1309
WdsConnectionRateElmnts . . . 1310
WdsDHCPv4ClientLeaseInd . . . 1310
WdsDHCPv4Config . . . 1311
wdsDhcpv4HwConfig . . . 1312
WdsDHCPv4HWConfig . . . 1312
wdsDhcpv4Option . . . 1313
WdsDHCPv4Option . . . 1314
WdsDHCPv4OptionList . . . 1314
wdsDhcpv4OptionList . . . 1315
WdsDHCPv4ProfileId . . . 1315
wdsDhcpv4ProfileId . . . 1316
WDSGetLoopbackData . . . 1316
WdsIpAddressInfoReq . . . 1317
WdsPktStatisticsElmnts . . . 1317
WdsPktStatisticsReq . . . 1319
WdsPktStatisticsResp . . . 1319
WdsProfileParam . . . 1320
WdsRunTimeSettings . . . 1320
wdsSetEventReportReq . . . 1321
WDSSetLoopbackData . . . 1323
WDSSWICurrentChannelRates . . . 1323

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 . . . 1325
common.h . . . 1325
dms.h . . . 1330
fms.h . . . 1381
loc.h . . . 1384
nas.h . . . 1399
pds.h . . . 1441
qaCbkCatEventReportInd.h . . . 1445
qaCbkSwiOmaDmEventReportInd.h . . . 1447
qaGobiApiAudio.h
Audio Service API function prototypes . . . 1448
qaGobiApiCat.h
Card Application Toolkit API function headers . . . 1452
qaGobiApiCbk.h
Callback Service API function prototypes . . . 1453
qaGobiApiDcs.h
Device Connectivity Service API function prototypes . . . 1538
qaGobiApiDms.h
Device Management Service API function prototypes . . . 1546
qaGobiApiFms.h
Firmware Management Service API function prototypes . . . 1584
qaGobiApiIms.h
IMS Service API function prototypes . . . 1603
qaGobiApiImsa.h
IMSA Service API function prototypes . . . 1609
qaGobiApiLoc.h
Location API function prototypes . . . 1612
qaGobiApiNas.h
Network Access Service API function prototypes . . . 1619
qaGobiApiOmadm.h
Open Mobile Alliance Device Management Service API function prototypes . . . 1666
qaGobiApiPds.h
Position Determination Service API function prototypes . . . 1669
qaGobiApiQos.h
Quality of Service API function prototypes . . . 1684

qaGobiApiRms.h
Remote Management Service API function prototypes . . . 1690
qaGobiApiSar.h
Specific Absorption Rate API function prototypes . . . 1692
qaGobiApiSms.h
Short Message Service API function prototypes . . . 1694
qaGobiApiSwi.h
SWI API function prototypes . . . 1716
qaGobiApiSwiAudio.h
M2M Audio Service API function prototypes . . . 1717
qaGobiApiSwiAvms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI AVMS QMI Service revision 1.14 . . . 1723
qaGobiApiSwiOmadms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6 . . . 1735
qaGobiApiTableBandClasses.h
Network Access Service API Band Classes table . . . 1746
qaGobiApiTableCallControlReturnReasons.h
Call Control Return Reasons table . . . 1749
qaGobiApiTableCallEndReasons.h
Wireless Data Service Call End Reasons . . . 1750
qaGobiApiTableCarrierCodes.h
Carrier Codes table . . . 1765
qaGobiApiTableCodingScheme.h
Data Coding Scheme . . . 1767
qaGobiApiTableGpsCapabilityCodes.h
Position Determination Service API GPS Capability Codes . . . 1770
qaGobiApiTablePowerModes.h
Device Management Service API Power Modes table . . . 1770
qaGobiApiTableRadioInterfaces.h
Network Access Service API Radio Interfaces table . . . 1771
qaGobiApiTableRegionCodes.h
Region Codes table . . . 1772
qaGobiApiTableServiceOptions.h
Voice Service Options . . . 1772
qaGobiApiTableSupServiceInfoClasses.h
Voice Supplementary Service Information Classes . . . 1774
qaGobiApiTableSwiAudio.h
Swi Audio related tables . . . 1775
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h
Update Complete Status table . . . 1775
qaGobiApiTableVoiceCallEndReasons.h
Voice Service Call and supplementary services end reasons . . . 1777
qaGobiApiTmd.h
Thermal Mitigation Device API function prototypes . . . 1783
qaGobiApiUim.h
Uim Service API function prototypes . . . 1785
qaGobiApiVoice.h
Voice Service API function prototypes . . . 1798
qaGobiApiWds.h
Wireless Data Service API function prototypes . . . 1818
qaNasGetRFBandInfo.h . . . 1860
qaNasPerformNetworkScan.h . . . 1861
qmerrno.h . . . 1862
qos.h . . . 1869
rms.h . . . 1877
sar.h . . . 1879

sms.h 1881
SwiDataTypes.h
SWI data types 1903
swiloc.h 1904
swioma.h 1906
SWIWWANCMAPI.h 1915
uim.h 1915
voice.h 1933
wds.h 1960

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 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

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	- • Optional 1 BYTE parameter indicating registration status of transport layer information events • Values: – 0x00 - Disabled – 0x01 - Enabled • 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	- • Optional 1 BYTEparameter indicating registration status of transport network registration information events • Values: – 0x00 - Disabled – 0x01 - Enabled • 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	- • Optional 1 BYTE parameter indicating registration status of call status information events • Values: – 0x00 - Disabled – 0x01 - Enabled • 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 – 0x00 - 3GPP – 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 • See QmiProfileInfo for more details

pExtErrCode	• pointer to a 2 byte extended error code • Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device. • 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: – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down
source	• entity which initiated the reset or power down, possible values listed below: – 0 - unknown – 1 - user requested – 2 - hardware switch – 3 - temperature critical – 4 - voltage critical – 5 - configuration update – 6 - LWM2M – 7 - OMA-DM – 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	- • Optional parameter indicating if transport layer is registered • Values: – 0x00 - Transport layer is not registered – 0x01 - Transport layer is registered • function SLQSGetTransLayerInfo() returns a default value 0xFF if no response is received from the device.
pTransLayerInfo	• Pointer to structure of transLayerInfo. – Optional parameter – See transLayerInfo for more information • 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	- • Optional 1 BYTE parameter indicating transport layer network registration status • Values: – 0x00 - No service – 0x01 - In progress – 0x02 - Failed – 0x03 - Limited Service – 0x04 - Full Service • 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 – 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 • Values: – 0 - unknown – 1 - normal – 2 - high(warning) – 3 - high(critical) – 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 – 1 - Disconnected – 2 - Connected – 3 - Suspended – 4 - Authenticating
reconfigReqd	• Indicates if the network interface on the host needs to be reconfigured – 0 - No need to reconfigure – 1 - Reconfiguration required
sessionEnd-Reason	• See qaGobiApiTableCallEndReasons.h for Call End Reason, 0xFFFF means invalid value
verboseSessn-EndReasonType	• Call End Reason Type – 0 - Unspecified – 1 - Mobile IP – 2 - Internal – 3 - Call Manager defined – 6 - 3GPP Specification defined – 7 - PPP – 8 - EHRPD – 9 - IPv6 – 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 – 4 - IPv4 – 6 - IPv6 – 0xFF - invalid value
techName	• Technology name of the packet data connection. – 32767 - CDMA – 32764 - UMTS – 30592 - EPC – 30590 - EMBMS – 30584 - Modem Link Local – 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 – 0x00 - Deactivated – 0x01 - Activated
num_instances	• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. – serviceCategory – language – selected
broadcastConfig	• A CDMABroadcastConfig structure array. • 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 – 0x00 - Deactivated – 0x01 - Activated
num_instances	• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. – fromServiceId – toServiceId – selected
broadcastConfig	• A BroadcastConfig structure array. • 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 • 1 - Firmware. • 2 - User App. • 3 - Legato Framework.
bStat	: State • 0x01 - No binary update available. • 0x02 - Query binary Download. • 0x03 - Binary Downloading. • 0x04 - Binary downloaded. • 0x05 -Query Binary Update. • 0x06 - Binary updating. • 0x07 - Binary updated.
bUserInput-Request	Bit mask of available user inputs. • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 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. • 0x01 - Mandatory. • 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. • 0x01 - AVMS Read Request. • 0x02 - AVMS Change Request. • 0x03 - AVMS Config Complete.
bUserInput-Request	Bit mask of available user inputs. • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 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. • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 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. • 0: Data session closed. • 1: Data session activated. • 2: Register made. -3: Data session error.
wErrorCode	LWM2M Session error code. • 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. • 100 - Continue • 101 - Switching Protocols • 200 - OK • 201 - Created • 202 - Accepted • 203 - Non-Authoritative Information • 204 - No Content • 205 - Reset Content • 206 - Partial Content • 300 - Multiple Choices • 301 - Moved Permanently • 302 - Found • 303 - See Other • 304 - Not Modified • 305 - Use Proxy • 307 - Temporary Redirect • 400 - Bad Request • 401 - Unauthorized • 402 - Payment Required • 403 - Forbidden • 404 - Not Found • 405 - Method Not Allowed • 406 - Not Acceptable • 407 - Proxy Authentication Required • 408 - Request Timeout • 409 - Conflict • 410 - Gone • 411 - Length Required • 412 - Precondition Failed • 413 - Payload Too Large • 414 - URI Too Long • 415 - Unsupported Media Type • 416 - Range Not Satisfiable • 417 - Expectation Failed • 426 - Upgrade Required • 500 - Internal Server Error • 501 - Not Implemented • 502 - Bad Gateway • 503 - Service Unavailable • 504 - Gateway Timeout • 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. • 0x14 - Module starts sending data to server. • 0x15 - Authentication with the server. • 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. • 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 wasa updated • 0x7F16: Factory reset successful: Factory reset succeeded • 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. • 0: Need Bootstrap. • 1: Bootstrap made. • 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. • 0: Bootstrap session. • 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. • By default set to 0xFF for all changes. • 0x01 - device_login • 0x02 - device_MD5_key • 0x04 - server_login • 0x08 - server_MD5_key • 0x10 - server_URL • 0x20 - Nonce • 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	- • Optional 1 BYTE parameter indicating registration status of transport layer information events • Values: – 0x00 - Disabled – 0x01 - Enabled – 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	- • Optional 1 BYTE parameter indicating registration status of transport network registration information events • Values: – 0x00 - Disabled – 0x01 - Enabled – 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	- • Optional 1 BYTE parameter indicating registration status of call status information events • Values: – 0x00 - Disabled – 0x01 - Enabled – 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 – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) – function SLQSGet3GPPConfigItem() returns a default value 255 if no LTE Attach Profile is configured • This setting is deprecated on MC/EM74xx
pProfileList	• Profile List – an array of 4 profile configurations – Each element points to a single WORD value indicating profile – Optional parameter with possible values * 1 - 16 (MC/EM73xx and before) * 1 - 24 (MC/EM74xx and onwards) – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
pDefaultPDN-Enabled	• Always Connect Default PDN – A single BYTE value indicating the status of Always connect default PDN * 0 - disabled * 1 - enabled – Optional parameter – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
p3gppRelease	• 3gpp release – A single BYTE value indicating the 3gpp release * 0 - Release 99 * 1 - Release 5 * 2 - Release 6 * 3 - Release 7 * 4 - Release 8 – In 9x30 and onwards * 5 - Release 9 * 6 - Release 10 * 7 - Release 11 – Optional parameter – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present

pLTEAttach-ProfileList	• pointer to WORD array indicating LTE Attach Profile List – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – 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 – valid range: 1-24 * 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 * pFwAutodownload
- BYTE * pFwAutoUpdate
- 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 – 0x00 - FALSE – 0x01 - TRUE
bAutoReboot[IN]	• Automatic device reboot when the request is received from AirVantage server – 0x00 - FALSE – 0x01 - TRUE
bFwAutodownload[IN]	• Firmware Auto Download – 0x00 - FALSE – 0x01 - TRUE
bFwAutoUpdate[IN]	• Firmware Auto Update – 0x00 - FALSE – 0x01 - TRUE
FwAutoSDM[Optional]	- OMA Automatic UI Alert Response • 0x00 - DISABLED • 0x01 - ENABLED ACCEPT • 0x02 - ENABLED REJECT
PollingTimer[Optional]	• 0-525600 (min) • 0:disabled
ConnectionRetryTimers[Optional]	• See QmiSwiAvmsSetSettingsConnectionRetryTimers for more information
APNInfo[Optional]	• See QmiSwiAvmsSetSettingsAPNInfo for more information
NotifStore[Optional]	• 0 : Disabled. • 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::pFwAutodownload

8.27.2.6 BYTE* _SLQSAVMSSetSettingsReqPara::pFwAutoSDM

8.27.2.7 BYTE* _SLQSAVMSSetSettingsReqPara::pFwAutoUpdate

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 * pFwAutodownload
- BYTE * pFwAutoUpdate
- 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::pFwAutodownload

8.28.1.6 BYTE* _SLQSAVMSSettings::pFwAutoSDM

8.28.1.7 BYTE* _SLQSAVMSSettings::pFwAutoUpdate

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 – 0x04 - GERAN – 0x05 - UMTS – 0x08 - LTE – 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. • Upon successful output, the actual number of elements in the network info instance array.
pNetworkInfo[OUT]	• Network info instance array – See SlqsNas3GppNetworkInfo for more information
pRATInstances[IN/OUT]	• Upon input, maximum number of elements that the RAT info instance array can contain. • Upon successful output, the actual number of elements in the RAT info instance array.
pRATInfo[OUT]	• RAT info instance array – See SlqsNas3GppNetworkRAT for more information
pPCSDigitInstances[IN/OUT]	• Upon input, maximum number of elements that the PCS Digit info instance array can contain. • Upon successful output, the actual number of elements in the PCS Digit info instance array.
pPCSDigitInfo[OUT]	• PCS Digit info instance array – See SlqsNasPcsDigit for more information
pScanResult[OUT]	• status of network scan • 0x00 - scan successful • 0x01 - scan was aborted • 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) – 0x01 - No Firmware available – 0x02 - Query Firmware Download – 0x03 - Firmware Downloading – 0x04 - Firmware Downloaded – 0x05 - Query Firmware Update – 0x06 - Firmware Updating – 0x07 - Firmware Updated
pUpdate-CompleteStatus	• 2 byte parameter indicating Update Complete Status(optional) – See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h Update Complete Status
pSeverity	• 1 byte parameter indicating severity(optional) – 0x01 - Mandatory – 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) – 0x00 - No session since boot – 0x01 - Sprint CI-DC Session – 0x02 - Sprint CI-PRL Session – 0x03 - Sprint CI-FUMO Session – 0x04 - Sprint HFA-DC Session – 0x05 - Sprint HFA-PRL Session – 0x06 - Sprint HFA-FUMO Session – 0x07 - Sprint NI Session
pSessionState	• 1 byte parameter indicating session state(optional) – 0x01 - idle – 0x02 - active – 0x03 - pending
pRetryCount	• 1 byte parameter indicating retries left count(optional) – 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 – 0x00000001 - Client-initiated device configuration – 0x00000002 - Network-initiated device configuration – 0x00000010 - Client-initiated FUMO – 0x00000020 - Network-initiated FUMO • function SLQSOMADMGetSettings2() returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.
pFOT-Adownload[OUT]	• 1 Byte parameter indicating support for FOTA Automatic download – 0x00 - Host permission required before downloading – 0x01 - Automatically start downloading, no host permission required – 0x02 - Automatically start downloading, while not roaming – 0x03 - Automatically reject download – 0x04 - Automatically reject download with “Enterprise Reject Policy” • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
pFOTAUpdate[-OUT]	• 1 byte parameter indicating FOTA Automatic update – 0x00 - User permission required before updating firmware – 0x01 - No user permission required before updating firmware – 0x02 - User permission required, auto update on power up • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
pAutosdm[OUT]	• 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
pFwAutoCheck[-OUT]	• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 0x01 - Enabled • 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 – 0x00 - Firmware auto download FALSE – 0x01 - Firmware auto download TRUE
FOTAUpdate	• 1 byte parameter indicating FOTA Automatic update – 0x00 - Firmware auto update FALSE – 0x01 - Firmware auto update TRUE
pAutosdm[IN]	• Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 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 – 0x00 - Firmware auto download FALSE – 0x01 - Firmware auto download TRUE
FOTAUpdate	• 1 byte parameter indicating FOTA Automatic update – 0x00 - Firmware auto update FALSE – 0x01 - Firmware auto update TRUE
pAutosdm[IN]	• Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject
pFwAutoCheck[-IN]	• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 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) • Null terminated ASCII String
bModelSize[IN/OUT]	• Host Device Model String Size
pModelString[OUT]	• Host Device Model String(Optional parameter) • Null terminated ASCII string.
bSWVerSize[IN/OUT]	• Host Device Software Version String Size
pSWVerString[OUT]	• Host Device Software Version String(Optional parameter) • Null terminated ASCII string
bPlasmaIDSize[IN/OUT]	• Host Device Plasma ID String Size
pPlasmaIDString[OUT]	• Host Device Plasma ID String(Optional parameter) • 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) • Null terminated ASCII string
bVersionSize[I-N/OUT]	• Operating System Version Size
pVersionString[-OUT]	• Operating System Version String(Optional parameter) • 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 • 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) • Null terminated ASCII String
bModelSize[IN]	• Host Device Model String Size
pModelString[IN]	• Host Device Model String(Optional parameter) • Null terminated ASCII string.
bSWVerSize[IN]	• Host Device Software Version String Size
pSWVerString[IN]	• Host Device Software Version String(Optional parameter) • Null terminated ASCII string
bPlasmaIDSize[IN]	• Host Device Plasma ID String Size
pPlasmaIDString[IN]	• Host Device Plasma ID String(Optional parameter) • 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) • Null terminated ASCII string
bVersionSize[IN]	• Operating System Version Size
pVersionString[IN]	• Operating System Version String(Optional parameter) • 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 • Values: – 0 - OFF (normal) – 1 - ON (Emergency) • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.

pBandPref	• Optional parameter • Bit mask representing the band preference • Bit values: – Bit 0 - Band Class 0, A-System – Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band – Bit 2 - Band Class 1, all blocks – Bit 3 - Band Class 2 place holder – Bit 4 - Band Class 3, A-System – Bit 5 - Band Class 4, all blocks – Bit 6 - Band Class 5, all blocks – Bit 7 - GSM_DCS_1800 band – Bit 8 - GSM Extended GSM (E-GSM) 900 band – Bit 9 - GSM Primary GSM (P-GSM) 900 band – Bit 10 - Band Class 6 – Bit 11 - Band Class 7 – Bit 12 - Band Class 8 – Bit 13 - Band Class 9 – Bit 14 - Band Class 10 – Bit 15 - Band Class 11 – Bit 16 - GSM 450 band – Bit 17 - GSM 480 band – Bit 18 - GSM 750 band – Bit 19 - GSM 850 band – Bit 20 - GSM Railways GSM 900 Band – Bit 21 - GSM PCS 1900 band – Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band – Bit 23 - WCDMA U.S. PCS 1900 band – Bit 24 - WCDMA Europe and China DCS 1800 band – Bit 25 - WCDMA U.S. 1700 band – Bit 26 - WCDMA U.S. 850 band – Bit 27 - WCDMA Japan 800 band – Bit 28 - Band Class 12 – Bit 29 - Band Class 14 – Bit 30 - Reserved – Bit 31 - Band Class 15 – Bit 32 to 47 - Reserved – Bit 48 - WCDMA Europe 2600 band – Bit 49 - WCDMA Europe and Japan 900 band – Bit 50 - WCDMA Japan 1700 band – Bit 51 to 55 - Reserved – Bit 56 - Band Class 16 – Bit 57 - Band Class 17 – Bit 58 - Band Class 18 – Bit 59 - Band Class 19 – Bit 60 to 64 - Reserved • function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.

pPRLPref	• Optional parameter indicating the CDMA PRL Preference • Values: – 0x0001 - Acquire available system only on the A side – 0x0002 - Acquire available system only on the B side – 0x3FFF - Acquire any available systems • function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device.
pRoamPref	• Optional parameter indicating the roaming Preference • Values: – 0x01 - Acquire only systems for which the roaming indicator is off – 0x02 - Acquire a system as long as its roaming indicator is not off – 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only – 0xFF - Acquire systems, regardless of their roaming indicator • function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device.

pLTEBandPref	• Optional parameter • Bit mask representing the LTE band preference • Bit Values – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – All other bits are reserved • function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.

pNetSelPref	• Optional parameter indicating network selection preference • Values: – 0x00 - Automatic network selection – 0x01 - Manual network selection • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.
pSrvDomainPref	• Optional parameter indicating Service domain preference • Values: – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach • 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 • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM • 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 • Values: – 0 - OFF (normal) – 1 - ON (Emergency)
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE

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

pPRLPref	• Optional parameter indicating the CDMA PRL Preference • Values: – 0x0001 - Acquire available system only on the A side – 0x0002 - Acquire available system only on the B side – 0x3FFF - Acquire any available systems
pRoamPref	• Optional parameter indicating the roaming Preference • Values: – 0x01 - Acquire only systems for which the roaming indicator is off – 0x02 - Acquire a system as long as its roaming indicator is not off – 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only – 0xFF - Acquire systems, regardless of their roaming indicator

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

pNetSelPref	- netSelectionPref • Optional parameter for specifying Network Selection Preference • Modem selects networks based on this parameter(if present). • Either of pNetSelPref or pCSGID can be set. • see netSelectionPref for more information
pChgDuration	• Optional parameter specifying the duration of the change • At least one system selection setting to be set if pChgDuration is populated. • Values: – 0x00 - Power cycle - Remains active until the next device power cycle – 0x01 - Permanent - Remains active through power cycles until changed by client – Device will use "0x01 - permanent" as default if this parameter is omitted
pMNCIncPCS-DigStat	• Optional parameter indicating if MNC includes PCS digit • pNetSelPref is expected if MNC includes PCS digit is set to 1. • Values: – TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
pSrvDomainPref	• Optional parameter indicating Service domain preference • Values: – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach
pGWAcqOrder-Pref	• Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM
pTdscdmaBand-Pref	• Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set. • Values: – 0x01 - TD-SCDMA Band A – 0x02 - TD-SCDMA Band B – 0x04 - TD-SCDMA Band C – 0x08 - TD-SCDMA Band D – 0x10 - TD-SCDMA Band E – 0x20 - TD-SCDMA Band F – All other bits are reserved

pAcqOrderPref	- acqOrderPref • Optional parameter for specifying Acquisition Order Preference • see acqOrderPref for more information
pSrvReg-Restriction	• Optional parameter indicating Network Selection Registration Restriction Preference • Values: – 0x00 - Device follows the normal registration process – 0x01 - Device camps on the network according to its provisioning, but does not register – 0x02 - Device selects the network for limited service – All other values are reserved.
pCSGID	- CSGID • Optional parameter for specifying CSG ID • Either of pNetSelPref or pCSGID can be set. • see CSGID for more information
pRAT	• Optional parameter Radio Access Technology order Preference • Values: – 0x04 - GSM – 0x05 - UMTS – 0x08 - LTE – 0x09 - TDSCDMA

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 – 0x00 - IMS
TransCap	• Transport Capability • Values: – 0x00 - CDMA – 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 • Values: – 0x00 - Transport layer is not registered – 0x01 - Transport layer is registered
pTransLayerInfo	• Optional parameter • 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 • Values: – 0x00 - No Service – 0x01 - In Progress – 0x02 - Failed – 0x03 - Limited Service – 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. • Values • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
samplesPer-Batch	• number of samples per batch the GNSS location engine is to receive. • samplingFrequency = samplesPerBatch ∗ batchesPerSecond • samplesPerBatch must be a nonzero positive value.
batchPerSec	• LTE NAS version minor • Number of sensor-data batches the GNSS location engine is to receive per second. • 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. • Values • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
samplesPer-Batch	• number of samples per batch the GNSS location engine is to receive. • samplingFrequency = samplesPerBatch ∗ batchesPerSecond • samplesPerBatch must be a nonzero positive value.
batchPerSec	• LTE NAS version minor • Number of sensor-data batches the GNSS location engine is to receive per second. • 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: – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA

8.51.2 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. • When the system index is not known, 0xFFFF is used.
regPrd	• Registration period after the CDMA system is acquired. • 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. • When the system index is not known, 0xFFFF is used.
cellBroadcast-Cap	• Cell broadcast capability of the serving system. • When the CDMA registration period is not valid, 0xFFFF is used. – NAS_CELL_BROADCAST_CAP_UNKNOWN - Cell broadcast support is unknown – NAS_CELL_BROADCAST_CAP_OFF - Cell broadcast is not supported – NAS_CELL_BROADCAST_CAP_ON - Cell broadcast is supported

8.54.2 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. • Range 0 to 3. • Some modems support only 1 or 2 NAMs. • 0xFF,if not available.
airTimerValue	• Time in minutes. • Cumulative air time is slammed. • 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 – 0x01 - GSM Default_Char – 0x02 - UCS2 – 0xFF - Not Available
alphaLen	• Number of sets of the following elements: – pAlpha_text • 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 • Valid values – 0 - Source is unknown – 1 - GPS is the source – 2 - Cell ID provided the source – 3 - Source is enhanced cell ID – 4 - Wi-Fi is the source – 5 - Terrestrial source – 6 - Hybrid terrestrial source – 7 - Altitude database is the source – 8 - Barometric altimeter is the source – 9 - Other sources
linkage	• Specifies the dependency between the horizontal and altitude position components • Valid values – 0 - Not specified – 1 - Fully interdependent – 2 - Depends on latitude and longitude – 3 - Fully independent
coverage	• Specifies the region of uncertainty. • Valid values – 0 - Not specified – 1 - Altitude uncertainty is valid at the injected horizontal position coordinates only – 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 • Valid values – 0 - Source is unknown – 1 - GPS is the source – 2 - Cell ID provided the source – 3 - Source is enhanced cell ID – 4 - Wi-Fi is the source – 5 - Terrestrial source – 6 - Hybrid terrestrial source – 7 - Altitude database is the source – 8 - Barometric altimeter is the source – 9 - Other sources
linkage	• Specifies the dependency between the horizontal and altitude position components • Valid values – 0 - Not specified – 1 - Fully interdependent – 2 - Depends on latitude and longitude – 3 - Fully independent
coverage	• Specifies the region of uncertainty. • Valid values – 0 - Not specified – 1 - Altitude uncertainty is valid at the injected horizontal position coordinates only – 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. – 0 - Unknown – 1 - SIM card – 2 - USIM application – 3 - RUIM card – 4 - CSIM application – 5 - ISIM application • Other values are reserved for the future and are to be handled as "Unknown".
appState	• Indicates the state of the application. – 0 - Unknown – 1 - Detected – 2 - PIN1 or UPIN is required – 3 - PUK1 or PUK for UPIN is required – 4 - Personalization state must be checked – 5 - PIN1 is blocked – 6 - Illegal – 7 - Ready
persoState	• Indicates the state of the personalization for the application. – 0 - Unknown – 1 - Personalization operation is in progress – 2 - Ready – 3 - Personalization code is required – 4 - PUK for personalization code is required – 5 - Permanently blocked

persoFeature	• Indicates the personalization feature. • This applies only when a personalization code is required to deactivate or unblock personalization. – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization – 11 - Unknown
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 • If zero(0) then no aidVal information exists.
aidVal[MAX_DESCRIPTION_LENGTH]	• Application identifier value.
univPin	• Indicates whether UPIN replaces PIN1. – 0 - PIN1 is used – 1 - UPIN replaces PIN1
pin1State	• Indicates the state of PIN1. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 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. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 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. – 0 - Unknown – 1 - SIM card – 2 - USIM application – 3 - RUIM card – 4 - CSIM application – 5 - ISIM application • Other values are reserved for the future and are to be handled as "Unknown".
appState	• Indicates the state of the application. – 0 - Unknown – 1 - Detected – 2 - PIN1 or UPIN is required – 3 - PUK1 or PUK for UPIN is required – 4 - Personalization state must be checked – 5 - PIN1 is blocked – 6 - Illegal – 7 - Ready
persoState	• Indicates the state of the personalization for the application. – 0 - Unknown – 1 - Personalization operation is in progress – 2 - Ready – 3 - Personalization code is required – 4 - PUK for personalization code is required – 5 - Permanently blocked

persoFeature	• Indicates the personalization feature. • This applies only when a personalization code is required to deactivate or unblock personalization. – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization – 11 - Unknown
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 • If zero(0) then no aidVal information exists.
aidVal[MAX_DESCRIPTION_LENGTH]	• Application identifier value.
univPin	• Indicates whether UPIN replaces PIN1. – 0 - PIN1 is used – 1 - UPIN replaces PIN1
pin1State	• Indicates the state of PIN1. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 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. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 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. • 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. – 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1 – 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2 – 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3 – 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5 – 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6 – 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7 – 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8 – 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. • 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. – 0x00 - ALERTING_LOCAL - Local – 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. • If zero(0) then no further information exists.
allCallsAlphaID-Info[MAX_NO_-OF_CALLS]	• Array of allCallsAlphaIDInfo. • 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. • If zero(0) then no further information exists.
CalledPartyNum[MAX_NO_OF_CALLS]	• Array of CalledPartyNum. • 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. • 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 . • 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. • If zero(0) then no further information exists.
getAllCallInfo[MAX_NO_OF_CALLS]	• Array of CallInfo. • 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. • If zero(0) then no further information exists.

ConnectedParty-Num[MAX_NO_-OF_CALLS]	• Array of ConnectedPartyNum. • 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. • If zero(0) then no further information exists.
DiagInfo[MAX_-NO_OF_CALLS]	• Array of DiagInfo. • 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. • If zero(0) then no further information exists.
RedirPartyNum[MAX_NO_OF_CALLS]	• Array of RedirPartyNum. • 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. • If zero(0) then no further information exists.
GetAllCallRmtPtyName[MAX_NO_OF_CALLS]	• Array of remotePartyName. • 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. • If zero(0) then no further information exists.
RmtPtyNum[MAX_NO_OF_CALLS]	• Array of remotePartyNum. • 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. • 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. • 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. • If zero(0) then no further information exists.
AllCallsUUSInfo[MAX_NO_OF_CALLS]	• Array of allCallsUUSInfo. • 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. • 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. – 0 - Runs the GSM algorithm (valid only on a 2GSIM card) – 1 - Runs the CAVE algorithm (valid only on a RUIM card) – 2 - GSM security context (valid only on a USIM application) – 3 - 3G security context (valid only on a USIM application) – 4 - VGCS/VBS security context (valid only on aUSIM application) – 5 - GBA security context, Bootstrapping mode (valid only on a USIM or ISIM application) – 6 - GBA security context, NAF Derivation mode (valid only on a USIM or ISIM application) – 7 - MBMS security context, MSK Update mode (valid only on a USIM application) – 8 - MBMS security context, MTK Generation mode (valid only on a USIM application) – 9 - MBMS security context, MSK Deletion mode (valid only on a USIM application) – 10 - MBMS security context, MUK Deletion mode (valid only on a USIM application) – 11 - IMS AKA security context (valid only on aISIM application) – 12 - HTTP-digest security context (valid only onan ISIM application) – 13 - Compute IP authentication, CHAP (valid onlyon RUIM or CSIM) – 14 - Compute IP authentication, MN-HA authenticator (valid only on RUIM or CSIM) – 15 - Compute IP authentication, MIP-RRQ hash (valid only on RUIM or CSIM) – 16 - Compute IP authentication, MN-AAA authenticator (valid only on RUIM or CSIM) – 17 - Compute IP authentication, HRPD access authenticator (valid only on RUIM or CSIM) • 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 • Bit 0 - Band class 0, A-system • Bit 1 - Band class 0, B-system • Bit 2 - Band class 1, all blocks • Bit 3 - Band class 2 • Bit 4 - Band class 3, A-system • Bit 5 - Band class 4, all blocks • Bit 6 - Band class 5, all blocks • Bit 7 - GSM DCS band (1800) • Bit 8 - GSM Extended GSM (E-GSM) band (900) • Bit 9 - GSM Primary GSM (P-GSM) band (900) • Bit 10 - Band class 6 • Bit 11 - Band class 7 • Bit 12 - Band class 8 • Bit 13 - Band class 9 • Bit 14 - Band class 10 • Bit 15 - Band class 11 • Bit 16 - GSM 450 band • Bit 17 - GSM 480 band • Bit 18 - GSM 750 band • Bit 19 - GSM 850 band • Bit 20 - GSM railways GSM band (900) • Bit 21 - GSM PCS band (1900) • Bit 22 - WCDMA (Europe, Japan, and China) 2100 band • Bit 23 - WCDMA US PCS 1900 band • Bit 24 - WCDMA (Europe and China) DCS 1800 band • Bit 25 - WCDMA US 1700 band • Bit 26 - WCDMA US 850 band • Bit 27 - WCDMA Japan 800 band • Bit 28 - Band class 12 • Bit 29 - Band class 14 • Bit 30 - Reserved • Bit 31 - Band class 15 • Bits 32 through 47 - Reserved • Bit 48 - WCDMA Europe 2600 band • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved • Bit 56 - Band class 16 • Bit 57 - Band class 17 • Bit 58 - Band class 18 • Bit 59 - Band class 19

OUT]	pLteBandCapability[OUT] Bitmask of LTE bands supported by the device • Bit 0 - LTE EUTRAN Band 1 UL:1920-1980; DL: 2110-2170 • Bit 1 - LTE EUTRAN Band 2 UL:1850-1910; DL: 1930-1990 • Bit 2 - LTE EUTRAN Band 3 UL:1710-1785; DL: 1805-1880 • Bit 3 - LTE EUTRAN Band 4 UL:1710-1755; DL: 2110-2155 • Bit 4 - LTE EUTRAN Band 5 UL: 824-849; DL: 869-894 • Bit 5 - LTE EUTRAN Band 6 UL: 830-840; DL: 875-885 • Bit 6 - LTE EUTRAN Band 7 UL:2500-2570; DL: 2620-2690 • Bit 7 - LTE EUTRAN Band 8 UL: 880-915; DL: 925-960 • Bit 8 - LTE EUTRAN Band 9 UL:1749.9-1784.9; DL: 1844.9-1879.9 • Bit 9 - LTE EUTRAN Band 10 UL:1710-1770; DL: 2110-2170 • Bit 10 - LTE EUTRAN Band 11 UL:1427.9-1452.9; DL: 1475.9-1500.9 • Bit 11 - LTE EUTRAN Band 12 UL:698-716; DL: 728-746 • Bit 12 - LTE EUTRAN Band 13 UL: 777-787; DL: 746-756 • Bit 13 - LTE EUTRAN Band 14 UL: 788-798; DL: 758-768 • Bits 14 and 15 - Reserved • Bit 16 - LTE EUTRAN Band 17 UL: 704-716; DL: 734-746 • Bit 17 - LTE EUTRAN Band 18 UL: 815-830; DL: 860-875 • Bit 18 - LTE EUTRAN Band 19 UL: 830-845; DL: 875-890 • Bit 19 - LTE EUTRAN Band 20 UL: 832-862; DL: 791-821 • Bit 20 - LTE EUTRAN Band 21 UL: 1447.9-1462.9; DL: 1495.9-1510.9 • Bit 21 - Reserved • Bit 22 - LTE EUTRAN Band 23 UL: 2000-2020; DL: 2180-2200 • Bit 23 - LTE EUTRAN Band 24 UL: 1626.5-1660.5; DL: 1525-1559 • Bit 24 - LTE EUTRAN Band 25 UL: 1850-1915; DL: 1930-1995 • Bit 25 - LTE EUTRAN Band 26 UL: 814-849; DL: 859-894 • Bit 26 - Reserved • Bit 27 - LTE EUTRAN Band 28 UL: 703-748; DL: 758-803 • Bit 28 - LTE EUTRAN Band 29 UL: 1850-1910 or 1710-1755; DL: 716-728 • Bits 29 through 31 - Reserved • Bit 32 - LTE EUTRAN Band 33 UL: 1900-1920; DL: 1900-1920 • Bit 33 - LTE EUTRAN Band 34 UL: 2010-2025; DL: 2010-2025 • Bit 34 - LTE EUTRAN Band 35 UL: 1850-1910; DL: 1850-1910 • Bit 35 - LTE EUTRAN Band 36 UL: 1930-1990; DL: 1930-1990 • Bit 36 - LTE EUTRAN Band 37 UL: 1910-1930; DL: 1910-1930 • Bit 37 - LTE EUTRAN Band 38 UL: 2570-2620; DL: 2570-2620 • Bit 38 - LTE EUTRAN Band 39 UL: 1880-1920; DL: 1880-1920 • Bit 39 - LTE EUTRAN Band 40 UL: 2300-2400; DL: 2300-2400 • Bit 40 - LTE EUTRAN Band 41 UL: 2496-2690; DL: 2496-2690 • Bit 41 - LTE EUTRAN Band 42 UL: 3400-3600; DL: 3400-3600 • Bit 42 - LTE EUTRAN Band 43 UL: 3600-3800; DL: 3600-3800 • Bits 43 through 64 - Reserved

OUT]	pTdsBandCapability[OUT] Bitmask of TDS bands supported by the device. • Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz • Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz • Bit 2 - TDS Band C 1910 to 1930 MHz • Bit 3 - TDS Band D 2570 to 2620 MHz • Bit 4 - TDS Band E 2300 to 2400 MHz • Bit 5 - TDS Band F 1880 to 1920 MHz

8.79.2 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. – Range for BDS: 201 to 237
mask	• Indicates if the ephemeris or almanac for a satellite is to be deleted • Valid values: – QMI_LOC_MASK_DELETE_EPHEMERIS (0x01) - Delete ephemeris for the satellite – 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: – gnssSvId – 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 – 0x00 – Not selected – 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. • 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. • 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. – NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only – NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only – NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type – NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types – NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown
psBarStatus	• Call barring status for packet-switched calls. – NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only – NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only – NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type – NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types – NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown

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

8.85.2 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 – Caller Id

SI	• Number screening indicator. • Values: – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
numType	• Number type. • Values: – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
numPlan	• Number plan. • Values: – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 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. – 0x00 - NSAP – 0x01 - USER
oddEvenInd	• Even/odd indicator. – 0x00 - Even number of address signals – 0x01 - Odd number of address signals
subAddrLen	• Number of sets of the following elements: – 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 – 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 – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION Reserved extension
numberPlan	• Call forwarding number plan – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS Reserved cordless telephony system – 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: – 0x00 - SERVICE_STATUS_INACTIVE - Inactive – 0x01 - SERVICE_STATUS_ACTIVE - Active
SvcClass	• Service Class is a combination (sum) of information class constants • See qaGobiApiTableSupServiceInfoClasses.h for service classes.
noReplyTimer	• No reply timer value in seconds • 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. • Values: – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
numType	• Number type. • Values: – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
numPlan	• Number plan. • Values: – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 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: – 0x00 - SERVICE_STATUS_INACTIVE - Inactive – 0x01 - SERVICE_STATUS_ACTIVE - Active
SvcClass	• Service Class is a combination (sum) of information class constants • 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 • 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. • If zero(0) then invalid.
callState	• Call state. – 0x01 - CALL_STATE_ORIGINATION - Origination – 0x02 - CALL_STATE_INCOMING - Incoming – 0x03 - CALL_STATE_CONVERSATION - Conversation – 0x04 - CALL_STATE_CC_IN_PROGRESS - Call is originating but waiting for call control to complete – 0x05 - CALL_STATE_ALERTING - Alerting – 0x06 - CALL_STATE_HOLD - Hold – 0x07 - CALL_STATE_WAITING - Waiting – 0x08 - CALL_STATE_DISCONNECTING - Disconnecting – 0x09 - CALL_STATE_END - End – 0x0A - CALL_STATE_SETUP - MT call is in Setup state in 3GPP – 0xFF - Not Available
callType	• Call type. – 0x00 - CALL_TYPE_VOICE - Voice – 0x02 - CALL_TYPE_VOICE_IP - Voice over IP – 0x06 - CALL_TYPE_OTAPA - OTAPA – 0x07 - CALL_TYPE_STD_OTASP - Standard OTASP – 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP – 0x09 - CALL_TYPE_EMERGENCY - Emergency – 0xFF - Not Available
direction	• Direction. – 0x01 - CALL_DIRECTION_MO - MO call – 0x02 - CALL_DIRECTION_MT - MT call – 0xFF - Not Available
mode	• Mode. • 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. – 0x01 - CALL_MODE_CDMA - CDMA – 0x02 - CALL_MODE_GSM - GSM – 0x03 - CALL_MODE_UMTS - UMTS – 0x04 - CALL_MODE_LTE - LTE – 0x05 - CALL_MODE_TDS - TD-SCDMA – 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 – Caller Id
SI	• Number screening indicator. • Values: – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network

numType	• Number type. • Values: – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
numPlan	• Number plan. • Values: – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 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. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.
index1xPri	• Index of the primary 1X provisioning application. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.

indexGwSec	• Index of the secondary GW provisioning application. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.
index1xSec	• Index of the secondary GW provisioning application. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.
numSlot	• Indicates the number of slots available on the device. • The following block is repeated for each slot. i.e. cardState • If zero(0) then no cardState information exists.
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 • Values – 0 - Device – 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: • 0x01 – End proactive session command type received from the card • 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 • 0x00000001 – User Activity Notify • 0x00000002 – Idle Screen Available • 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 • 0x01 – Refresh start • 0x02 – Refresh success • 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. – 0x01 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ACTIVATE - Activate – 0x02 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_DEACTIVATE - Deactivate – 0x03 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER - Register – 0x04 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ERASE - Erase – 0x05 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_INTERROGATE - Interrogate – 0x06 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER_PASSWORD - Register password – 0x07 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_USSD - USSD – 0xFF - Not Available
reason	• Call control supplementary service result reason • Values: – 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 – 0x00 – Not selected – 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. – 0xFFFF - Not Available
priChB	• B Channel number for the primary carrier. – 0xFFFF - Not Available
secChA	• A Channel number for the secondary carrier. – 0xFFFF - Not Available
secChB	• B Channel number for the secondary carrier. – 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) • 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. – 0xFFFF - Not Available

nid	• Network ID. – 0xFFFF - Not Available
baseId	• Base station ID. – 0xFFFF - Not Available
refpn	• Reference PN. – 0xFFFF - Not Available
baseLat	• Latitude of the current base station in units of 0.25 sec. – 0xFFFFFFFF - Not Available
baseLong	• Longitude of the current base station in units of 0.25 sec. – 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) • Returns the priority setting of the message 0x00 - normal 0x01 - interactive 0x02 - urgent 0x03 - emergency 0xFF - unavailable setting
pPrivacy[OUT](optional	parameter) • Returns the privacy setting of the message 0x00 - not restricted 0x01 - restricted 0x02 - confidential 0x03 - secret 0xFF - unavailable setting
pLanguage[OUT]	(optional parameter) • 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) • 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) • Returns the absolute validity period setting for this message.This field takes the same form as mcTimeStamp
pRelativeValidity[OUT]	(optional parameter) • 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) • Returns the display mode parameter 0x00 - immediate display 0x01 - mobile default setting 0x02 - user invoked 0x03 - reserved 0xFF - unavailable parameter
pUserAcknowledgementReq[OUT]	(optional parameter) • 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) • 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) • 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) • returns the length of Callback address string (including the NULL termination)
pCallbkAddr[OUT]	(optional parameter) • 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 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

8.110.1 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) • 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) • 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) • 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.

8.110.2 Field Documentation

8.110.2.1 BYTE cdmaMsgEncodingParams::messageId

8.110.2.2 CHAR∗ cdmaMsgEncodingParams::pCallbackAddr

8.110.2.3 CHAR∗ cdmaMsgEncodingParams::pDestAddr

8.110.2.4 BYTE∗ cdmaMsgEncodingParams::pEncodingAlphabet

8.110.2.5 BYTE∗ cdmaMsgEncodingParams::pMessage

8.110.2.6 BYTE∗ cdmaMsgEncodingParams::pMessageSize

8.110.2.7 BYTE∗ cdmaMsgEncodingParams::pPriority

8.110.2.8 BYTE* cdmaMsgEncodingParams::pRelValidity

8.110.2.9 WORD* cdmaMsgEncodingParams::pTextMsg

8.110.2.10 ULONG cdmaMsgEncodingParams::textMsgLength

8.111 CDMARSSIThresh Struct Reference

Data Fields

- BYTE CDMARSSIThreshListLen
- WORD * pCDMARSSIThreshList

8.111.1 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) • maximum of 32 values. • Range for RSSI values:-105 to -21 (in dBm).

8.111.2 Field Documentation

8.111.2.1 BYTE CDMARSSIThresh::CDMARSSIThreshListLen

8.111.2.2 WORD* CDMARSSIThresh::pCDMARSSIThreshList

8.112 CDMASSInfo Struct Reference

Data Fields

- INT8 rssi
- SHORT ecio

8.112.1 Detailed Description

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

Parameters

rssi	• RSSI in dBm (signed value). • 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).

8.112.2 Field Documentation

8.112.2.1 SHORT CDMASSInfo::ecio

8.112.2.2 INT8 CDMASSInfo::rssi

8.113 cdmaSSInfo Struct Reference

Data Fields

- int8_t rssi
- int16_t ecio

8.113.1 Detailed Description

Parameters

rssi	RSSI in dBm.
ecio	ECIO value representing negative 0.5 dBm increment

8.113.2 Field Documentation

8.113.2.1 int16_t cdmaSSInfo::ecio

8.113.2.2 int8_t cdmaSSInfo::rssi

8.114 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]

8.114.1 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. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
isSysPrlMatch	• Indicates whether the system is in a PRL. • Only applies to CDMA/HDR. – 0x00 - System is not in a PRL – 0x01 - System is in a PRL – 0xFF - Not Available • If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL. • If the system is in a PRL, roam_status is set to the value based on the standard specification.
pRevInUseValid	• Indicates whether the P_Rev in use is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
pRevInUse	• The lesser of the base station P_Rev and mobile P_Rev • Only applicable for CDMA. – 0xFF - Not Available
bsPRevValid	• Indicates whether the base station P_Rev is valid – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
bsPRev	• Base station P_Rev. • Only applicable for CDMA. – 0xFF - Not Available

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

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

8.114.2 Field Documentation

8.114.2.1 WORD CDMASysInfo::baseId

8.114.2.2 ULONG CDMASysInfo::baseLat

8.114.2.3 ULONG CDMASysInfo::baseLong

8.114.2.4 BYTE CDMASysInfo::bsInfoValid

8.114.2.5 BYTE CDMASysInfo::bsPRev

8.114.2.6 BYTE CDMASysInfo::bsPRevValid

8.114.2.7 BYTE CDMASysInfo::ccsSupported

8.114.2.8 BYTE CDMASysInfo::ccsSupportedValid

8.114.2.9 BYTE CDMASysInfo::cdmaSysIdValid

8.114.2.10 BYTE CDMASysInfo::isSysPrlMatch

8.114.2.11 BYTE CDMASysInfo::isSysPrlMatchValid

8.114.2.12 BYTE CDMASysInfo::MCC[3]

8.114.2.13 BYTE CDMASysInfo::MNC[3]

8.114.2.14 WORD CDMASysInfo::networkID

8.114.2.15 BYTE CDMASysInfo::networkIdValid

8.114.2.16 WORD CDMASysInfo::packetZone

8.114.2.17 BYTE CDMASysInfo::packetZoneValid

8.114.2.18 BYTE CDMASysInfo::pRevInUse

8.114.2.19 BYTE CDMASysInfo::pRevInUseValid

8.114.2.20 sysInfoCommon CDMASysInfo::sysInfoCDMA

8.114.2.21 WORD CDMASysInfo::systemID

8.115 CDMASysInfoExt Struct Reference

Data Fields

- WORD MCC
- BYTE imsi_11_12

8.115.1 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

8.115.2 Field Documentation

8.115.2.1 BYTE CDMASysInfoExt::imsi_11_12

8.115.2.2 WORD CDMASysInfoExt::MCC

8.116 CellDb Struct Reference

Data Fields

- ULONG mask

8.116.1 Detailed Description

This structure contains the cell database

Parameters

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

8.116.2 Field Documentation

8.116.2.1 ULONG CellDb::mask

8.117 cellParams Struct Reference

Data Fields

- WORD pci
- SHORT rsrq
- SHORT rsrp
- SHORT rssi
- SHORT srxlev

8.117.1 Detailed Description

This structure contains information about the Cell parameters.

Parameters

pci	• Physical cell ID. • Range: 0 to 503.
rsrq	• Current RSRQ in 1/10 dB as measured by L1. • Range: -20.0 dB to -3.0 dB.
rsrp	• Current RSRP in 1/10 dBm as measured by L1. • Range: -140.0 dBm to -44.0 dBm.
rssi	• Current RSSI in 1/10 dBm as measured by L1. • Range: -120.0 dBm to 0.

srxlev	• Cell selection Rx level (Srxlev) value. • Range: -128 to 128. • This field is only valid when ue_in_idle is TRUE.

8.117.2 Field Documentation

8.117.2.1 WORD cellParams::pci

8.117.2.2 SHORT cellParams::rsrp

8.117.2.3 SHORT cellParams::rsrq

8.117.2.4 SHORT cellParams::rssi

8.117.2.5 SHORT cellParams::srxlev

8.118 changeUIMPIN Struct Reference

Data Fields

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

8.118.1 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. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN – 4 - Hidden key
oldPINLen	• Length of the following elements i.e. old pin value.
oldPINVal[MAX_DESCRIPTION_LENGTH]	• Old PIN value. • 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. • This value is a sequence of ASCII characters.

8.118.2 Field Documentation

8.118.2.1 BYTE changeUIMPIN::oldPINLen

8.118.2.2 BYTE changeUIMPIN::oldPINVal[255]

8.118.2.3 BYTE changeUIMPIN::pinID

8.118.2.4 BYTE changeUIMPIN::pinLen

8.118.2.5 BYTE changeUIMPIN::pinVal[255]

8.119 channelRate Struct Reference

Data Fields

- ULONG CurrChanTxRate
- ULONG CurrChanRxRate

8.119.1 Detailed Description

This structure contains Channel Rate

Parameters

CurrChanTxRate	• Max channel Tx rate in bits per second
CurrChanRxRate	• Max channel Rx rate in bits per second

8.119.2 Field Documentation

8.119.2.1 ULONG channelRate::CurrChanRxRate

8.119.2.2 ULONG channelRate::CurrChanTxRate

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 CLIPResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.121.1 Detailed Description

This structure contains information about the Calling Line Identification Presentation (CLIP) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available

ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.121.2 Field Documentation

8.121.2.1 BYTE CLIPResp::ActiveStatus

8.121.2.2 BYTE CLIPResp::ProvisionStatus

8.122 CLIRResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.122.1 Detailed Description

This structure contains information about the Calling Line Identification Restriction (CLIR) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED_PERMANENT - Permanently provisioned – 0x02 - PROVISION_STATUS_PRESENTATION_RESTRICTED - Restricted presentation – 0x03 - PROVISION_STATUS_PRESENTATION_ALLOWED - Allowed presentation – 0xFF - Not Available

8.122.2 Field Documentation

8.122.2.1 BYTE CLIRResp::ActiveStatus

8.122.2.2 BYTE CLIRResp::ProvisionStatus

8.123 ClkInfo Struct Reference

Data Fields

- ULONG mask

8.123.1 Detailed Description

This structure contains the clock Info

Parameters

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

8.123.2 Field Documentation

8.123.2.1 ULONG ClkInfo::mask

8.124 CNAPResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.124.1 Detailed Description

This structure contains information about the Calling Name Presentation (CNAP) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.124.2 Field Documentation

8.124.2.1 BYTE CNAPResp::ActiveStatus

8.124.2.2 BYTE CNAPResp::ProvisionStatus

8.125 COLPResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.125.1 Detailed Description

This structure contains information about the Connected Line Identification Presentation (COLP) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.125.2 Field Documentation

8.125.2.1 BYTE COLPResp::ActiveStatus

8.125.2.2 BYTE COLPResp::ProvisionStatus

8.126 COLRResp Struct Reference

Data Fields

- BYTE ActiveStatus
- BYTE ProvisionStatus

8.126.1 Detailed Description

This structure contains information about the Connected Line Identification Restriction (COLR) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.126.2 Field Documentation

8.126.2.1 BYTE COLRResp::ActiveStatus

8.126.2.2 BYTE COLRResp::ProvisionStatus

8.127 CommInfo Struct Reference

Data Fields

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

8.127.1 Detailed Description

Structure for storing the common information for the device.

Parameters

temperature	• Temperature. – 8-bit signed integer – 0xFF - Not Available.
modemMode	• Modem Operating Mode. – 0x00 - POWERING OFF – 0x01 - FACTORY TEST – 0x02 - OFFLINE – 0x03 - OFFLINE_AMPS – 0x04 - OFFLINE_CDMA – 0x05 - ONLINE – 0x06 - LOW POWER MODE – 0x07 - RESETTING – 0x08 - NETWORK TEST – 0x09 - OFFLINE REQUEST – 0x0A - PSEUDO ONLINE – 0x0B - RESETTING MODEM – 0xFF - Unknown

systemMode	• System Acquisition Mode. – 0x00 - No service – 0x01 - AMPS – 0x02 - CDMA – 0x03 - GSM – 0x04 - HDR – 0x05 - WCDMA – 0x06 - GPS – 0x08 - WLAN – 0x09 - LTE – 0xFF - Unknown
imsRegState	• IMS Registration State. – 0x00 - NO SRV – 0x01 - IN PROG – 0x02 - FAILED – 0x03 - LIMITED – 0x04 - FULL SRV – 0xFF - Unknown
psState	• PS Attach State. – 0x00 - Attached – 0x01 - Detached – 0xFF - Unknown

8.127.2 Field Documentation

8.127.2.1 BYTE CommInfo::imsRegState

8.127.2.2 BYTE CommInfo::modemMode

8.127.2.3 BYTE CommInfo::psState

8.127.2.4 BYTE CommInfo::systemMode

8.127.2.5 BYTE CommInfo::temperature

8.128 ConnectionStatus Struct Reference

Data Fields

- BYTE MDMConnStatus
- ULONGLONG MDMCallDuration

8.128.1 Detailed Description

This structure contains modem connection status

Parameters

MDMConn-Status	• Current link status – 0x01 - DISCONNECTED – 0x02 - CONNECTED
MDMCall-Duration	• Call duration in milliseconds. • If the modem connection status is connected, this represent the duration of the current DUN call. • 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).

8.128.2 Field Documentation

8.128.2.1 ULONGLONG ConnectionStatus::MDMCallDuration

8.128.2.2 BYTE ConnectionStatus::MDMConnStatus

8.129 connectionStatus Struct Reference

Data Fields

- uint8_t MDMConnStatus
- uint64_t MDMCallDuration

8.129.1 Detailed Description

Parameters

MDMConn-Status	Current link status 1-Disconected 2-Connected
MDMCall-Duration	Call duration in milliseconds

8.129.2 Field Documentation

8.129.2.1 uint64_t connectionStatus::MDMCallDuration

8.129.2.2 uint8_t connectionStatus::MDMConnStatus

8.130 connectNumInfo Struct Reference

Data Fields

- BYTE numPresInd

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

8.130.1 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 – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
screeningInd	• Screening indicator. – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network – 0xFF - Not Available
numType	• Number type. – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension – 0xFF - Not Available

numPlan	• Number plan. – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension – 0xFF - Not Available
callerIDLen	• Provides the length of caller ID which follow. • If zero(0) then no further information exists.
callerID[MAX_CALL_NO_LEN]	• callerID of numLen length, NULL terminated.

8.130.2 Field Documentation

8.130.2.1 BYTE connectNumInfo::callerID[81]

8.130.2.2 BYTE connectNumInfo::callerIDLen

8.130.2.3 BYTE connectNumInfo::numPlan

8.130.2.4 BYTE connectNumInfo::numPresInd

8.130.2.5 BYTE connectNumInfo::numType

8.130.2.6 BYTE connectNumInfo::screeningInd

8.131 CrashInfo Struct Reference

Data Fields

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

8.131.1 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

8.131.2 Field Documentation

8.131.2.1 ULONG CrashInfo::crashData

8.131.2.2 ULONG CrashInfo::crashId

8.131.2.3 WORD CrashInfo::crashStrLen

8.131.2.4 WORD CrashInfo::gcDumpStrLen

8.131.2.5 WORD CrashInfo::numCrashes

8.131.2.6 CHAR* CrashInfo::pCrashString

8.131.2.7 CHAR* CrashInfo::pGCDumpString

8.132 crashInfoParams Struct Reference

Data Fields

- uint8_t crashStatus
- crashInformation crashInfo

8.132.1 Detailed Description

This structure contains crashInfoParams

Parameters

Tlvresult	• Unpack Result
crashInfoParam[-OUT]	• See crashInfoParams

8.132.2 Field Documentation

8.132.2.1 crashInformation crashInfoParams::crashInfo

8.132.2.2 uint8_t crashInfoParams::crashStatus

8.133 CrashInfoParams Struct Reference

Data Fields

- BYTE * pDevCrashStatus
- CrashInfo * pCrashInfo

8.133.1 Detailed Description

This structure is used to store Crash Information

Parameters

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

8.133.2 Field Documentation

8.133.2.1 CrashInfo* CrashInfoParams::pCrashInfo

8.133.2.2 BYTE* CrashInfoParams::pDevCrashStatus

8.134 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]

8.134.1 Detailed Description

This structure contains crashInformation

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

8.134.2 Field Documentation

8.134.2.1 uint32_t crashInformation::crashData

8.134.2.2 uint32_t crashInformation::crashId

8.134.2.3 char crashInformation::crashString[255]

8.134.2.4 uint16_t crashInformation::crashStrlen

8.134.2.5 char crashInformation::gcdumpString[1024]

8.134.2.6 uint16_t crashInformation::gcdumpStrlen

8.134.2.7 uint16_t crashInformation::numCrashes

8.135 CreateProfileIn Struct Reference

Data Fields

- BYTE * pProfileID

- BYTE * pProfileType
- QmiProfileInfo curProfile

8.135.1 Detailed Description

This structure contains the input parameters for SLQSCreateProfile

Parameters

ProfileID	• 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
ProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2 – 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

8.135.2 Field Documentation

8.135.2.1 QmiProfileInfo CreateProfileIn::curProfile

8.135.2.2 BYTE* CreateProfileIn::pProfileID

8.135.2.3 BYTE* CreateProfileIn::pProfileType

8.136 CreateProfileOut Struct Reference

Data Fields

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

8.136.1 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

8.136.2 Field Documentation

8.136.2.1 USHORT* CreateProfileOut::pExtErrorCode

8.136.2.2 BYTE* CreateProfileOut::pProfileIndex

8.136.2.3 BYTE* CreateProfileOut::pProfileType

8.137 CsgId Struct Reference

Data Fields

- ULONG csgId
- BYTE TlvPresent

8.137.1 Detailed Description

This structure contains csg ID.

Parameters

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

8.137.2 Field Documentation

8.137.2.1 ULONG CsgId::csgId

8.137.2.2 BYTE CsgId::TlvPresent

8.138 CSGID Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- BYTE mncPcsDigits

- ULONG id
- BYTE rat

8.138.1 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 • 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: – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE – 0x09 - RADIO_IF_TDSCDMA - TDS

8.138.2 Field Documentation

8.138.2.1 ULONG CSGID::id

8.138.2.2 WORD CSGID::mcc

8.138.2.3 WORD CSGID::mnc

8.138.2.4 BYTE CSGID::mncPcsDigits

8.138.2.5 BYTE CSGID::rat

8.139 CUGInfo Struct Reference

Data Fields

- WORD CUGIndex
- BYTE SuppPrefCUG
- BYTE SuppOA

8.139.1 Detailed Description

This structure contains Closed User Group Information

Parameters

CUGIndex	• Range 0x00... 0x7FFF
SuppPrefCUG	• Suppress preferential CUG – 0x00 - FALSE – 0x01 - TRUE
SuppOA	• Suppress OA subscription option – 0x00 - FALSE – 0x01 - TRUE

8.139.2 Field Documentation

8.139.2.1 WORD CUGInfo::CUGIndex

8.139.2.2 BYTE CUGInfo::SuppOA

8.139.2.3 BYTE CUGInfo::SuppPrefCUG

8.140 curAMRConfig Struct Reference

Data Fields

- BYTE gsmAmrStat
- BYTE wcdmaAmrStat

8.140.1 Detailed Description

This structure contains the Current Adaptive Multi Rate Configuration Information.

Parameters

gsmAmrStat	• GSM AMR Status – 0x00 - Disable – 0x01 - Enable – 0xFF - Not Available
wcdmaAmrStat	• WCDMA AMR Status • One or a combination of the following bitmask values: – Bit 0 - AMR codec advertised is not supported – Bit 1 - Controls WCDMA AMR wideband – Bit 2 - Controls GSM half rate AMR – Bit 3 - Controls GSM AMR wideband – Bit 4 - Controls GSM AMR narrowband • 0xFF, if not available

8.140.2 Field Documentation

8.140.2.1 BYTE curAMRConfig::gsmAmrStat

8.140.2.2 BYTE curAMRConfig::wcdmaAmrStat

8.141 CurrDataSysStat Struct Reference

Data Fields

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

8.141.1 Detailed Description

Data System Status

Parameters

pPrefNetwork	[OUT] • Preferred Network • Values: – 0 - 3GPP – 1 - 3GPP2
pNetworkInfoLen	[IN/OUT] • As input parameter size assigned to next parameter i.e. network information • As output the actual number of network information elements returned by the device
pCurrNetwork-Info	[OUT] • Network information • See CurrNetworkInfo for more details

8.141.2 Field Documentation

8.141.2.1 CurrNetworkInfo* CurrDataSysStat::pCurrNetworkInfo

8.141.2.2 BYTE* CurrDataSysStat::pNetworkInfoLen

8.141.2.3 BYTE* CurrDataSysStat::pPrefNetwork

8.142 currentCatEvent Union Reference

Data Fields

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

8.142.1 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

8.142.2 Field Documentation

8.142.2.1 struct CatAlPhaIdentifierTlv currentCatEvent::CatAlphaIdtfr

8.142.2.2 struct CatEndProactiveSessionTlv currentCatEvent::CatEndPS

8.142.2.3 struct CatEventListTlv currentCatEvent::CatEventLst

8.142.2.4 struct CatEventIDDataTlv currentCatEvent::CatEvIDData

8.142.2.5 struct CatRefreshTlv currentCatEvent::CatRefresh

8.143 CurrentImgList Struct Reference

Data Fields

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

8.143.1 Detailed Description

This structure is used to store image list

Parameters

numEntries[IN/-OUT]	• Number of entries in the image list to follow • 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

8.143.2 Field Documentation

8.143.2.1 CHAR CurrentImgList::carrier[16]

8.143.2.2 CHAR CurrentImgList::fwvers[16]

8.143.2.3 BYTE CurrentImgList::numEntries

8.143.2.4 CurrImageInfo* CurrentImgList::pCurrImgInfo

8.143.2.5 CHAR CurrentImgList::pkgver[16]

8.143.2.6 CHAR CurrentImgList::priver[16]

8.144 currentPLMN Struct Reference

Data Fields

- WORD MCC

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

8.144.1 Detailed Description

This structure contains the current PLMN parameters

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

Parameters

MCC	• mobile country code – A 16 bit representation of MCC – Range 0 to 999
MNC	• mobile network code – A 16 bit representation of MNC – Range 0 to 999
netDescrLength	• Length of Network description field • Defaults to zero
netDescr	• Network Description – optional string containing network name or description

8.144.2 Field Documentation

8.144.2.1 WORD currentPLMN::MCC

8.144.2.2 WORD currentPLMN::MNC

8.144.2.3 BYTE currentPLMN::netDescr[255]

8.144.2.4 BYTE currentPLMN::netDescrLength

8.145 CurrImageInfo Struct Reference

Data Fields

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

8.145.1 Detailed Description

This structure is used to store image information

Parameters

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

8.145.2 Field Documentation

8.145.2.1 BYTE CurrImageInfo::buildID[255]

8.145.2.2 BYTE CurrImageInfo::buildIDLen

8.145.2.3 BYTE CurrImageInfo::imageType

8.145.2.4 BYTE CurrImageInfo::uniqueID[16]

8.146 CurrNetworkInfo Struct Reference

Data Fields

- BYTE NetworkType
- ULONG RATMask
- ULONG SOMask

8.146.1 Detailed Description

Network information structure

Parameters

NetworkType	• Values: – 0 - 3GPP – 1 - 3GPP2

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

8.146.2 Field Documentation

8.146.2.1 BYTE CurrNetworkInfo::NetworkType

8.146.2.2 ULONG CurrNetworkInfo::RATMask

8.146.2.3 ULONG CurrNetworkInfo::SOMask

8.147 currNetworkInfo Struct Reference

Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

8.147.1 Detailed Description

Parameters

NetworkType	network type 0-3GPP 1-3GPP2
RATMask	Radio access technology mask to indicate technology.
SOMask	Service Option mask to indicate the service option.

8.147.2 Field Documentation

8.147.2.1 uint8_t currNetworkInfo::NetworkType

8.147.2.2 uint32_t currNetworkInfo::RATMask

8.147.2.3 uint32_t currNetworkInfo::SOMask

8.148 custFeaturesInfo Struct Reference

Data Fields

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

8.148.1 Detailed Description

This structure contains current settings of custom features

Parameters

GpsEnable[OUT]	• describes if GPS is enabled or disabled • values: – 0x00 - GPS is disabled – 0x01 - GPS is enabled • function SLQSGetCustFeatures() returns a default value FFFFFFFF if no value is returned by the modem
pDisableIMSI[OUT]	• optional 1 byte parameter • describes if IMSI display is enabled or disabled • values: – 0x00 - Allow display of IMSI – 0x01 - Do not display IMSI • function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem
pIPFamSupport[OUT]	• optional 2 byte BitMask • bitmask representing the IP families supported • values: – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - IPv4v6 • function SLQSGetCustFeatures() returns a default value FFFF if no value is returned by the modem
pRMAutoConnect[OUT]	• optional 1 byte parameter • QMI Mode RM Net Auto Connect Support • values: – 0x00 - Not Supported – 0x01 - Supported • function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem
pGPSSel[OUT]	• optional 1 byte parameter • GPS Antenna Select • values: – 0x00 - Dedicated GPS Port – 0x01 - GPS Rx over AUX Port – 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied • function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem

pSMSSupport[-OUT]	• optional 1 byte parameter • SMS support • values: – 0x00 - Not supported – 0x01 - supported • Used to determine whether or not to hide SMS from user • 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 • Voice support • values: – 0x00 - Enable voice on both AT and QMI interface (default) – 0x01 - Reserved – 0x02 - Disable voice on both AT and QMI interface
pDHCPRelay-Enabled[OUT]	• optional 1 byte parameter • DHCP Relay support • values: – 0x00 - Disable DHCP relay – 0x01 - Enable DHCP relay
pGPSLPM[OUT]	• optional 1 byte parameter • GPSLPM support • values: – 0x00 - Enable GPS in Low Power Mode – 0x01 - Disable GPS in Low Power Mode

8.148.2 Field Documentation

8.148.2.1 ULONG custFeaturesInfo::GpsEnable

8.148.2.2 BYTE* custFeaturesInfo::pDHCPRelayEnabled

8.148.2.3 BYTE* custFeaturesInfo::pDisableIMSI

8.148.2.4 BYTE* custFeaturesInfo::pGPSLPM

8.148.2.5 BYTE* custFeaturesInfo::pGPSSel

8.148.2.6 WORD* custFeaturesInfo::pIPFamSupport

8.148.2.7 BYTE* custFeaturesInfo::pIsVoiceEnabled

8.148.2.8 BYTE* custFeaturesInfo::pRMAutoConnect

8.148.2.9 BYTE* custFeaturesInfo::pSMSSupport

8.149 custFeaturesSetting Struct Reference

Data Fields

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

8.149.1 Detailed Description

This structure contains settings to be used for custom features

Parameters

pGPSSel	• optional 1 byte parameter • GPS Antenna Select • values: – 0x00 - Dedicated GPS Port – 0x01 - GPS Rx over AUX Port – 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied
pGPSEnable	• optional 4 byte parameter • GPS Enable/Disable • values: The value of 7 least significant bits: – 0 - Disabled – 1 - MT & MO enabled – 2 - MO enabled – 3 - MT enabled – 4 - MT & MO enabled if GPS_DISABLE pin is not asserted – 5 - MO GPS enabled if GPS_DISABLE pin is not asserted – 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 • voice enabled/disabled • values: – 0 - Enable voice on both AT and QMI interface (default) – 1 - Reserved – 2 - Disable voice on both AT and QMI interface

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

8.149.2 Field Documentation

8.149.2.1 BYTE* custFeaturesSetting::pDHCPRelayEnabled

8.149.2.2 ULONG* custFeaturesSetting::pGPSEnable

8.149.2.3 BYTE* custFeaturesSetting::pGPSLPM

8.149.2.4 BYTE* custFeaturesSetting::pGPSSel

8.149.2.5 BYTE* custFeaturesSetting::pIsVoiceEnabled

8.150 custSettingInfo Struct Reference

Data Fields

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

8.150.1 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 – bit 0: Values: * 0 - read only * 1 - read/write

8.150.2 Field Documentation

8.150.2.1 WORD custSettingInfo::cust_attr

8.150.2.2 CHAR custSettingInfo::cust_id[64+1]

8.150.2.3 BYTE custSettingInfo::cust_value[8+1]

8.150.2.4 WORD custSettingInfo::id_length

8.150.2.5 WORD custSettingInfo::value_length

8.151 custSettingList Struct Reference

Data Fields

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

8.151.1 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

8.151.2 Field Documentation

8.151.2.1 custSettingInfo custSettingList::custSetting[256]

8.151.2.2 BYTE custSettingList::list_type

8.151.2.3 WORD custSettingList::num_instances

8.152 dataBearers Struct Reference

Data Fields

- BYTE dataBearerMask
- QmiWDSDataBearerTechnology * pCurDataBearerTechnology
- QmiWDSDataBearerTechnology * pLastCallDataBearerTechnology

8.152.1 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. – NULL if the parameter is not required
pLastCallDataBearerTechnology[OUT]	• last call data bearer technology value. – NULL if the parameter is not required

8.152.2 Field Documentation

8.152.2.1 BYTE dataBearers::dataBearerMask

8.152.2.2 QmiWDSDataBearerTechnology* dataBearers::pCurDataBearerTechnology

8.152.2.3 QmiWDSDataBearerTechnology* dataBearers::pLastCallDataBearerTechnology

8.153 DataBearerTech Struct Reference

Data Fields

- ULONG techType
- ULONG ratValue
- ULONGLONG soMask

8.153.1 Detailed Description

Network information structure

Parameters

TechType	• Technology type • Values: – 0 - WDS_BEARER_TECH_NETWORK_3GPP - 3GPP – 1 - WDS_BEARER_TECH_NETWORK_3GPP2 - 3GPP2
ratValue	• Radio Access Technology (RAT) value • Values: – 0x00 - WDS_BEARER_TECH_RAT_EX_NULL_BEARER - NULL bearer – 0x01 - WDS_BEARER_TECH_RAT_EX_3GPP_WCDMA - 3GPP WCDMA – 0x02 - WDS_BEARER_TECH_RAT_EX_3GPP_GERAN - 3GPP GERAN – 0x03 - WDS_BEARER_TECH_RAT_EX_3GPP_LTE - 3GPP LTE – 0x04 - WDS_BEARER_TECH_RAT_EX_3GPP_TDSCDMA - 3GPP TDSCDMA – 0x05 - WDS_BEARER_TECH_RAT_EX_3GPP_WLAN - 3GPP WLAN – 0x64 - WDS_BEARER_TECH_RAT_EX_3GPP_MAX - 3GPP maximum – 0x65 - WDS_BEARER_TECH_RAT_EX_3GPP2_1X - 3GPP2 1X – 0x66 - WDS_BEARER_TECH_RAT_EX_3GPP2_HRPD - 3GPP2 HRPD – 0x67 - WDS_BEARER_TECH_RAT_EX_3GPP2_EHRPD - 3GPP2 EHRPD – 0x68 - WDS_BEARER_TECH_RAT_EX_3GPP2_WLAN - 3GPP2 WLAN – 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. • Values: – 0x00 - SO mask unspecified • 3GPP SO mask: – 0x01 - WCDMA – 0x02 - HSDPA – 0x04 - HSUPA – 0x08 - HSDPAPLUS – 0x10 - DC HSDPAPLUS – 0x20 - 64 QAM – 0x40 - HSPA – 0x80 - GPRS – 0x100 - EDGE – 0x200 - GSM – 0x400 - S2B – 0x800 - LTE limited service – 0x1000 - LTE FDD – 0x2000 - LTE TDD • 3GPP2 SO mask: – 0x01000000 - 1X IS95 – 0x02000000 - 1X IS2000 – 0x04000000 - 1X IS2000 REL A – 0x08000000 - HDR REV0 DPA – 0x10000000 - HDR REVA DPA – 0x20000000 - HDR REVB DPA – 0x40000000 - HDR REVA MPA – 0x80000000 - HDR REVB MPA – 0x100000000 - HDR REVA EMPA – 0x200000000 - HDR REVB EMPA – 0x400000000 - HDR REVB MMPA – 0x800000000 - HDR EVDO FMC

8.153.2 Field Documentation

8.153.2.1 ULONG DataBearerTech::ratValue

8.153.2.2 ULONGLONG DataBearerTech::soMask

8.153.2.3 ULONG DataBearerTech::techType

8.154 DataBearerTechExt Struct Reference

Data Fields

- DataBearerTech * pBearerTech
- DataBearerTech * pLastBearerTech

8.154.1 Detailed Description

Data Bearer Technology Ext

Parameters

pBearerTech	[OUT] • See DataBearerTech for more information
pLastBearerTech	[OUT] • See DataBearerTech for more information

8.154.2 Field Documentation

8.154.2.1 DataBearerTech* DataBearerTechExt::pBearerTech

8.154.2.2 DataBearerTech* DataBearerTechExt::pLastBearerTech

8.155 dataBearerTechnology Struct Reference

Data Fields

- BYTE currentNetwork
- ULONG ratMask
- ULONG soMask

8.155.1 Detailed Description

Structure to hold the current data bearer technology values

Parameters

pCurrent-Network[OUT]	• current selected network – 0 - UNKNOWN – 1 - 3GPP2 – 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) – 0x8000 - NULL Bearer – 0x0000 - DO_NOT_CARE CDMA RAT mask – 0x01 - CDMA_1X – 0x02 - EVDO_REV0 – 0x04 - EVDO_REVA UMTS RAT mask – 0x01 - WCDMA – 0x02 - GPRS – 0x04 - HSDPA – 0x08 - HSUPA – 0x10 - EDGE – 0x20 - LTE – 0x40 - HSDPA+ – 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) – 0x00 - DO_NOT_CARE CDMA 1X SO mask – 0x01 - CDMA_1X_IS95 – 0x02 - CDMA_1X_IS2000 – 0x04 - CDMA_1X_IS2000_REL_A CDMA EV-DO Rev A SO mask – 0x01 - EVDO_REVA_DPA – 0x02 - EVDO_REVA_MFPA – 0x04 - EVDO_REVA_EMPA – 0x08 - EVDO_REVA_EMPA_EHRPD

8.155.2 Field Documentation

8.155.2.1 BYTE dataBearerTechnology::currentNetwork

8.155.2.2 ULONG dataBearerTechnology::ratMask

8.155.2.3 ULONG dataBearerTechnology::soMask

8.156 dataRate Struct Reference

Data Fields

- ULONG dataRateMax
- ULONG guaranteedRate

8.156.1 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)

8.156.2 Field Documentation

8.156.2.1 ULONG dataRate::dataRateMax

8.156.2.2 ULONG dataRate::guaranteedRate

8.157 dataSrvCapabilities Struct Reference

Data Fields

- BYTE dataCapabilitiesLen
- BYTE dataCapabilities [0x20]

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

8.157.2 Field Documentation

8.157.2.1 BYTE dataSrvCapabilities::dataCapabilities[0x20]

8.157.2.2 BYTE dataSrvCapabilities::dataCapabilitiesLen

8.158 DataStatusDetail Struct Reference

Data Fields

- ULONG IPAddress
- BYTE LastErrCode

8.158.1 Detailed Description

This structure contains Data Status Details

Parameters

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

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

8.158.2 Field Documentation

8.158.2.1 ULONG DataStatusDetail::IPAddress

8.158.2.2 BYTE DataStatusDetail::LastErrCode

8.159 DataULongLongTlv Struct Reference

Data Fields

- BYTE TlvPresent
- ULONGLONG ullData

8.159.1 Field Documentation

8.159.1.1 BYTE DataULongLongTlv::TlvPresent

8.159.1.2 ULONGLONG DataULongLongTlv::ullData

8.160 DataULongTlv Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG ulData

8.160.1 Field Documentation

8.160.1.1 BYTE DataULongTlv::TlvPresent

8.160.1.2 ULONG DataULongTlv::ulData

8.161 DcsUsbPortNames Struct Reference

Data Fields

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

8.161.1 Field Documentation

8.161.1.1 CHAR DcsUsbPortNames::AtCmdPort[32]

8.161.1.2 CHAR DcsUsbPortNames::DmPort[32]

8.161.1.3 CHAR DcsUsbPortNames::NmeaPort[32]

8.162 delAssistDataStatus Struct Reference

Data Fields

- ULONG status

8.162.1 Detailed Description

Contain the parameters passed for SetLocDeleteAssistDataCallback by the device.

Parameters

status	• Status of the Delete Assist Data request • Valid values: – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure

Note

None

8.162.2 Field Documentation

8.162.2.1 ULONG delAssistDataStatus::status

8.163 depersonalizationInformation Struct Reference

Data Fields

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

8.163.1 Detailed Description

This structure contains the Depersonalization Information.

Parameters

feature	• Indicates the personalization feature to de-activate or unblock. – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization
operation	• Indicates the operation to perform. – 0 - Deactivate personalization. – 1 - Unblock personalization.
ckLen	• Length of the following elements i.e. control key value.
ckVal[MAX_DESCRIPTION_LENGTH]	• Control key value. • This value is a sequence of ASCII characters.

8.163.2 Field Documentation

8.163.2.1 BYTE depersonalizationInformation::ckLen

8.163.2.2 BYTE depersonalizationInformation::ckVal[255]

8.163.2.3 BYTE depersonalizationInformation::feature

8.163.2.4 BYTE depersonalizationInformation::operation

8.164 detailSvcInfo Struct Reference

Data Fields

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

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

8.164.2 Field Documentation

8.164.2.1 BYTE detailSvcInfo::hdrHybrid

8.164.2.2 BYTE detailSvcInfo::hdrSrvStatus

8.164.2.3 BYTE detailSvcInfo::isSysForbidden

8.164.2.4 BYTE detailSvcInfo::srvCapability

8.164.2.5 BYTE detailSvcInfo::srvStatus

8.165 DeviceConfigDetail Struct Reference

Data Fields

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

8.165.1 Detailed Description

This structure contains Device Configuration Details

Parameters

Technology	• Current technology in use – 0x00 - 1x RTT – 0x01 - EVDO Rev 0 – 0x02 - EVDO Rev A – 0xFF - NA
QLIC	• Quasi Linear Interference Cancellation – 0x00 - Not supported – 0x01 - Supported
Chipset	• Qualcomm platform – 0x4E - MDM6200 – 0x4F - MDM6600 – 0xFF - NA
HWVersion	• Hardware version – 0x00 - BSHWREV0 – 0x01 - BSHWREV1 – 0x02 - BSHWREV2 – 0x03 - BSHWREV3 – 0x04 - BSHWREVMAX – 0xFF - BSHWREVUNKNOWN

8.165.2 Field Documentation

8.165.2.1 BYTE DeviceConfigDetail::Chipset

8.165.2.2 BYTE DeviceConfigDetail::HWVersion

8.165.2.3 BYTE DeviceConfigDetail::QLIC

8.165.2.4 BYTE DeviceConfigDetail::Technology

8.166 DHCPOption Struct Reference

Data Fields

- BYTE optCode
- BYTE optValLen
- BYTE * pOptVal

8.166.1 Detailed Description

This structure contains DHCPv4 lease option values

Parameters

optCode	• Values – Option code
optValLen	• Values – Option value length
pOptValue	• Val – Option value

8.166.2 Field Documentation

8.166.2.1 BYTE DHCPOption::optCode

8.166.2.2 BYTE DHCPOption::optValLen

8.166.2.3 BYTE* DHCPOption::pOptVal

8.167 DHCPOptionList Struct Reference

Data Fields

- BYTE numOpt
- DHCPOption * pOptions

8.167.1 Detailed Description

This structure contains DHCPv4 lease option list

Parameters

optListSize	• Values – Size of Option List
pOptions	• Values – Options

8.167.2 Field Documentation

8.167.2.1 BYTE DHCPOptionList::numOpt

8.167.2.2 DHCPOption* DHCPOptionList::pOptions

8.168 diagInfo Struct Reference

Data Fields

- BYTE diagInfoLen
- BYTE diagnosticInfo [255]

8.168.1 Detailed Description

This structure contains Diagnostic Information

Parameters

diagInfoLen	• Provides the length of information which follow. • If zero(0) then no further information exists.
diagnosticInfo[MAX_DESCRIPTION_LENGTH]	• Diagnostic information.

8.168.2 Field Documentation

8.168.2.1 BYTE diagInfo::diagInfoLen

8.168.2.2 BYTE diagInfo::diagnosticInfo[255]

8.169 dirNum Struct Reference

Data Fields

- BYTE dirNumLen
- BYTE dirNum [255]

8.169.1 Detailed Description

This structure contains the parameters for Directory Number Information

Parameters

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

8.169.2 Field Documentation

8.169.2.1 BYTE dirNum::dirNum[255]

8.169.2.2 BYTE dirNum::dirNumLen

8.170 dms_ActivationStatusTlv Struct Reference

Data Fields

- uint16_t TlvPresent
- uint32_t activationStatus

8.170.1 Detailed Description

Activation Status Tlv

Parameters

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

8.170.2 Field Documentation

8.170.2.1 uint32_t dms_ActivationStatusTlv::activationStatus

8.170.2.2 uint16_t dms_ActivationStatusTlv::TlvPresent

8.171 dms_OperatingModeTlv Struct Reference

Data Fields

- uint16_t TlvPresent
- uint32_t operatingMode

8.171.1 Detailed Description

Operating Mode Tlv

Parameters

operatingMode	• 0 - Online • 1 - Low power • 2 - Factory test mode • 3 - Offline • 4 - Resetting • 5 - Shutting down • 6 - Persistent low power • 7 - Mode-only low power

8.171.2 Field Documentation

8.171.2.1 uint32_t dms_OperatingModeTlv::operatingMode

8.171.2.2 uint16_t dms_OperatingModeTlv::TlvPresent

8.172 dmsCurrentPRLInfo Struct Reference

Data Fields

- WORD * pPRLVersion
- BYTE * pPRLPreference

8.172.1 Detailed Description

This structure contains GetCurrentPRLInfo response parameter

Parameters

pPRLVersion[OUT]	- Optional • PRL version of device.
pPRLPreference	[OUT]- Optional • PRL Preference – 0 - Unset – 1 - Set

8.172.2 Field Documentation

8.172.2.1 BYTE* dmsCurrentPRLInfo::pPRLPreference

8.172.2.2 WORD* dmsCurrentPRLInfo::pPRLVersion

8.173 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

8.173.1 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 – bit 0: Values: * 0 - read only * 1 - read/write

8.173.2 Field Documentation

8.173.2.1 uint16_t DMScustSettingInfo::cust_attr

8.173.2.2 uint8_t DMScustSettingInfo::cust_id[64+1]

8.173.2.3 uint8_t DMScustSettingInfo::cust_value[8+1]

8.173.2.4 uint16_t DMScustSettingInfo::id_length

8.173.2.5 uint16_t DMScustSettingInfo::value_length

8.174 DMScustSettingList Struct Reference

Data Fields

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

8.174.1 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

8.174.2 Field Documentation

8.174.2.1 DMScustSettingInfo DMScustSettingList::custSetting[255+1]

8.174.2.2 uint8_t DMScustSettingList::list_type

8.174.2.3 uint16_t DMScustSettingList::num_instances

8.175 DMSgetCustomFeatureV2 Struct Reference

Data Fields

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

8.175.1 Detailed Description

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

Parameters

IN]	pGetCustomInput • See getCustomInput for more information
OUT]	pCustSettingInfo • See custSettingInfo for more information

OUT]	pCustSettingList • See custSettingList for more information

8.175.2 Field Documentation

8.175.2.1 DMScustSettingInfo∗ DMSgetCustomFeatureV2::pCustSettingInfo

8.175.2.2 DMScustSettingList∗ DMSgetCustomFeatureV2::pCustSettingList

8.175.2.3 DMSgetCustomInput∗ DMSgetCustomFeatureV2::pGetCustomInput

8.176 DMSgetCustomInput Struct Reference

Data Fields

- uint8_t cust_id [64+1]
- uint8_t list_type

8.176.1 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

8.176.2 Field Documentation

8.176.2.1 uint8_t DMSgetCustomInput::cust_id[64+1]

8.176.2.2 uint8_t DMSgetCustomInput::list_type

8.177 dmsIndicationRegisterReq Struct Reference

Data Fields

- BYTE ∗ pSwiGetResetInd

8.177.1 Detailed Description

This structure contains the SLQSDmsSwiIndicationRegister request parameters.

Parameters

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

Note

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

8.177.2 Field Documentation

8.177.2.1 BYTE* dmsIndicationRegisterReq::pSwiGetResetInd

8.178 dmsSwiGetResetInfo Struct Reference

Data Fields

- BYTE type
- BYTE source

8.178.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

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

8.178.2 Field Documentation

8.178.2.1 BYTE dmsSwiGetResetInfo::source

8.178.2.2 BYTE dmsSwiGetResetInfo::type

8.179 Domain Struct Reference

Data Fields

- WORD domainLen
- CHAR domainName [256]

8.179.1 Detailed Description

This structure contains the DomainName Information

Parameters

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

8.179.2 Field Documentation

8.179.2.1 WORD Domain::domainLen

8.179.2.2 CHAR Domain::domainName[256]

8.180 DomainNameList Struct Reference

Data Fields

- BYTE numInstances
- struct Domain domain [10]

8.180.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

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

8.180.2 Field Documentation

8.180.2.1 struct Domain DomainNameList::domain[10]

8.180.2.2 BYTE DomainNameList::numInstances

8.181 DRCParams Struct Reference

Data Fields

- BYTE DRCValue
- BYTE DRCCover

8.181.1 Detailed Description

This structure contains Data Rate Channel parameters

Parameters

DRCValue	• Current Data Rate Channel
DRCCover	• Current Data Rate Channel cover

8.181.2 Field Documentation

8.181.2.1 BYTE DRCParams::DRCCover

8.181.2.2 BYTE DRCParams::DRCValue

8.182 DTMFInfo Struct Reference

Data Fields

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

8.182.1 Detailed Description

This structure contains information about the DTMF (Dual Tone Multi-Frequency).

Parameters

callID	• Call identifier for the current call.
DTMFEvent	• DTMF event – 0x00 - DTMF_EVENT_REV_BURST - Sends a CDMA-burst DTMF – 0x01 - DTMF_EVENT_REV_START_CONT - Starts a continuous DTMF tone – 0x03 - DTMF_EVENT_REV_STOP_CONT - Stops a continuous DTMF tone – 0x05 - DTMF_EVENT_FWD_BURST - Received a CDMA-burst DTMF message – 0x06 - DTMF_EVENT_FWD_START_CONT - Received a start-continuous DTMF tone order – 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

8.182.2 Field Documentation

8.182.2.1 BYTE DTMFInfo::callID

8.182.2.2 BYTE DTMFInfo::digitBuff[255]

8.182.2.3 BYTE DTMFInfo::digitCnt

8.182.2.4 BYTE DTMFInfo::DTMFEvent

8.183 DTMFLengths Struct Reference

Data Fields

- BYTE DTMFPulseWidth
- BYTE DTMFInterdigitInterval

8.183.1 Detailed Description

This structure contains Voice Burst DTMF pulse length information

Parameters

DTMFPulse-Width	• DTMF pulse width. Values: – 0x00 - DTMF_ONLENGTH_95MS - 95 ms – 0x01 - DTMF_ONLENGTH_150MS - 150 ms – 0x02 - DTMF_ONLENGTH_200MS - 200 ms – 0x03 - DTMF_ONLENGTH_250MS - 250 ms – 0x04 - DTMF_ONLENGTH_300MS - 300 ms – 0x05 - DTMF_ONLENGTH_350MS - 350 ms – 0x06 - DTMF_ONLENGTH_SMS SMS Tx special pulse width
DTMFInterdigit-Interval	• DTMF interdigit interval Values: – 0x00 - DTMF_OFFLENGTH_60MS - 60 ms – 0x01 - DTMF_OFFLENGTH_100MS - 100 ms – 0x02 - DTMF_OFFLENGTH_150MS - 150 ms – 0x03 - DTMF_OFFLENGTH_200MS - 200 ms

8.183.2 Field Documentation

8.183.2.1 BYTE DTMFLengths::DTMFInterdigitInterval

8.183.2.2 BYTE DTMFLengths::DTMFPulseWidth

8.184 DUNCallInfoInd Struct Reference

Data Fields

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

8.184.1 Field Documentation

8.184.1.1 WORD DUNCallInfoInd::CallEndReason

8.184.1.2 channelRate DUNCallInfoInd::ChannelRate

8.184.1.3 BYTE DUNCallInfoInd::DataBearerTech

8.184.1.4 BYTE DUNCallInfoInd::DormancyStatus

8.184.1.5 BYTE DUNCallInfoInd::MdmConnStatus

8.184.1.6 ULONGLONG DUNCallInfoInd::RXOKBytesCount

8.184.1.7 ULONGLONG DUNCallInfoInd::TXOKBytesCount

8.185 dunchannelRate Struct Reference

Data Fields

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

8.185.1 Detailed Description

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
MaxChanRx-Rate	maximum Rx rate that can be assigned to the device

8.185.2 Field Documentation

8.185.2.1 uint32_t dunchannelRate::CurrChanRxRate

8.185.2.2 uint32_t dunchannelRate::CurrChanTxRate

8.185.2.3 uint32_t dunchannelRate::MaxChanRxRate

8.185.2.4 uint32_t dunchannelRate::MaxChanTxRate

8.186 ecioListElement Struct Reference

Data Fields

- SHORT ecio
- BYTE radioIf

8.186.1 Detailed Description

This structure contains the ECIO Information

Parameters

ecio	• ECIO value in dBm
radioIf	• Radio interface technology of the signal being measured – 0x00 - RADIO_IF_NO_SVC - None (no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000Â® 1X – 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS

8.186.2 Field Documentation

8.186.2.1 SHORT ecioListElement::ecio

8.186.2.2 BYTE ecioListElement::radioIf

8.187 ECIOThresh Struct Reference

Data Fields

- BYTE ECIOThresListLen
- SHORT * pECIOThresList

8.187.1 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 • 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 – Each ECIO threshold value is a signed 2 byte value – Each ECIO threshold value increments in negative 0.5 dB, e.g., an ECIO threshold value of 2 means -1dB – Maximum number of threshold values is 16 – At least one value must be specified – Threshold values specified above are used for all RATs

8.187.2 Field Documentation

8.187.2.1 BYTE ECIOThresh::ECIOThresListLen

8.187.2.2 SHORT* ECIOThresh::pECIOThresList

8.188 ECTNum Struct Reference

Data Fields

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

8.188.1 Detailed Description

Contains the parameters passed for Explicit Communication Transfer by the device.

Parameters

ECTCallState	• ECT call state: – 0x00 - ECT_CALL_STATE_NONE - None – 0x01 - ECT_CALL_STATE_ALERTING - Alerting – 0x02 - ECT_CALL_STATE_ACTIVE - Active
presentationInd	• Presentation indicator – 0x00 - presentationAllowedAddress – 0x01 - presentationRestricted – 0x02 - numberNotAvailable – 0x04 - presentationRestrictedAddress
number	• Number in ASCII characters terminated by NULL

8.188.2 Field Documentation

8.188.2.1 BYTE ECTNum::ECTCallState

8.188.2.2 BYTE ECTNum::number[81]

8.188.2.3 BYTE ECTNum::presentationInd

8.189 encryptedPIN1 Struct Reference

Data Fields

- BYTE pin1Len
- BYTE pin1Val [255]

8.189.1 Detailed Description

This structure contains the encrypted PIN1 Information.

Parameters

pin1Len	• Number of sets of the following elements ie encrypted PIN1 value. • 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.

8.189.2 Field Documentation

8.189.2.1 BYTE encryptedPIN1::pin1Len

8.189.2.2 BYTE encryptedPIN1::pin1Val[255]

8.190 eriDataparams Struct Reference

Data Fields

- uint16_t eriDataLen
- uint8_t eriData [1024]

8.190.1 Field Documentation

8.190.1.1 uint8_t eriDataparams::eriData[1024]

8.190.1.2 uint16_t eriDataparams::eriDataLen

8.191 ERIFileparams Struct Reference

Data Fields

- WORD * pFileSize
- BYTE * pFile

8.191.1 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. • Upon successful output, actual number of bytes written to file contents array
pFile[OUT]	• ERI data read from persistent storage(Max size is 1024)

8.191.2 Field Documentation

8.191.2.1 BYTE* ERIFileparams::pFile

8.191.2.2 WORD* ERIFileparams::pFileSize

8.192 errorRateListElement Struct Reference

Data Fields

- USHORT errorRate
- BYTE radioIf

8.192.1 Detailed Description

This structure contains the Error Rate Information

Parameters

errorRate	• Error rate value corresponds to the RAT that is currently registered. – For CDMA, the error rate reported is Frame Error Rate: * Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% * A value of 0xFFFF indicates that the error rate is unknown or unavailable – For HDR, the error rate reported is Packet Error Rate: * Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% * A value of 0xFFFF indicates that the error rate is unknown or unavailable – For GSM, the error rate reported is Bit Error Rate: * Valid values are 0, 100, 200, 300, 400, 500, 600, and 700 The reported value divided by 100 gives the error rate as an RxQual value, e.g.,a value of 300 represents an RxQual value of 3. * A value of 25500 indicates No Data – For WCDMA, the error rate reported is Block Error Rate (BLER): * Valid values are 1 to 10000 * The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3%. * A value of 0 indicates No Data
radioIf	• Radio interface technology of the signal being measured – 0x00 - RADIO_IF_NO_SVC - None (no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000Â® 1X – 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS

8.192.2 Field Documentation

8.192.2.1 USHORT errorRateListElement::errorRate

8.192.2.2 BYTE errorRateListElement::radioIf

8.193 eTWSPLMNInfoTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSEtwsPlmnInfo ETWSPLMNInfo

8.193.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
ETWSPLMNInfo	• ETWS PLMN Information • See sMSEtwsPlmnInfo for more information

8.193.2 Field Documentation

8.193.2.1 sMSEtwsPlmnInfo eTWSPLMNInfoTlv::ETWSPLMNInfo

8.193.2.2 uint8_t eTWSPLMNInfoTlv::TlvPresent

8.194 extDispRecInfo Struct Reference

Data Fields

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

8.194.1 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: – 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.

8.194.2 Field Documentation

8.194.2.1 BYTE extDispRecInfo::dispType

8.194.2.2 BYTE extDispRecInfo::extDispInfo[255]

8.194.2.3 BYTE extDispRecInfo::extDispInfoLen

8.195 FactorySequenceNumber Struct Reference

Data Fields

- BYTE FSNumber [255]

8.195.1 Detailed Description

This structure used to store Factory Sequence Number parameter

Parameters

FSNumber[OUT]	• Facorty Sequence Number • Maximum Length is 255 Bytes

8.195.2 Field Documentation

8.195.2.1 BYTE FactorySequenceNumber::FSNumber[255]

8.196 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]

8.196.1 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. – 0 - Transparent – 1 - Cyclic – 2 - Linear fixed – 3 - Dedicated file – 4 - Master file
recordSize	• Indicates the size of the records. • Only for cyclic and linear fixed files
recordCount	• Indicates the total no. of the records. • Only for linear fixed files
secRead	• Read security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secReadMask	• Mask with read security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
secWrite	• Write security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition

secWriteMask	• Mask with write security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
secIncrease	• Increase security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secIncrease-Mask	• Mask with increase security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
secDeactivate	• Deactivate security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secDeactivate-Mask	• Mask with deactivate security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
secActivate	• Activate security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition

secActivateMask	• Mask with activate security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
rawLen	• Length of the following elements i.e. raw value.
rawValue[MAX_-DESCRIPTION-_LENGTH]	• Raw value of file attributes.

8.196.2 Field Documentation

8.196.2.1 WORD fileAttributes::fileID

8.196.2.2 WORD fileAttributes::fileSize

8.196.2.3 BYTE fileAttributes::fileType

8.196.2.4 WORD fileAttributes::rawLen

8.196.2.5 BYTE fileAttributes::rawValue[255]

8.196.2.6 WORD fileAttributes::recordCount

8.196.2.7 WORD fileAttributes::recordSize

8.196.2.8 BYTE fileAttributes::secActivate

8.196.2.9 WORD fileAttributes::secActivateMask

8.196.2.10 BYTE fileAttributes::secDeactivate

8.196.2.11 WORD fileAttributes::secDeactivateMask

8.196.2.12 BYTE fileAttributes::secIncrease

8.196.2.13 WORD fileAttributes::secIncreaseMask

8.196.2.14 BYTE fileAttributes::secRead

8.196.2.15 WORD fileAttributes::secReadMask

8.196.2.16 BYTE fileAttributes::secWrite

8.196.2.17 WORD fileAttributes::secWriteMask

8.197 fileInfo Struct Reference

Data Fields

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

8.197.1 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).

8.197.2 Field Documentation

8.197.2.1 WORD fileInfo::fileID

8.197.2.2 WORD fileInfo::path[255]

8.197.2.3 BYTE fileInfo::pathLen

8.198 FirmwareUpdatStat Struct Reference

Data Fields

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

8.198.1 Detailed Description

This structure is used to store Firmware Update Status

Parameters

ResCode[OUT]	• FW Update Result Code • Values: – 0x00000001 - Successful – 0xFFFFFFFF - Unknown (due to power off reset after firmware update) – 0x100000nn - File update errors while nn will be the exact error number: * 00 - General error – 0x200000nn - NVUP update errors while nn will be the exact error number: * 00 - General error – 0x40000nnn - FOTA update agent errors while nnn will be the exact error number: * 000 $\sim$ 0FF - Insignia defined error code * 100 $\sim$ 1FF - Sierra defined error code * See qaGobiApiTableFwDldErrorCodes.h for more detailed information – 0x800000nn - FDT/SSDP reported errors while nn will be the exact error number * See qaGobiApiTableFwDldErrorCodes.h for more detailed information
pImgType[OUT]	• Optional parameter • Firmware image type that failed the update
pRefData[OUT]	• Optional parameter • Failed image reference data • This is normally the offset of the image that caused the failure
pRefStringLen[IN/OUT]	• Length of Reference String parameter to follow • As input parameter specifies length assigned to pRefString parameter • As output parameter specifies length of actual value retrieved from the device
pRefString[OUT]	• Optional parameter • 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 • As input parameter specifies length assigned to pRefString parameter • As output parameter specifies length of actual value retrieved from the device
pLogString[OUT]	• Optional parameter • Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.

8.198.2 Field Documentation

8.198.2.1 BYTE* FirmwareUpdatStat::pImgType

8.198.2.2 BYTE* FirmwareUpdatStat::pLogString

8.198.2.3 BYTE* FirmwareUpdatStat::pLogStringLen

8.198.2.4 ULONG* FirmwareUpdatStat::pRefData

8.198.2.5 BYTE* FirmwareUpdatStat::pRefString

8.198.2.6 BYTE* FirmwareUpdatStat::pRefStringLen

8.198.2.7 ULONG FirmwareUpdatStat::ResCode

8.199 FMSImageElement Struct Reference

Data Fields

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

8.199.1 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

8.199.2 Field Documentation

8.199.2.1 uint8_t FMSImageElement::buildId[100]

8.199.2.2 uint8_t FMSImageElement::buildIdLength

8.199.2.3 uint8_t FMSImageElement::imageId[16]

8.199.2.4 uint8_t FMSImageElement::imageType

8.200 FMSImageIdElement Struct Reference

Data Fields

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

8.200.1 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)

8.200.2 Field Documentation

8.200.2.1 uint8_t FMSImageIdElement::buildID[100]

8.200.2.2 uint8_t FMSImageIdElement::buildIDLength

8.200.2.3 uint8_t FMSImageIdElement::failureCount

8.200.2.4 uint8_t FMSImageIdElement::imageID[16]

8.200.2.5 uint8_t FMSImageIdElement::storageIndex

8.201 FMSImageIDEntries Struct Reference

Data Fields

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

8.201.1 Detailed Description

This structure contains the list entry Information

Parameters

imageType	• Type of image – 0 - Modem – 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) • See FMSImageIdElement

8.201.2 Field Documentation

8.201.2.1 uint8_t FMSImageIDEntries::executingImage

8.201.2.2 FMSImageIdElement FMSImageIDEntries::imageIDElement[50]

8.201.2.3 uint8_t FMSImageIDEntries::imageIDSize

8.201.2.4 uint8_t FMSImageIDEntries::imageType

8.201.2.5 uint8_t FMSImageIDEntries::maxImages

8.202 FMSImageList Struct Reference

Data Fields

- uint8_t listSize
- FMSImageIDEntries imageIDEntries [2]

8.202.1 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)

8.202.2 Field Documentation

8.202.2.1 FMSImageIDEntries FMSImageList::imageIDEntries[2]

8.202.2.2 uint8_t FMSImageList::listSize

8.203 FMSPrefImageList Struct Reference

Data Fields

- uint8_t listSize
- FMSImageElement listEntries [2]

8.203.1 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 • See FMSImageElement

8.203.2 Field Documentation

8.203.2.1 FMSImageElement FMSPrefImageList::listEntries[2]

8.203.2.2 uint8_t FMSPrefImageList::listSize

8.204 fwinfo_s Struct Reference

Data Fields

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

8.204.1 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)

8.204.2 Field Documentation

8.204.2.1 ULONG fwinfo_s::Carrier

8.204.2.2 ULONG fwinfo_s::FirmwareID

8.204.2.3 ULONG fwinfo_s::GPSCapability

8.204.2.4 ULONG fwinfo_s::Region

8.204.2.5 ULONG fwinfo_s::Technology

8.205 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]

8.205.1 Detailed Description

This structure contains information about the GERAN Network.

Parameters

cellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.

plmn[PLMN_LENGTH]	• MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when nmrCellID is not present.
lac	• Location area code. • This field is ignored when nmrCellID is not present. – 0xFFFF - Not Available
arfcn	• Absolute RF channel number. – 0xFFFF - Not Available
bsic	• Base station identity code. – 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. – 0xFFFFFFFF - Not Available
rxLev	• Serving Cell Rx measurement. • Values range between 0 and 63. • Mapped to a measured signal level: – Rxlev 0 is a signal strength less than -110 dBm – Rxlev 1 is -110 dBm to -109 dBm – Rxlev 2 is -109 dBm to -108 dBm – ... – Rxlev 62 is -49 dBm to -48 dBm – Rxlev 63 is greater than -48 dBm – 0xFFFF - Not Available
nmrInst	• Provides the number of set of instances which follow. • If 0(zero), then no information follows it.
insNmrCellInfo[MAX_DESCRIPTION_LENGTH]	• See nmrCellInfo for more information.

8.205.2 Field Documentation

8.205.2.1 WORD GERANInfo::arfcn

8.205.2.2 BYTE GERANInfo::bsic

8.205.2.3 ULONG GERANInfo::cellID

8.205.2.4 nmrCellInfo GERANInfo::insNmrCellInfo[255]

8.205.2.5 WORD GERANInfo::lac

8.205.2.6 BYTE GERANInfo::nmrInst

8.205.2.7 BYTE GERANInfo::plmn[3]

8.205.2.8 WORD GERANInfo::rxLev

8.205.2.9 ULONG GERANInfo::timingAdvance

8.206 geranInstInfo Struct Reference

Data Fields

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

8.206.1 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. • 0xFF indicates information is not present.
geranBsicBcc	• Base station identity code base station color code. • 0xFF indicates information is not present.
geranRssi	• Received signal strength indicator.

8.206.2 Field Documentation

8.206.2.1 WORD geranInstInfo::geranArfcn

8.206.2.2 BYTE geranInstInfo::geranBsicBcc

8.206.2.3 BYTE geranInstInfo::geranBsicNcc

8.206.2.4 SHORT geranInstInfo::geranRssi

8.207 getAllCallInformation Struct Reference

Data Fields

- callInfo Callinfo
- BYTE isEmpty
- BYTE ALS

8.207.1 Detailed Description

This structure contains information related to call state change.

Parameters

Callinfo	• See callInfo for more information.
isEmpty	• Multiparty indicator. – 0x00 - False – 0x01 - True
ALS	• Alternate Line Service line indicator. • Feature for supporting two different phone numbers on the same mobile device. – 0x00 - ALS_LINE1 - Line 1 (default) – 0x01 - ALS_LINE2 - Line 2

8.207.2 Field Documentation

8.207.2.1 BYTE getAllCallInformation::ALS

8.207.2.2 callInfo getAllCallInformation::Callinfo

8.207.2.3 BYTE getAllCallInformation::isEmpty

8.208 getAllCallRmtPtyName Struct Reference

Data Fields

- BYTE callID
- remotePartyName RemotePartyName

8.208.1 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.

8.208.2 Field Documentation

8.208.2.1 BYTE getAllCallRmtPtyName::callID

8.208.2.2 remotePartyName getAllCallRmtPtyName::RemotePartyName

8.209 getAllCallRmtPtyNum Struct Reference

Data Fields

- BYTE callID
- remotePartyNum RemotePartyNum

8.209.1 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.

8.209.2 Field Documentation

8.209.2.1 BYTE getAllCallRmtPtyNum::callID

8.209.2.2 remotePartyNum getAllCallRmtPtyNum::RemotePartyNum

8.210 GetAudioPathConfigReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Item

8.210.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig request parameters

Parameters

Profile	• Audio Profile – 0-9
Item	• Item – 0 - AV_EC – 1 - AV_NS – 2 - AV_TXVOL – 3 - AV_DTMFTXG – 4 - AV_CODECSTG – 5 - AV_TXPCMIIRFLTR – 6 - AV_RXPCMIIRFLTR – 7 - AV_MICGAIN – 8 - AV_RXAGC – 9 - AV_TXAGC – 10 - AV_RXAGCLIST – 11 - AV_RXAVCLIST – 12 - AV_TXAGCLIST

8.210.2 Field Documentation

8.210.2.1 BYTE GetAudioPathConfigReq::Item

8.210.2.2 BYTE GetAudioPathConfigReq::Profile

8.211 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

8.211.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig response parameters.

Parameters

pECMode	[Optional] • AV_EC – 0 - Echo cancellation off – 1 - Handset mode – 2 - Headset mode – 3 - Car kit mode – 4 - Speaker Mode
pNSEnable	[Optional] • AV_NS – 0 - Noise suppression off – 1 - Noise suppression on
pTXGain	[Optional] • AV_TXVOL – 0x0000 - 0xffff
pDTMFTXGain	[Optional] • AV_DTMFTXG – 0x0000 - 0xffff
pCodecSTGain	[Optional] • AV_CODECSTG – 0x0000 - 0xffff
pTXPCMIIRFltr	[Optional] • See TXPCMIIRFltr for more information
pRXPCMIIRFltr	[Optional] • See RXPCMIIRFltr for more information
pMICGainSelect	[Optional] • AV_MICGAIN
pRXAVCAGC-Switch	[Optional] • RX AVC/AGC Switch
pTXAVCSwitch	[Optional] • TX AVC Switch
pRXAGCList	[Optional] • See RXAGCList for more information
pRXAVCList	[Optional] • See RXAVCList for more information
pTXAGCList	[Optional] • See TXAGCList for more information

8.211.2 Field Documentation

8.211.2.1 WORD* GetAudioPathConfigResp::pCodecSTGain

8.211.2.2 WORD* GetAudioPathConfigResp::pDTMFTXGain

8.211.2.3 BYTE* GetAudioPathConfigResp::pECMode

8.211.2.4 BYTE* GetAudioPathConfigResp::pMICGainSelect

8.211.2.5 BYTE* GetAudioPathConfigResp::pNSEnable

8.211.2.6 RXAGCList* GetAudioPathConfigResp::pRXAGCList

8.211.2.7 BYTE* GetAudioPathConfigResp::pRXAVCAGCSwitch

8.211.2.8 RXAVCList* GetAudioPathConfigResp::pRXAVCList

8.211.2.9 RXPCMIIRFltr* GetAudioPathConfigResp::pRXPCMIIRFltr

8.211.2.10 TXAGCList* GetAudioPathConfigResp::pTXAGCList

8.211.2.11 BYTE* GetAudioPathConfigResp::pTXAVCSwitch

8.211.2.12 WORD* GetAudioPathConfigResp::pTXGain

8.211.2.13 TXPCMIIRFltr* GetAudioPathConfigResp::pTXPCMIIRFltr

8.212 GetAudioProfileReq Struct Reference

Data Fields

- BYTE Generator

8.212.1 Detailed Description

This structure contains the SLQSGetAudioProfile request parameters

Parameters

Generator	• Audio Generator – 0 - Voice – 1 - Key Beep – 2 - MIDI

8.212.2 Field Documentation

8.212.2.1 BYTE GetAudioProfileReq::Generator

8.213 GetAudioProfileResp Struct Reference

Data Fields

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

8.213.1 Detailed Description

This structure contains the SLQSGetAudioProfile response parameters.

Parameters

Profile	• Audio Profile – 0 - Handset – 1 - Headset – 2 - Car Kit – 3 - Speaker phone – 4 - Auxiliary – 5 - TTY – 6 - Auxiliary external PCM – 7 - Primary external PCM – 8 - External slave PCM – 9 - I2S
EarMute	• Ear Mute Setting – 0 - unmuted – 1 - muted
MicMute	• MIC Mute Setting – 0 - unmuted – 1 - muted
Volume	• Audio Volume Level – 0 to 7

8.213.2 Field Documentation

8.213.2.1 BYTE GetAudioProfileResp::EarMute

8.213.2.2 BYTE GetAudioProfileResp::MicMute

8.213.2.3 BYTE GetAudioProfileResp::Profile

8.213.2.4 BYTE GetAudioProfileResp::Volume

8.214 GetAudioVolTLBConfigReq Struct Reference

Data Fields

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

8.214.1 Detailed Description

This structure contains the SLQSGetAudioVolTLBConfig request parameters

Parameters

Profile	• Audio Profile – 0-9
Generator	• Audio Generator – 0-2
Volume	• Audio Volume Level – 0-7
Item	• Item – 13 - AV_RXVOLDB – 14 - AV_DTMFVOLDB – 15 - AV_PAD

8.214.2 Field Documentation

8.214.2.1 BYTE GetAudioVolTLBConfigReq::Generator

8.214.2.2 BYTE GetAudioVolTLBConfigReq::Item

8.214.2.3 BYTE GetAudioVolTLBConfigReq::Profile

8.214.2.4 BYTE GetAudioVolTLBConfigReq::Volume

8.215 GetAudioVolTLBConfigResp Struct Reference

Data Fields

- WORD ResCode

8.215.1 Detailed Description

This structure contains the SLQSGetAudioVolTLBConfig response parameters.

Parameters

ResCode	• Result of requested item

8.215.2 Field Documentation

8.215.2.1 WORD GetAudioVolTLBConfigResp::ResCode

8.216 getCallFWExtInfo Struct Reference

Data Fields

- BYTE numInstances
- callFWExtInfo CallFWExtInfo [20]

8.216.1 Detailed Description

This structure contains an array of Call Forwarded Extended Information.

Parameters

numInstances	• Number of callFWExtInfo that follow. • If zero(0) then no further information exists.
CallFWExtInfo[-MAX_NO_OF_-CALLS]	• Array of CallFWExtInfo. – See CallFWExtInfo for more information.

8.216.2 Field Documentation

8.216.2.1 callFWExtInfo getCallFWExtInfo::CallFWExtInfo[20]

8.216.2.2 BYTE getCallFWExtInfo::numInstances

8.217 getCallFWInfo Struct Reference

Data Fields

- BYTE numInstances
- callFWInfo CallFWInfo [20]

8.217.1 Detailed Description

This structure contains an array of Call Forwarded Information.

Parameters

numInstances	• Number of callFWInfo that follow. • If zero(0) then no further information exists.
CallFWInfo[MAX_NO_OF_CALLS]	• Array of callFWInfo. – See callFWInfo for more information.

8.217.2 Field Documentation

8.217.2.1 callFWInfo getCallFWInfo::CallFWInfo[20]

8.217.2.2 BYTE getCallFWInfo::numInstances

8.218 getCustomFeatureV2 Struct Reference

Data Fields

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

8.218.1 Detailed Description

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

Parameters

IN]	pGetCustomInput[IN] • Optional parameter • See getCustomInput for more information
OUT]	pCustSettingInfo[OUT] • Optional parameter • See custSettingInfo for more information
OUT]	pCustSettingList[OUT] • Optional parameter • See custSettingList for more information

8.218.2 Field Documentation

8.218.2.1 custSettingInfo* getCustomFeatureV2::pCustSettingInfo

8.218.2.2 custSettingList* getCustomFeatureV2::pCustSettingList

8.218.2.3 getCustomInput* getCustomFeatureV2::pGetCustomInput

8.219 getCustomInput Struct Reference

Data Fields

- CHAR cust_id [64+1]
- BYTE list_type

8.219.1 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

8.219.2 Field Documentation

8.219.2.1 CHAR getCustomInput::cust_id[64+1]

8.219.2.2 BYTE getCustomInput::list_type

8.220 getDUNCallInfoReq Struct Reference

Data Fields

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

8.220.1 Detailed Description

This structure contains the DUN Call Info Request parameters.

Parameters

Mask	• Mandatory parameter • Set the bits corresponding to the information requested to 1 • All other bits must be set to 0. • If any values are not available or applicable, the corresponding TLVs are not returned in the response. – Bit 0 - Connection Status – Bit 1 - Last call end reason – Bit 2 - Tx/Rx bytes OK – Bit 3 - Dormancy status – Bit 4 - Data bearer – Bit 5 - Channel rate – Bit 6 - Call active duration
pReportConn-Status	• Connect Status Indicator – 0 - Do not report – 1 - Report connection status and call end reason
pTransferStatInd	• See TransferStatInd for more information
pReportDorm-Status	• Dormancy Status Indicator – 0 - Do not report – 1 - Report traffic channel state of interface used for data connection
pReportData-BearerTech	• Current Data Bearer Technology Indicator – 0 - Do not report – 1 - Report radio interface used for data transfer when it changes
pReport-ChannelRate	• Channel Rate Indicator – 0 - Do not report – 1 - Report channel rate

8.220.2 Field Documentation

8.220.2.1 ULONG getDUNCallInfoReq::Mask

8.220.2.2 BYTE* getDUNCallInfoReq::pReportChannelRate

8.220.2.3 BYTE* getDUNCallInfoReq::pReportConnStatus

8.220.2.4 BYTE* getDUNCallInfoReq::pReportDataBearerTech

8.220.2.5 BYTE* getDUNCallInfoReq::pReportDormStatus

8.220.2.6 TransferStatInd* getDUNCallInfoReq::pTransferStatInd

8.221 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

8.221.1 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 • See qaGobiApiTableCallEndReasons.h for Call End Reason • Only valid if the last call made was DUN, else zero is returned
pTXOKBytes-Count	• Number of bytes transmitted without error • Returned only if a data call is up
pRXOKBytes-Count	• Number of bytes received without error • Returned only if a data call is up
pDormancy-Status	• Current traffic channel status • Returned if a data call is up – 0x01 - Traffic channel dormant – 0x02 - Traffic channel active

pDataBearer-Tech	• Current data bearer technology • Returned only if a data call is up – 0x01 - cdma2000 1X – 0x02 - cdma2000 HRPD (1xEV-DO) – 0x03 - GSM – 0x04 - UMTS – 0x05 - cdma200 HRPD (1xEV-DO RevA) – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - cdma2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 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). • 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). • 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 • If the modem connection status is connected, this represents the active duration of the current DUN call • 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 • Returned only if not in a call and when the previous call was made using DUN – 0x01 - cdma2000 1X – 0x02 - cdma2000 HRPD (1xEV-DO) – 0x03 - GSM – 0x04 - UMTS – 0x05 - cdma200 HRPD (1xEV-DO Rev A) – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - cdma2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 0xFF - Unknown

8.221.2 Field Documentation

8.221.2.1 WORD* getDUNCallInfoResp::pCallEndReason

8.221.2.2 ChannelRate* getDUNCallInfoResp::pChannelRate

8.221.2.3 ConnectionStatus* getDUNCallInfoResp::pConnectionStatus

8.221.2.4 BYTE* getDUNCallInfoResp::pDataBearerTech

8.221.2.5 BYTE* getDUNCallInfoResp::pDormancyStatus

8.221.2.6 BYTE* getDUNCallInfoResp::pLastCallDataBearerTech

8.221.2.7 ULONGLONG* getDUNCallInfoResp::pLastCallRXOKBytesCnt

8.221.2.8 ULONGLONG* getDUNCallInfoResp::pLastCallTXOKBytesCnt

8.221.2.9 ULONGLONG* getDUNCallInfoResp::pMdmCallDurationActive

8.221.2.10 ULONGLONG* getDUNCallInfoResp::pRXOKBytesCount

8.221.2.11 ULONGLONG* getDUNCallInfoResp::pTXOKBytesCount

8.222 getDyingGaspCfg Struct Reference

Data Fields

- BYTE * pDestSMSNum
- BYTE * pDestSMSContent

8.222.1 Detailed Description

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

Parameters

OUT]	pDestSMSNum[OUT] • SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars. • Optional parameter.
OUT]	pDestSMSContent[OUT] • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars. • Optional parameter.

8.222.2 Field Documentation

8.222.2.1 BYTE* getDyingGaspCfg::pDestSMSContent

8.222.2.2 BYTE* getDyingGaspCfg::pDestSMSNum

8.223 getDyingGaspStatistics Struct Reference

Data Fields

- ULONG * pTimeStamp
- BYTE * pSMSAttemptedFlag

8.223.1 Detailed Description

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

Parameters

OUT]	pDestSMSNum[OUT] • SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars. • Optional parameter.
OUT]	pDestSMSContent[OUT] • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars. • Optional parameter.

8.223.2 Field Documentation

8.223.2.1 BYTE* getDyingGaspStatistics::pSMSAttemptedFlag

8.223.2.2 ULONG* getDyingGaspStatistics::pTimeStamp

8.224 GetErrRateResp Struct Reference

Data Fields

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

8.224.1 Detailed Description

This structure contains information about the SLQSGetErrorRate response parameters.

Parameters

pCDMAFrame-ErrRate[Out]	• CDMA Frame Error Rate • 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%. • A value of 0xFFFF indicates that the error rate is unknown/unavailable.
pHDRPackErr-Rate[Out]	• HDR Packet Error Rate • 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%. • A value of 0xFFFF indicates that the error rate is unknown/unavailable.
pGSMBER[Out]	• GSM Bit Error Rate • Valid error rate values between 1 and 100 are returned to indicate the percentage value. • A 0% block error rate (BLER) indicates No Data.
pWCDMABER[-Out]	• WCDMA Block Error Rate • Valid error rate values between 1 and 100 are returned to indicate the percentage value. • A value of 0xFF indicates that the error rate is unknown/unavailable.

8.224.2 Field Documentation

8.224.2.1 WORD* GetErrRateResp::pCDMAFrameErrRate

8.224.2.2 BYTE* GetErrRateResp::pGSMBER

8.224.2.3 WORD* GetErrRateResp::pHDRPackErrRate

8.224.2.4 BYTE* GetErrRateResp::pWCDMABER

8.225 GetHRPDStatsResp Struct Reference

Data Fields

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

8.225.1 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 • Size must be 16 bytes
pPilotSetData[-Out]	• See PilotSetData for more information.

8.225.2 Field Documentation

8.225.2.1 DRCParams* GetHRPDStatsResp::pDRCParams

8.225.2.2 PilotSetData* GetHRPDStatsResp::pPilotSetData

8.225.2.3 BYTE* GetHRPDStatsResp::pUATI

8.226 GetIMSSMSConfigParams Struct Reference

Data Fields

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

8.226.1 Detailed Description

This structure contains the SLQSGetIMSSMSConfig response parameters.

Parameters

pSettingResp	• Settings Response
pSMSFormat	• SMS format – 0 - 3GPP – 1 - 3GPP2
pSMSOverIPNwInd	• SMS over IP Network Indication Flag – TRUE - Turn on mobile-originated SMS – FALSE - Turn off mobile-originated SMS
pPhoneCtxtURILen[IN/OUT]	• Size in bytes assigned to the Phone context Universal Resource Identifier to follow
pPhoneCtxtURI	• Phone context universal resource identifier • Length of this string must be specified in pPhoneCtxtURILen parameter

8.226.2 Field Documentation

8.226.2.1 BYTE* GetIMSSMSConfigParams::pPhoneCtxtURI

8.226.2.2 BYTE* GetIMSSMSConfigParams::pPhoneCtxtURILen

8.226.2.3 BYTE* GetIMSSMSConfigParams::pSettingResp

8.226.2.4 BYTE* GetIMSSMSConfigParams::pSMSFormat

8.226.2.5 BYTE* GetIMSSMSConfigParams::pSMSOverIPNwInd

8.227 GetIMSUserConfigParams Struct Reference

Data Fields

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

8.227.1 Detailed Description

This structure contains the SLQSGetIMSUserConfig response parameters.

Parameters

pSettingResp	• Settings Response
pIMSDomainLen[IN/OUT]	• Length of IMS Domain Name to follow

pIMSDomain	• IMS domain name • Length of this string must be specified in pIMSDomainLen parameter

8.227.2 Field Documentation

8.227.2.1 BYTE* GetIMSUserConfigParams::pIMSDomain

8.227.2.2 BYTE* GetIMSUserConfigParams::pIMSDomainLen

8.227.2.3 BYTE* GetIMSUserConfigParams::pSettingResp

8.228 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

8.228.1 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 • Values: – True - Enable – False - Disable

pScrAmrEnable	• Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB) • Values: – True - Enable – False - Disable
pScrAmrWb-Enable	• Flag to enable/disable SCR for AMR WB Audio • Values: – True - Enable – False - Disable
pAmrMode	• BitMask for AMR NB modes allowed • Values: – 0x1 - 4.75 kbps – 0x2 - 5.15 kbps – 0x4 - 5.9 kbps – 0x8 - 6.17 kbps – 0x10 - 7.4 kbps – 0x20 - 7.95 kbps – 0x40 - 10.2 kbps – 0x80 - 12.2 kbps
pAmrWBMode	• BitMask for AMR WB modes allowed • Values: – 0x1 - 6.60 kbps – 0x2 - 8.85 kbps – 0x4 - 12.65 kbps – 0x8 - 14.25 kbps – 0x10 - 15.85 kbps – 0x20 - 18.25 kbps – 0x40 - 19.85 kbps – 0x80 - 23.05 kbps – 0x100 - 23.85 kbps
pAmrOctet-Aligned	• Flag to indicate if the octet is aligned for AMR NB Audio • Values: – True - Aligned – False - Not aligned, Bandwidth Efficient mode
pAmrWBOctet-Aligned	• Flag to indicate if the octet is aligned for AMR WB Audio • Values: – True - Aligned – 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.

8.228.2 Field Documentation

8.228.2.1 BYTE* GetIMSVoIPConfigResp::pAmrMode

8.228.2.2 BYTE* GetIMSVoIPConfigResp::pAmrOctetAligned

8.228.2.3 BYTE* GetIMSVoIPConfigResp::pAmrWbEnable

8.228.2.4 WORD* GetIMSVoIPConfigResp::pAmrWBMode

8.228.2.5 BYTE* GetIMSVoIPConfigResp::pAmrWBOctetAligned

8.228.2.6 WORD* GetIMSVoIPConfigResp::pMinSessionExpiryTimer

8.228.2.7 WORD* GetIMSVoIPConfigResp::pRingBackTimer

8.228.2.8 WORD* GetIMSVoIPConfigResp::pRingingTimer

8.228.2.9 WORD* GetIMSVoIPConfigResp::pRTPRTCPInactTimer

8.228.2.10 BYTE* GetIMSVoIPConfigResp::pScrAmrEnable

8.228.2.11 BYTE* GetIMSVoIPConfigResp::pScrAmrWbEnable

8.228.2.12 WORD* GetIMSVoIPConfigResp::pSessionExpiryTimer

8.228.2.13 BYTE* GetIMSVoIPConfigResp::pSettingResp

8.229 GetInstIDResp Struct Reference

Data Fields

- BYTE * pInstanceID
- BYTE * pIPFamily

8.229.1 Field Documentation

8.229.1.1 BYTE* GetInstIDResp::pInstanceID

8.229.1.2 BYTE* GetInstIDResp::pIPFamily

8.230 GetM2MAudioProfileReq Struct Reference

Data Fields

- BYTE * pGenerator

8.230.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile request parameters.

Parameters

pGenerator[optional]	• Generator – 0 - Voice

8.230.2 Field Documentation

8.230.2.1 BYTE* GetM2MAudioProfileReq::pGenerator

8.231 GetM2MAudioProfileResp Struct Reference

Data Fields

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

8.231.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile response parameters.

Parameters

Profile	• Audio Profile – 0-5
EarMute	• Ear Mute – 0 - Mute – 1 - Unmute

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

8.231.2 Field Documentation

8.231.2.1 BYTE GetM2MAudioProfileResp::CwtMute

8.231.2.2 BYTE GetM2MAudioProfileResp::EarMute

8.231.2.3 BYTE GetM2MAudioProfileResp::Generator

8.231.2.4 BYTE GetM2MAudioProfileResp::MicMute

8.231.2.5 BYTE GetM2MAudioProfileResp::Profile

8.231.2.6 BYTE GetM2MAudioProfileResp::Volume

8.232 GetM2MAudioVolumeReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Generator

8.232.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume request parameters.

Parameters

Profile	• Audio Profile – 0-5

Generator	• Generator – 0 - Voice

8.232.2 Field Documentation

8.232.2.1 BYTE GetM2MAudioVolumeReq::Generator

8.232.2.2 BYTE GetM2MAudioVolumeReq::Profile

8.233 GetM2MAudioVolumeResp Struct Reference

Data Fields

- BYTE Level

8.233.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume response parameters.

Parameters

Level	• The RX Volume Level – 0-5

8.233.2 Field Documentation

8.233.2.1 BYTE GetM2MAudioVolumeResp::Level

8.234 GetM2MAVMuteReq Struct Reference

Data Fields

- BYTE Profile

8.234.1 Detailed Description

This structure contains the SLQSGetM2MAVMute request parameters.

Parameters

Profile	• Audio Profile Number – 0-5

8.234.2 Field Documentation

8.234.2.1 BYTE GetM2MAVMuteReq::Profile

8.235 GetM2MAVMuteResp Struct Reference

Data Fields

- BYTE EarMute
- BYTE MicMute
- BYTE CwtMute

8.235.1 Detailed Description

This structure contains the SLQSGetM2MAVMute response parameters.

Parameters

pEarMute	• Ear Mute – 0-Mute – 1-UnMute
pMicMute	• Mic Mute – 0-Mute – 1-unmute
CwtMute	• Waiting tone Mute – 0-5

8.235.2 Field Documentation

8.235.2.1 BYTE GetM2MAVMuteResp::CwtMute

8.235.2.2 BYTE GetM2MAVMuteResp::EarMute

8.235.2.3 BYTE GetM2MAVMuteResp::MicMute

8.236 GetM2MSpkrGainReq Struct Reference

Data Fields

- BYTE Profile

8.236.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain request parameters.

Parameters

pProfile	• Audio Profile Number – 0-5

8.236.2 Field Documentation

8.236.2.1 BYTE GetM2MSpkrGainReq::Profile

8.237 GetM2MSpkrGainResp Struct Reference

Data Fields

- WORD Value

8.237.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain response parameters.

Parameters

Value	• RX speakerphone gain – 0x0 - 0x7fff

8.237.2 Field Documentation

8.237.2.1 WORD GetM2MSpkrGainResp::Value

8.238 getMsgWaitingInfo Struct Reference

Data Fields

- BYTE numInstances
- messageWaitingInfoContent msgWaitInfo [0xFF]

8.238.1 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. – See messageWaitingInfoContent for more information.

8.238.2 Field Documentation

8.238.2.1 messageWaitingInfoContent getMsgWaitingInfo::msgWaitInfo[0xFF]

8.238.2.2 BYTE getMsgWaitingInfo::numInstances

8.239 GetNetworkTimeResp Struct Reference

Data Fields

- timeInfo * p3GPP2TimeInfo
- timeInfo * p3GPPTimeInfo

8.239.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

Parameters

p3GPP2TimeInfo	[Optional] • See timeInfo for more information
p3GPPTimeInfo	[Optional] • See timeInfo for more information

8.239.2 Field Documentation

8.239.2.1 timeInfo* GetNetworkTimeResp::p3GPP2TimeInfo

8.239.2.2 timeInfo* GetNetworkTimeResp::p3GPPTimeInfo

8.240 GetRegMgrConfigParams Struct Reference

Data Fields

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

8.240.1 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 • Length of this string must be specified in pPriCSCFPortNameLen parameter
pIMSTestMode	• IMS Test mode Enabled. – TRUE - Enabled – FALSE - Disabled

Note

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

8.240.2 Field Documentation

8.240.2.1 BYTE* GetRegMgrConfigParams::pIMSTestMode

8.240.2.2 WORD* GetRegMgrConfigParams::pPCSCFPort

8.240.2.3 BYTE* GetRegMgrConfigParams::pPriCSCFPortName

8.240.2.4 BYTE* GetRegMgrConfigParams::pPriCSCFPortNameLen

8.240.2.5 BYTE* GetRegMgrConfigParams::pSettingResp

8.241 GetSessionIDResp Struct Reference

Data Fields

- ULONG * pSessionIDv4
- ULONG * pSessionIDv6

8.241.1 Field Documentation

8.241.1.1 ULONG* GetSessionIDResp::pSessionIDv4

8.241.1.2 ULONG* GetSessionIDResp::pSessionIDv6

8.242 GetSIPConfigResp Struct Reference

Data Fields

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

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

8.242.1 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 – TRUE - Sig Comp Enabled – FALSE - Sig Comp Disabled

8.242.2 Field Documentation

8.242.2.1 BYTE* GetSIPConfigResp::pSettingResp

8.242.2.2 BYTE* GetSIPConfigResp::pSigCompEnabled

8.242.2.3 WORD* GetSIPConfigResp::pSIPLocalPort

8.242.2.4 ULONG* GetSIPConfigResp::pSubscribeTimer

8.242.2.5 ULONG* GetSIPConfigResp::pTimerSIPReg

8.242.2.6 ULONG* GetSIPConfigResp::pTimerT1

8.242.2.7 ULONG* GetSIPConfigResp::pTimerT2

8.242.2.8 ULONG* GetSIPConfigResp::pTimerTf

8.243 GnssData Struct Reference

Data Fields

- ULONGLONG mask

8.243.1 Detailed Description

This structure contains the GNSS data

Parameters

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

8.243.2 Field Documentation

8.243.2.1 ULONGLONG GnssData::mask

8.244 gnssSvInfoNotification Struct Reference

Data Fields

- BYTE bAltitudeAssumed
- satelliteInfo * pSatelliteInfo

8.244.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

Parameters

bAltitude-Assumed	• Indicates whether altitude is assumed or calculated – 0x00 (FALSE) - Valid altitude is calculated – 0x01 (TRUE) - Valid altitude is assumed; there may not be enough satellites to determine precise altitude
pSatelliteInfo	• See satelliteInfo for more information.

Note

None

8.244.2 Field Documentation

8.244.2.1 BYTE gnssSvInfoNotification::bAltitudeAssumed

8.244.2.2 satelliteInfo* gnssSvInfoNotification::pSatelliteInfo

8.245 GPRSQoS Struct Reference

Data Fields

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

8.245.1 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

8.245.2 Field Documentation

8.245.2.1 ULONG GPRSQoS::delayClass

8.245.2.2 ULONG GPRSQoS::meanThroughputClass

8.245.2.3 ULONG GPRSQoS::peakThroughputClass

8.245.2.4 ULONG GPRSQoS::precedenceClass

8.245.2.5 ULONG GPRSQoS::reliabilityClass

8.246 GPRSRequestedQoS Struct Reference

Data Fields

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

8.246.1 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

8.246.2 Field Documentation

8.246.2.1 ULONG GPRSRequestedQoS::delayClass

8.246.2.2 ULONG GPRSRequestedQoS::meanThroughputClass

8.246.2.3 ULONG GPRSRequestedQoS::peakThroughputClass

8.246.2.4 ULONG GPRSRequestedQoS::precedenceClass

8.246.2.5 ULONG GPRSRequestedQoS::reliabilityClass

8.247 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

8.247.1 Detailed Description

GPS state Info.

Parameters

EngineState	• Values: – 0 - OFF – 1 - ON • This field is always valid
ValidMask	• Mask of valid state information data. • Values: – 0x00000001 - Position(latitude/longitude/horizontal uncertainty) – 0x00000002 - Altitude and vertical uncertainty – 0x00000004 - Time ms – 0x00000008 - Time week number – 0x00000010 - Time uncertainty – 0x00000020 - Iono validity – 0x00000040 - GPS ephemeris – 0x00000080 - GPS almanac – 0x00000100 - GPS health – 0x00000200 - GPS visible SVs – 0x00000400 - GLONASS ephemeris – 0x00000800 - GLONASS almanac – 0x00001000 - GLONASS health – 0x00002000 - GLONASS visible SVs – 0x00004000 - SBAS ephemeris – 0x00008000 - SBAS almanac – 0x00010000 - SBAS health – 0x00020000 - SBAS visible SVs – 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. • Units: Decimal degrees • Range: -90 to +90 degrees. • 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. • Units: Decimal degrees • Range: -180 to +180 degrees • 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. • 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 • Range -500 to 15883 • 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. • 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. • Values: – 0 - Invalid – 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

8.247.2 Field Documentation

8.247.2.1 ULONG GPSStateInfo::Altitude

8.247.2.2 BYTE GPSStateInfo::EngineState

8.247.2.3 ULONG GPSStateInfo::glo_almanac_sv_msk

8.247.2.4 ULONG GPSStateInfo::glo_ephemeris_sv_msk

8.247.2.5 ULONG GPSStateInfo::glo_health_sv_msk

8.247.2.6 ULONG GPSStateInfo::glo_visible_sv_msk

8.247.2.7 ULONG GPSStateInfo::gps_almanac_sv_msk

8.247.2.8 ULONG GPSStateInfo::gps_ephemeris_sv_msk

8.247.2.9 ULONG GPSStateInfo::gps_health_sv_msk

8.247.2.10 ULONG GPSStateInfo::gps_visible_sv_msk

8.247.2.11 ULONG GPSStateInfo::HorizontalUncertainty

8.247.2.12 BYTE GPSStateInfo::Iono_valid

8.247.2.13 ULONGLONG GPSStateInfo::Latitude

8.247.2.14 ULONGLONG GPSStateInfo::Longitude

8.247.2.15 ULONG GPSStateInfo::sbas_almanac_sv_msk

8.247.2.16 ULONG GPSStateInfo::sbas_ephemeris_sv_msk

8.247.2.17 ULONG GPSStateInfo::sbas_health_sv_msk

8.247.2.18 ULONG GPSStateInfo::sbas_visible_sv_msk

8.247.2.19 ULONG GPSStateInfo::Time_uncert_ms

8.247.2.20 WORD GPSStateInfo::TimeStmp_gps_week

8.247.2.21 ULONG GPSStateInfo::TimeStmp_tow_ms

8.247.2.22 ULONG GPSStateInfo::ValidMask

8.247.2.23 ULONG GPSStateInfo::VerticalUncertainty

8.247.2.24 WORD GPSStateInfo::xtra_start_gps_minutes

8.247.2.25 WORD GPSStateInfo::xtra_start_gps_week

8.247.2.26 WORD GPSStateInfo::xtra_valid_duration_hours

8.248 gpsTime_s Struct Reference

Data Fields

- WORD gpsWeek
- ULONG gpsTimeOfWeekMs

8.248.1 Detailed Description

This structure contains GPS Time info.

Parameters

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

8.248.2 Field Documentation

8.248.2.1 ULONG gpsTime_s::gpsTimeOfWeekMs

8.248.2.2 WORD gpsTime_s::gpsWeek

8.249 gsmCellInfo Struct Reference

Data Fields

- WORD arfcn
- BYTE band1900

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

8.249.1 Detailed Description

This structure contains information about the GSM Cell.

Parameters

arfcn	• GSM frequency being reported. • Range: 0 to 1023.
band1900	• Band indicator for the GSM ARFCN • This field is only valid if arfcn is in the overlapping region. • If TRUE and the cell is in the overlapping region, the ARFCN is on the 1900 band. • If FALSE, it is on the 1800 band.
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. • The lower 6 bits can be set to any value.
rssi	• Measured RSSI value in 1/10 dB. • Range: -200.0 dB to 0
srxlev	• Cell selection Rx level (Srxlev) value. • Range: -128 to 128. • This field is only valid when ue_in_idle is TRUE.

8.249.2 Field Documentation

8.249.2.1 WORD gsmCellInfo::arfcn

8.249.2.2 BYTE gsmCellInfo::band1900

8.249.2.3 BYTE gsmCellInfo::bsicId

8.249.2.4 BYTE gsmCellInfo::cellIdValid

8.249.2.5 SHORT gsmCellInfo::rssi

8.249.2.6 SHORT gsmCellInfo::srxlev

8.250 GSMRSSIThresh Struct Reference

Data Fields

- BYTE GSMRSSIThreshListLen
- WORD * pGSMRSSIThreshList

8.250.1 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) • Maximum of 32 values • Range for RSSI values: -111 to -48 (in dBm)

8.250.2 Field Documentation

8.250.2.1 BYTE GSMRSSIThresh::GSMRSSIThreshListLen

8.250.2.2 WORD* GSMRSSIThresh::pGSMRSSIThreshList

8.251 GSMSrvStatusInfo Struct Reference

Data Fields

- BYTE srvStatus
- BYTE trueSrvStatus
- BYTE isPrefDataPath

8.251.1 Detailed Description

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

Parameters

srvStatus	• Service status of the system. – 0x00 - No service – 0x01 - Limited service – 0x02 - Service – 0x03 - Limited regional service – 0x04 - Power save – 0xFF - Not Available

trueSrvStatus	• True service status of the system. • Not applicable to CDMA/HDR. – 0x00 - No service – 0x01 - Limited service – 0x02 - Service – 0x03 - Limited regional service – 0x04 - Power save – 0xFF - Not Available
isPrefDataPath	• Whether the RAT is the preferred data path. – 0x00 - Not preferred – 0x01 - Preferred – 0xFF - Not Available

8.251.2 Field Documentation

8.251.2.1 BYTE GSMSrvStatusInfo::isPrefDataPath

8.251.2.2 BYTE GSMSrvStatusInfo::srvStatus

8.251.2.3 BYTE GSMSrvStatusInfo::trueSrvStatus

8.252 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

8.252.1 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.. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
lac	• Location area code. • Only applies to 3GPP. – 0xFFFF - Not Available
cellIdValid	• Indicates whether the cell ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
cellId	• Cell ID. – 0xFFFFFFFF - Not Available
regRejectInfo-Valid	• Indicates whether the registration reject information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
rejectSrvDomain	• Type of service domain in which the registration is rejected. – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
rejCause	• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. – 0xFF - Not Available
networkIdValid	• Indicates whether the network ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

MCC[PLMN_LENGTH]	• Mobile Country Code. • MCC digits in ASCII characters
MNC[PLMN_LENGTH]	• Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.
egprsSuppValid	• Indicates whether the EGPRS support is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
egprsSupp	• EGPRS support indication. • Only applicable for GSM. – 0x00 - Not available – 0x01 - Available – 0xFF - Not Available
dtmSuppValid	• Indicates whether Dual Transfer mode support is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
dtmSupp	• Dual Transfer mode support indication. • Only applicable for GSM. – 0x00 - Not available – 0x01 - Available – 0xFF - Not Available

8.252.2 Field Documentation

8.252.2.1 ULONG GSMSysInfo::cellId

8.252.2.2 BYTE GSMSysInfo::cellIdValid

8.252.2.3 BYTE GSMSysInfo::dtmSupp

8.252.2.4 BYTE GSMSysInfo::dtmSuppValid

8.252.2.5 BYTE GSMSysInfo::egprsSupp

8.252.2.6 BYTE GSMSysInfo::egprsSuppValid

8.252.2.7 WORD GSMSysInfo::lac

8.252.2.8 BYTE GSMSysInfo::lacValid

8.252.2.9 BYTE GSMSysInfo::MCC[3]

8.252.2.10 BYTE GSMSysInfo::MNC[3]

8.252.2.11 BYTE GSMSysInfo::networkIdValid

8.252.2.12 BYTE GSMSysInfo::regRejectInfoValid

8.252.2.13 BYTE GSMSysInfo::rejCause

8.252.2.14 BYTE GSMSysInfo::rejectSrvDomain

8.252.2.15 sysInfoCommon GSMSysInfo::sysInfoGSM

8.253 gyroAcceptReady_s Struct Reference

Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

8.253.1 Detailed Description

This structure contains Gyroscope Accept Ready Info

Parameters

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

8.253.2 Field Documentation

8.253.2.1 WORD gyroAcceptReady_s::batchPerSec

8.253.2.2 BYTE gyroAcceptReady_s::injectEnable

8.253.2.3 WORD gyroAcceptReady_s::samplesPerBatch

8.254 gyroTempAcceptReady_s Struct Reference

Data Fields

- BYTE injectEnable
- WORD samplesPerBatch
- WORD batchPerSec

8.254.1 Detailed Description

This structure contains Gyroscope Temperature Accept Ready Info

Parameters

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

8.254.2 Field Documentation

8.254.2.1 WORD gyroTempAcceptReady_s::batchPerSec

8.254.2.2 BYTE gyroTempAcceptReady_s::injectEnable

8.254.2.3 WORD gyroTempAcceptReady_s::samplesPerBatch

8.255 HDRECIOThresh Struct Reference

Data Fields

- BYTE HDRECIOThreshListLen
- WORD ∗ pHDRECIOThreshList

8.255.1 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) • Maximum of 32 values • Range for ECIO values: -31.5 to 0 (in dB).

8.255.2 Field Documentation

8.255.2.1 BYTE HDRECIOThresh::HDRECIOThreshListLen

8.255.2.2 WORD* HDRECIOThresh::pHDRECIOThreshList

8.256 HDRIOThresh Struct Reference

Data Fields

- BYTE HDRIOThreshListLen
- WORD * pHDRIOThreshList

8.256.1 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) • Maximum of 32 values • Range for IO values: -128 to -13 (in dBm).

8.256.2 Field Documentation

8.256.2.1 BYTE HDRIOThresh::HDRIOThreshListLen

8.256.2.2 WORD* HDRIOThresh::pHDRIOThreshList

8.257 HDRPersonalityInd Struct Reference

Data Fields

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

8.257.1 Field Documentation

8.257.1.1 WORD* HDRPersonalityInd::pCurrentPersonality

8.257.1.2 BYTE* HDRPersonalityInd::pPersonalityListLength

8.257.1.3 protocolSubtypeElement* HDRPersonalityInd::pProtocolSubtypeElement

8.258 HDRPersonalityResp Struct Reference

Data Fields

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

8.258.1 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. • maximum input value is 3
pProtocol-Subtype-Element[Out]	• See protocolSubtypeElement for more information.

8.258.2 Field Documentation

8.258.2.1 WORD* HDRPersonalityResp::pCurrentPersonality

8.258.2.2 BYTE* HDRPersonalityResp::pPersonalityListLength

8.258.2.3 protocolSubtypeElement* HDRPersonalityResp::pProtocolSubtypeElement

8.259 HDRProtSubtypResp Struct Reference

Data Fields

- WORD * pCurrentPrsnlty
- BYTE * pPersonalityListLength

- protocolSubtypeElement * pProtoSubTypElmnt
- ULONGLONG * pAppSubType

8.259.1 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. • maximum input value is 4
pProtocol-Subtype-Element[Out]	• See protocolSubtypeElement for more information.
pAppSubType[-Out]	• Stream application subtype • Application subtype for each stream,

8.259.2 Field Documentation

8.259.2.1 ULONGLONG* HDRProtSubtypResp::pAppSubType

8.259.2.2 WORD* HDRProtSubtypResp::pCurrentPrsnlty

8.259.2.3 BYTE* HDRProtSubtypResp::pPersonalityListLength

8.259.2.4 protocolSubtypeElement* HDRProtSubtypResp::pProtoSubTypElmnt

8.260 HDRRSSIThresh Struct Reference

Data Fields

- BYTE HDRRSSIThreshListLen
- WORD * pHDRRSSIThreshList

8.260.1 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) • Maximum of 32 values. • Range for RSSI values: -118 to -13 (in dBm).

8.260.2 Field Documentation

8.260.2.1 BYTE HDRRSSIThresh::HDRRSSIThreshListLen

8.260.2.2 WORD* HDRRSSIThresh::pHDRRSSIThreshList

8.261 HDRSINRThresh Struct Reference

Data Fields

- BYTE HDRSINRThresListLen
- BYTE * pHDRSINRThresList

8.261.1 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 • 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 – SINR is reported only for HDR – Each SINR threshold value is an unsigned 1 byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.261.2 Field Documentation

8.261.2.1 BYTE HDRSINRThresh::HDRSINRThresListLen

8.261.2.2 BYTE* HDRSINRThresh::pHDRSINRThresList

8.262 HDRSINRThreshold Struct Reference

Data Fields

- BYTE HDRSINRThreshListLen
- WORD * pHDRSINRThreshList

8.262.1 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) • maximum of 32 values. • Valid levels are 0 to 8 – 0x00 - SINR_LEVEL_0 is -9 dB – 0x01 - SINR_LEVEL_1 is -6 dB – 0x02 - SINR_LEVEL_2 is -4.5 dB – 0x03 - SINR_LEVEL_3 is -3 dB – 0x04 - SINR_LEVEL_4 is -2 dB – 0x05 - SINR_LEVEL_5 is +1 dB – 0x06 - SINR_LEVEL_6 is +3 dB – 0x07 - SINR_LEVEL_7 is +6 dB – 0x08 - SINR_LEVEL_8 is +9 dB

8.262.2 Field Documentation

8.262.2.1 BYTE HDRSINRThreshold::HDRSINRThreshListLen

8.262.2.2 WORD* HDRSINRThreshold::pHDRSINRThreshList

8.263 HDRSSInfo Struct Reference

Data Fields

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

8.263.1 Detailed Description

This structure contains the parameters for HDR Signal Strength Information

Parameters

rssi	• RSSI in dBm (signed value). • 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. • SINR is only applicable for 1xEV-DO. • Valid levels are 0 to 8, where the 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 – 0xFF - Not Available
io	• Received IO in dBm. • IO is only applicable for 1xEV-DO.

8.263.2 Field Documentation

8.263.2.1 SHORT HDRSSInfo::ecio

8.263.2.2 INT32 HDRSSInfo::io

8.263.2.3 INT8 HDRSSInfo::rssi

8.263.2.4 BYTE HDRSSInfo::sinr

8.264 hdrSSInfo Struct Reference

Data Fields

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

8.264.1 Detailed Description

Parameters

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

8.264.2 Field Documentation

8.264.2.1 int16_t hdrSSInfo::ecio

8.264.2.2 int32_t hdrSSInfo::io

8.264.2.3 int8_t hdrSSInfo::rssi

8.264.2.4 uint8_t hdrSSInfo::sinr

8.265 HDRSysInfo Struct Reference

Data Fields

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

8.265.1 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. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

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

8.265.2 Field Documentation

8.265.2.1 BYTE HDRSysInfo::hdrActiveProt

8.265.2.2 BYTE HDRSysInfo::hdrActiveProtValid

8.265.2.3 BYTE HDRSysInfo::hdrPersonality

8.265.2.4 BYTE HDRSysInfo::hdrPersonalityValid

8.265.2.5 BYTE HDRSysInfo::is856SysId[16]

8.265.2.6 BYTE HDRSysInfo::is856SysIdValid

8.265.2.7 BYTE HDRSysInfo::isSysPrlMatch

8.265.2.8 BYTE HDRSysInfo::isSysPrlMatchValid

8.265.2.9 sysInfoCommon HDRSysInfo::sysInfoHDR

8.266 homeSIDNID Struct Reference

Data Fields

- BYTE numInstances
- sidNid SidNid [255]

8.266.1 Detailed Description

This structure contains the parameters for Home SID/NID Information

Parameters

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

8.266.2 Field Documentation

8.266.2.1 BYTE homeSIDNID::numInstances

8.266.2.2 sidNid homeSIDNID::SidNid[255]

8.267 hotSwapStatus Struct Reference

Data Fields

- BYTE hotSwapLength
- BYTE hotSwap [255]

8.267.1 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. – 0 - Hot-swap is not supported – 1 - Hot-swap is supported, but the status of the switch is not supported – 2 - Switch indicates that the card is present – 3 - Switch indicates that the card is not present

8.267.2 Field Documentation

8.267.2.1 BYTE hotSwapStatus::hotSwap[255]

8.267.2.2 BYTE hotSwapStatus::hotSwapLength

8.268 image_info_t Struct Reference

Data Fields

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

8.268.1 Field Documentation

8.268.1.1 uint8_t image_info_t::buildID[255]

8.268.1.2 uint8_t image_info_t::buildIDLen

8.268.1.3 uint8_t image_info_t::imageType

8.268.1.4 uint8_t image_info_t::uniqueID[16]

8.269 ImageElement Struct Reference

Data Fields

- BYTE imageType

- BYTE imageId [16]
- BYTE buildIdLength
- CHAR buildId [100]

8.269.1 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)

8.269.2 Field Documentation

8.269.2.1 CHAR ImageElement::buildId[100]

8.269.2.2 BYTE ImageElement::buildIdLength

8.269.2.3 BYTE ImageElement::imageId[16]

8.269.2.4 BYTE ImageElement::imageType

8.270 ImageIdElement Struct Reference

Data Fields

- BYTE storageIndex
- BYTE failureCount
- BYTE imageID [16]
- BYTE buildIDLength
- CHAR buildID [100]

8.270.1 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)

8.270.2 Field Documentation

8.270.2.1 CHAR ImageIdElement::buildID[100]

8.270.2.2 BYTE ImageIdElement::buildIDLength

8.270.2.3 BYTE ImageIdElement::failureCount

8.270.2.4 BYTE ImageIdElement::imageID[16]

8.270.2.5 BYTE ImageIdElement::storageIndex

8.271 ImageIDEntries Struct Reference

Data Fields

- BYTE imageType
- BYTE maxImages
- BYTE executingImage
- BYTE imageIDSize
- struct ImageIdElement imageIDElement [50]

8.271.1 Detailed Description

This structure contains the list entry Information

Parameters

imageType	• Type of image – 0 - Modem – 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)

8.271.2 Field Documentation

8.271.2.1 BYTE ImageIDEntries::executingImage

8.271.2.2 struct ImageIdElement ImageIDEntries::imageIDElement[50]

8.271.2.3 BYTE ImageIDEntries::imageIDSize

8.271.2.4 BYTE ImageIDEntries::imageType

8.271.2.5 BYTE ImageIDEntries::maxImages

8.272 ImageList Struct Reference

Data Fields

- BYTE listSize
- struct ImageIDEntries imageIDEntries [2]

8.272.1 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)

8.272.2 Field Documentation

8.272.2.1 struct ImageIDEntries ImageList::imageIDEntries[2]

8.272.2.2 BYTE ImageList::listSize

8.273 IMSAIndRegisterInfo Struct Reference

Data Fields

- BYTE * pRegStatusConfig
- BYTE * pServiceStatusConfig
- BYTE * pRatHandoverStatusConfig

- BYTE * pPdpStatusConfig

8.273.1 Detailed Description

This structure contains parameters of IMSA Config Indication Register

Parameters

pRegStatus-Config(optional)	• Register Indication For Registration status. • When this registration is enabled, the device learns of Registration status via the QMI_IMSA_-REGISTRATION_STATUS_IND indication. – 0x00 - Disable – 0x01 - Enable
pServiceStatus-Config(optional)	• Register Indication For Service status Events. • When this registration is enabled, the device learns of Service status via the QMI_IMSA_SER-VICE_STATUS_IND indication. – 0x00 - Disable – 0x01 - Enable
pRatHandover-Status-Config(optional)	• Registration Indication For RAT handover status. • When this registration is enabled, the device learns of RAT handover status via the QMI_IMSA-_RAT_HANDOVER_STATUS_IND indication. – 0x00 - Disable – 0x01 - Enable
pPdpStatus-Config(optional)	• PDP Status Configuration. – 0x00 - Disable – 0x01 - Enable

Note

One of the optional parameter is mandatory to be present in the request.

8.273.2 Field Documentation

8.273.2.1 BYTE* IMSAIndRegisterInfo::pPdpStatusConfig

8.273.2.2 BYTE* IMSAIndRegisterInfo::pRatHandoverStatusConfig

8.273.2.3 BYTE* IMSAIndRegisterInfo::pRegStatusConfig

8.273.2.4 BYTE* IMSAIndRegisterInfo::pServiceStatusConfig

8.274 imsaPdpStatusInfo Struct Reference

Data Fields

- BYTE connetionState

- ULONG * pFailErrorCode

8.274.1 Detailed Description

Contains the parameters passed for SLQSSetIMSAPdpStatusCallback by the device.

Parameters

connetionState	• IMS PDP connection state information. • Values – TRUE – IMS PDP is connected – FALSE – IMS PDP is not connected
pFailErrorCode	• IMS PDP connection failure error reason code when the IMS PDP Connection State TLV is FALSE. • Values – 0 - Generic failure reason for other than specified – 1 - Option is unsubscribed. – 2 - PDP status was unknown.

8.274.2 Field Documentation

8.274.2.1 BYTE imsaPdpStatusInfo::connetionState

8.274.2.2 ULONG* imsaPdpStatusInfo::pFailErrorCode

8.275 imsaRatStatusInfo Struct Reference

Data Fields

- ULONG * pRATStatus
- ULONG * pSrcRAT
- ULONG * pTgtRAT
- BYTE * pErrorCodeStr

8.275.1 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

8.275.2 Field Documentation

8.275.2.1 BYTE* imsaRatStatusInfo::pErrorCodeStr

8.275.2.2 ULONG* imsaRatStatusInfo::pRATStatus

8.275.2.3 ULONG* imsaRatStatusInfo::pSrcRAT

8.275.2.4 ULONG* imsaRatStatusInfo::pTgtRAT

8.276 IMSARegistrationStatus Struct Reference

Data Fields

- BYTE * pImsRegStatus
- WORD * pImsRegErrCode
- ULONG * pNewImsRegStatus

8.276.1 Detailed Description

This structure contains response parameters of registration status.

Parameters

pImsRegStatus	• IMS Registration Status (Deprecated). • Values – TRUE - UE is registered on the IMS network – FALSE - UE is not registered on the IMS network
pImsRegErrCode	• IMS Registration Error Code. • An error code is returned when the IMS registration status is IMSA_STATUS_NOT_REGISTERED. -Values – 3xx – Redirection responses – 4xx – Client failure responses – 5xx – Server failure responses – 6xx – Global failure responses
pNewImsRegStatus	• New IMS Registration Status • Values – 0 - Not registered for IMS – 1 - Registering for IMS – 2 - Registered for IMS

8.276.2 Field Documentation

8.276.2.1 WORD* IMSARegistrationStatus::pImsRegErrCode

8.276.2.2 BYTE* IMSARegistrationStatus::pImsRegStatus

8.276.2.3 ULONG* IMSARegistrationStatus::pNewImsRegStatus

8.277 imsaRegStatusInfo Struct Reference

Data Fields

- BYTE * pbIMSRegistered
- WORD * pRegStatusErrorCode
- ULONG * pImsRegStatus

8.277.1 Detailed Description

Contains the parameters passed for SLQSSetIMSARegStatusCallback by the device.

Parameters

pbIMS-Registered	• TRUE/FALSE
pRegStatus-ErrorCode	• 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

8.277.2 Field Documentation

8.277.2.1 BYTE* imsaRegStatusInfo::pbIMSRegistered

8.277.2.2 ULONG* imsaRegStatusInfo::pImsRegStatus

8.277.2.3 WORD* imsaRegStatusInfo::pRegStatusErrorCode

8.278 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

8.278.1 Detailed Description

This structure contains response parameters of service status for various IMS services.

Parameters

pSmsService-Status	• SMS Service Status. • Values – 0 - IMS SMS service is not available – 1 - IMS SMS is in limited service – 2 - IMS SMS is in full service
pVoipSerive-Status	• VoIP Service Status. -Values – 0 - IMS VoIP service is not available – 2 - IMS VoIP is in full service
pVtService-Status	• VT Service Status • Values – 0 - IMS VT service is not available – 2 - IMS VT is in full service
pSmsServiceRat	• SMS service RAT • Values – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN
pVoipServiceRat	• VoIP service RAT. • Values – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN
pVtServiceRat	• VT service RAT. • Values – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN
pUtService-Status	• UT service Status. • Values – 0 - IMS UT service is not available – 2 - IMS UT is in full service

pUtServiceRat	• UT service RAT. • Values – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN
pVsService-Status	• VS service Status. • Values – 0 - IMS UT service is not available – 2 - IMS UT is in full service
pVsServiceRat	• VS service RAT. • Values – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN

8.278.2 Field Documentation

8.278.2.1 ULONG* IMSAServiceStatus::pSmsServiceRat

8.278.2.2 ULONG* IMSAServiceStatus::pSmsServiceStatus

8.278.2.3 ULONG* IMSAServiceStatus::pUtServiceRat

8.278.2.4 ULONG* IMSAServiceStatus::pUtServiceStatus

8.278.2.5 ULONG* IMSAServiceStatus::pVoipServiceRat

8.278.2.6 ULONG* IMSAServiceStatus::pVoipServiceStatus

8.278.2.7 ULONG* IMSAServiceStatus::pVsServiceRat

8.278.2.8 ULONG* IMSAServiceStatus::pVsServiceStatus

8.278.2.9 ULONG* IMSAServiceStatus::pVtServiceRat

8.278.2.10 ULONG* IMSAServiceStatus::pVtServiceStatus

8.279 IMSASupportedFieldsResp Struct Reference

Data Fields

- struct ReqFieldsList * pReqFieldsList
- struct RespFieldsList * pRespFieldsList
- struct IndFieldsList * pIndFieldsList

8.279.1 Detailed Description

This structure contains response of supported fields by the currently running software.

Parameters

pReqFieldsList	• List of Supported Request Fields. • See ReqFieldsList for more information
pRespFieldsList	• List of Supported Request Fields. • See RespFieldsList for more information
pIndFieldsList	• List of Supported Request Fields. • See IndFieldsList for more information

8.279.2 Field Documentation

8.279.2.1 struct IndFieldsList* IMSASupportedFieldsResp::pIndFieldsList

8.279.2.2 struct ReqFieldsList* IMSASupportedFieldsResp::pReqFieldsList

8.279.2.3 struct RespFieldsList* IMSASupportedFieldsResp::pRespFieldsList

8.280 IMSASupportedMsgInfo Struct Reference

Data Fields

- struct SupportedMsgList * pSupportedMsgList

8.280.1 Detailed Description

This structure contains Queries the set of messages implemented by the currently running software.

Parameters

pSupportedMsg-List	• List of Supported Messages. • See SupportedMsgList for more information

8.280.2 Field Documentation

8.280.2.1 struct SupportedMsgList* IMSASupportedMsgInfo::pSupportedMsgList

8.281 imsaSvcStatusInfo Struct Reference

Data Fields

- ULONG * pSMSSvcStatus

- ULONG * pVOIPSvcStatus
- ULONG * pVTSvcStatus
- ULONG * pSMSSvcRAT
- ULONG * pVOIPSvcRAT
- ULONG * pVTSvcRAT
- ULONG * pUTSvcStatus
- ULONG * pUTSvcRAT

8.281.1 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

8.281.2 Field Documentation

8.281.2.1 ULONG* imsaSvcStatusInfo::pSMSSvcRAT

8.281.2.2 ULONG* imsaSvcStatusInfo::pSMSSvcStatus

8.281.2.3 ULONG* imsaSvcStatusInfo::pUTSvcRAT

8.281.2.4 ULONG* imsaSvcStatusInfo::pUTSvcStatus

8.281.2.5 ULONG* imsaSvcStatusInfo::pVOIPSvcRAT

8.281.2.6 ULONG* imsaSvcStatusInfo::pVOIPSvcStatus

8.281.2.7 ULONG* imsaSvcStatusInfo::pVTSvcRAT

8.281.2.8 ULONG* imsaSvcStatusInfo::pVTSvcStatus

8.282 imsCfgIndRegisterInfo Struct Reference

Data Fields

- BYTE * pSIPConfigEvents
- BYTE * pRegMgrConfigEvents
- BYTE * pSMSConfigEvents
- BYTE * pUserConfigEvents
- BYTE * pVoIPConfigEvents

8.282.1 Detailed Description

This structure contains parameters of IMS Config Indication Register

Parameters

pSIPConfig-Events(optional)	• Registration Indication For SIP Configuration Events. • When this registration is enabled, the device learns of SIP config events via the QMI_IMS_SIP-_CONFIG_IND indication. – 0x00 - Disable – 0x01 - Enable
pRegMgrConfig-Events(optional)	• Registration Indication For Registration Manager Configuration Events. • When this registration is enabled, the device learns of Reg Mgr config events via the QMI_IMS-_REG_MGR_CONFIG_IND indication. – 0x00 - Disable – 0x01 - Enable
pSMSConfig-Events(optional)	• Registration Indication For SMS Configuration Events. • When this registration is enabled, the device learns of SMS config events via the QMI_IMS_S-MS_CONFIG_IND indication. – 0x00 - Disable – 0x01 - Enable
pUserConfig-Events(optional)	• Registration Indication For User Configuration Events. • When this registration is enabled, the device learns of user config events via the QMI_IMS_US-ER_CONFIG_IND indication. – 0x00 - Disable – 0x01 - Enable
pVoIPConfig-Events(optional)	• Registration Indication For VoIP Configuration Events. • When this registration is enabled, the device learns of VOIP config events via the QMI_IMS_V-OIP_CONFIG_IND indication. – 0x00 - Disable – 0x01 - Enable

Note

One of the optional parameter is mandatory to be present in the request.

8.282.2 Field Documentation

8.282.2.1 BYTE* imsCfgIndRegisterInfo::pRegMgrConfigEvents

8.282.2.2 BYTE* imsCfgIndRegisterInfo::pSIPConfigEvents

8.282.2.3 BYTE* imsCfgIndRegisterInfo::pSMSConfigEvents

8.282.2.4 BYTE* imsCfgIndRegisterInfo::pUserConfigEvents

8.282.2.5 BYTE* imsCfgIndRegisterInfo::pVoIPConfigEvents

8.283 imsRegMgrConfigInfo Struct Reference

Data Fields

- WORD * pPriCSCFPort
- BYTE * pCSCFPortName
- BYTE * pIMSTestMode

8.283.1 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 • Length of this string can be of maximum 255 bytes
pIMSTestMode	• IMS Test mode Enabled. – TRUE - Enable, no IMS registration – FALSE - Disable, IMS registration is initiated

8.283.2 Field Documentation

8.283.2.1 BYTE* imsRegMgrConfigInfo::pCSCFPortName

8.283.2.2 BYTE* imsRegMgrConfigInfo::pIMSTestMode

8.283.2.3 WORD* imsRegMgrConfigInfo::pPriCSCFPort

8.284 imsSIPConfigInfo Struct Reference

Data Fields

- WORD * pSIPLocalPort
- ULONG * pTimerSIPReg
- ULONG * pSubscribeTimer
- ULONG * pTimerT1
- ULONG * pTimerT2
- ULONG * pTimerTf
- BYTE * pSigCompEnabled

8.284.1 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 – TRUE - Enable – FALSE - Disable

Note

None

8.284.2 Field Documentation

8.284.2.1 BYTE* imsSIPConfigInfo::pSigCompEnabled

8.284.2.2 WORD* imsSIPConfigInfo::pSIPLocalPort

8.284.2.3 ULONG* imsSIPConfigInfo::pSubscribeTimer

8.284.2.4 ULONG* imsSIPConfigInfo::pTimerSIPReg

8.284.2.5 ULONG* imsSIPConfigInfo::pTimerT1

8.284.2.6 ULONG* imsSIPConfigInfo::pTimerT2

8.284.2.7 ULONG* imsSIPConfigInfo::pTimerTf

8.285 imsSMSConfigInfo Struct Reference

Data Fields

- BYTE * pSMSFormat

- BYTE * pSMSOverIPNwInd
- BYTE * pPhoneCtxtURI

8.285.1 Detailed Description

Contains the parameters passed for SLQSSetIMSSMSConfigCallback by the device.

Parameters

pSMSFormat	• SMS format – 0 - 3GPP – 1 - 3GPP2
pSMSOverIPNw-Ind	• SMS over IP Network Indication Flag – TRUE - Mobile-Originated(MO) SMS turned on – FALSE - MO SMS turned off
pPhoneCtxtURI	• Phone context universal resource identifier • Length of this string can be of maximum 255 bytes

8.285.2 Field Documentation

8.285.2.1 BYTE* imsSMSConfigInfo::pPhoneCtxtURI

8.285.2.2 BYTE* imsSMSConfigInfo::pSMSFormat

8.285.2.3 BYTE* imsSMSConfigInfo::pSMSOverIPNwInd

8.286 imsUserConfigInfo Struct Reference

Data Fields

- BYTE * pIMSDomain

8.286.1 Detailed Description

Contains the parameters passed for SLQSSetIMSUserConfigCallback by the device.

Parameters

pIMSDomain	• IMS domain name • Length of this string can be of maximum 255 bytes

8.286.2 Field Documentation

8.286.2.1 BYTE* imsUserConfigInfo::pIMSDomain

8.287 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

8.287.1 Detailed Description

Contains the parameters passed for SLQSSetIMSVoIPConfigCallback by the device.

Parameters

pSessionExpiryTimer	• Session duration, in seconds
pMinSessionExpiryTimer	• Minimum allowed value for session expiry timer, in seconds
pAmrWbEnable	• Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio • Values: – True - Enabled – False - Disabled
pScrAmrEnable	• Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB) • Values: – True - Enabled – False - Disabled
pScrAmrWbEnable	• Flag to enable/disable SCR for AMR WB Audio • Values: – True - Enabled – False - Disabled

pAmrMode	• BitMask for AMR NB modes allowed • Values: – 0x1 - 4.75 kbps – 0x2 - 5.15 kbps – 0x4 - 5.9 kbps – 0x8 - 6.17 kbps – 0x10 - 7.4 kbps – 0x20 - 7.95 kbps – 0x40 - 10.2 kbps – 0x80 - 12.2 kbps
pAmrWBMode	• BitMask for AMR WB modes allowed • Values: – 0x1 - 6.60 kbps – 0x2 - 8.85 kbps – 0x4 - 12.65 kbps – 0x8 - 14.25 kbps – 0x10 - 15.85 kbps – 0x20 - 18.25 kbps – 0x40 - 19.85 kbps – 0x80 - 23.05 kbps – 0x100 - 23.85 kbps
pAmrOctet-Aligned	• Flag to indicate if the octet is aligned for AMR NB Audio • Values: – True - Aligned – False - Not aligned, Bandwidth Efficient mode
pAmrWBOctet-Aligned	• Flag to indicate if the octet is aligned for AMR WB Audio • Values: – True - Aligned – 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.

8.287.2 Field Documentation

8.287.2.1 BYTE* imsVoIPConfigInfo::pAmrMode

8.287.2.2 BYTE* imsVoIPConfigInfo::pAmrOctetAligned

8.287.2.3 BYTE* imsVoIPConfigInfo::pAmrWbEnable

8.287.2.4 WORD* imsVoIPConfigInfo::pAmrWBMode

8.287.2.5 BYTE* imsVoIPConfigInfo::pAmrWBOctetAligned

8.287.2.6 WORD* imsVoIPConfigInfo::pMinSessionExpiryTimer

8.287.2.7 WORD* imsVoIPConfigInfo::pRingBackTimer

8.287.2.8 WORD* imsVoIPConfigInfo::pRingingTimer

8.287.2.9 WORD* imsVoIPConfigInfo::pRTPRTCPInactTimer

8.287.2.10 BYTE* imsVoIPConfigInfo::pScrAmrEnable

8.287.2.11 BYTE* imsVoIPConfigInfo::pScrAmrWbEnable

8.287.2.12 WORD* imsVoIPConfigInfo::pSessionExpiryTimer

8.288 IndFieldsList Struct Reference

Data Fields

- BYTE indicationFieldsLen
- BYTE indicationFields [256]

8.288.1 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. • Format is same as request field.

8.288.2 Field Documentation

8.288.2.1 BYTE IndFieldsList::indicationFields[256]

8.288.2.2 BYTE IndFieldsList::indicationFieldsLen

8.289 infoInterFreq Struct Reference

Data Fields

- WORD earfcn
- BYTE threshXLow
- BYTE threshXHigh
- BYTE cell_resel_priority
- BYTE cells_len
- cellParams cellInterFreqParams [255]

8.289.1 Detailed Description

This structure contains information about the inter-frequency.

Parameters

earfcn	• E-UTRA absolute radio frequency channel number of the serving cell. • Range: 0 to 65535.
threshXLow	• Cell Srxlev low threshold. • Range: 0 to 31. • 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. • Range: 0 to 31. • When the serving cell exceeds thresh_serving_low, the value of an evaluated cell must be greater than this value to be considered for re-selection.
cell_resel_priority	• Cell re-selection priority • Range: 0 to 7. • 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.

8.289.2 Field Documentation

8.289.2.1 BYTE infoInterFreq::cell_resel_priority

8.289.2.2 cellParams infoInterFreq::cellInterFreqParams[255]

8.289.2.3 BYTE infoInterFreq::cells_len

8.289.2.4 WORD infoInterFreq::earfcn

8.289.2.5 BYTE infoInterFreq::threshXHigh

8.289.2.6 BYTE infoInterFreq::threshXLow

8.290 IOThresh Struct Reference

Data Fields

- BYTE IOThresListLen
- INT32 * pIOThresList

8.290.1 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 • 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 – IO is applicable only for HDR – Each IO threshold value is a signed 4 byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.290.2 Field Documentation

8.290.2.1 BYTE IOThresh::IOThresListLen

8.290.2.2 INT32* IOThresh::pIOThresList

8.291 IPv4Addr Struct Reference

Data Fields

- ULONG addr
- ULONG subnetMask

8.291.1 Detailed Description

This structure contains the IPv4 filter address

Parameters

addr	IPv4 address
subnetMask	A packet matches if: • (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

8.291.2 Field Documentation

8.291.2.1 ULONG IPv4Addr::addr

8.291.2.2 ULONG IPv4Addr::subnetMask

8.292 IPv6Addr Struct Reference

Data Fields

- BYTE addr [16]
- BYTE prefixLen

8.292.1 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.

8.292.2 Field Documentation

8.292.2.1 BYTE IPv6Addr::addr[16]

8.292.2.2 BYTE IPv6Addr::prefixLen

8.293 IPV6AddressInfo Struct Reference

Data Fields

- BYTE IPV6PrefixLen
- USHORT IPAddressV6 [8]

8.293.1 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 – 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.

8.293.2 Field Documentation

8.293.2.1 USHORT IPV6AddressInfo::IPAddressV6[8]

8.293.2.2 BYTE IPV6AddressInfo::IPV6PrefixLen

8.294 ipv6AddressInfo Struct Reference

Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

8.294.1 Detailed Description

Parameters

IPV6PrefixLen	Length of the received IPv6 address
IPAddressV6	IPv6 address(in network byte order)

8.294.2 Field Documentation

8.294.2.1 uint16_t ipv6AddressInfo::IPAddressV6[8]

8.294.2.2 uint8_t ipv6AddressInfo::IPV6PrefixLen

8.295 IPV6GWAddressInfo Struct Reference

Data Fields

- BYTE gwV6PrefixLen
- USHORT gwAddressV6 [8]

8.295.1 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.

8.295.2 Field Documentation

8.295.2.1 USHORT IPV6GWAddressInfo::gwAddressV6[8]

8.295.2.2 BYTE IPV6GWAddressInfo::gwV6PrefixLen

8.296 IPv6TrafCls Struct Reference

Data Fields

- BYTE val
- BYTE mask

8.296.1 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: • IPv6_filter_tc_val = 00101000 • IPv6_filter_tc_mask = 11111100 Filter will compare only the first 6 bits in IPv6_filter_traffic_class with the first 6 bits in the traffic class field of the IP packet; first 6 bits in the traffic class field of the IP packet must be 001010 to match filter; last 2 bits can be anything, since they are ignored by filtering

8.296.2 Field Documentation

8.296.2.1 BYTE IPv6TrafCls::mask

8.296.2.2 BYTE IPv6TrafCls::val

8.297 LibPackGPRSRequestedQoS Struct Reference

Data Fields

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

8.297.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

Parameters

precedence-Class	• Precedence class
delayClass	• Delay class
reliabilityClass	• Reliability class
peak-Throughput-Class	• Peak throughput class
mean-Throughput-Class	• Mean throughput class

8.297.2 Field Documentation

8.297.2.1 uint32_t LibPackGPRSRequestedQoS::delayClass

8.297.2.2 uint32_t LibPackGPRSRequestedQoS::meanThroughputClass

8.297.2.3 uint32_t LibPackGPRSRequestedQoS::peakThroughputClass

8.297.2.4 uint32_t LibPackGPRSRequestedQoS::precedenceClass

8.297.2.5 uint32_t LibPackGPRSRequestedQoS::reliabilityClass

8.298 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

8.298.1 Detailed Description

Parameters

extended	error
profile	info 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.
pProfileName	• One or more uint8_ts 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 uint8_ts.
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 – 0x00 - PDP-IP (IPv4) – 0x01 - PDP-PPP – 0x02 - PDP-IPV6 – 0x03 - PDP-IPV4V6
pPdpHdrComp-Type	• PDP header compression type – 0 - PDP header compression is OFF – 1 - Manufacturer preferred compression – 2 - PDP header compression based on RFC 1144 – 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095

pPdpDataCompType	• PDP data compression type – 0 - PDP data compression is OFF – 1 - Manufacturer preferred compression – 2 - V.42BIS data compression – 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 uint8_ts.
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 uint8_ts.
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 uint8_ts.

pAuthentication-Pref	• Authentication Preference – Bit map that indicates the authentication algorithm preference * Bit 0 - PAP preference · 0 - PAP is never performed · 1 - PAP may be performed * Bit 1 - CHAP preference · 0 - CHAP is never performed · 1 - CHAP may be performed * If more than one bit is set, then the device decides which authentication procedure is performed while setting up the data session. For example, the device may have a policy to select the most secure authentication mechanism.
pIPv4AddrPref	• IPv4 Address Preference
pPcscfAddr-UsingPCO	• P-CSCF Address using PCO Flag – 1 - (TRUE) implies request PCSCF address using PCO – 0 - (FALSE) implies do not request By default, this value is 0
pPdpAccess-ConFlag	• PDP access control flag – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission
pPcscfAddr-UsingDhcp	• P-CSCF address using DHCP – 1 - (TRUE) implies Request PCSCF address using DHCP – 0 - (FALSE) implies do not request By default, value is 0
pImCnFlag	• IM CN flag – 1 - (TRUE) implies request IM CN flag for this profile – 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 – 1 - (TRUE) implies this is secondary profile – 0 - (FALSE) implies this is not secondary profile

pPrimaryID	• PDP context primary ID • 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 – 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 – This enumerated value may be used to indicate the address allocation preference * 0 - NAS signaling is used for address allocation * 1 - DHCP is used for address allocation
pQosClassID	• 3GPP LTE QoS parameters
pAPNDisabled-Flag	• Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If set, the profile can not be used for making data calls • Any data call is failed locally • Values: – 0 - FALSE(default) – 1 - True • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
pPDNInactiv-Timeout	• Optional 4 uint8_ts indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

pAPNClass	• Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

8.298.2 Field Documentation

8.298.2.1 uint8_t* LibpackProfile3GPP::pAddrAllocPref

8.298.2.2 uint8_t* LibpackProfile3GPP::pAPNClass

8.298.2.3 uint8_t* LibpackProfile3GPP::pAPNDisabledFlag

8.298.2.4 uint8_t* LibpackProfile3GPP::pAPNName

8.298.2.5 uint16_t* LibpackProfile3GPP::pAPNnameSize

8.298.2.6 uint8_t* LibpackProfile3GPP::pAuthenticationPref

8.298.2.7 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSMinimumQoS

8.298.2.8 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSRequestedQos

8.298.2.9 uint8_t* LibpackProfile3GPP::pImCnFlag

8.298.2.10 uint32_t* LibpackProfile3GPP::pIPv4AddrPref

8.298.2.11 uint16_t* LibpackProfile3GPP::pIPv6AddPref

8.298.2.12 uint8_t* LibpackProfile3GPP::pPassword

8.298.2.13 uint16_t* LibpackProfile3GPP::pPasswordSize

8.298.2.14 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingDhcp

8.298.2.15 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingPCO

8.298.2.16 uint32_t* LibpackProfile3GPP::pPDNInactivTimeout

8.298.2.17 uint8_t* LibpackProfile3GPP::pPdpAccessConFlag

8.298.2.18 uint8_t* LibpackProfile3GPP::pPdpContext

8.298.2.19 uint8_t* LibpackProfile3GPP::pPdpDataCompType

8.298.2.20 uint8_t* LibpackProfile3GPP::pPdpHdrCompType

8.298.2.21 uint8_t* LibpackProfile3GPP::pPDPtype

8.298.2.22 uint32_t* LibpackProfile3GPP::pPriDNSIPv4AddPref

8.298.2.23 uint16_t* LibpackProfile3GPP::pPriDNSIPv6addpref

8.298.2.24 uint8_t* LibpackProfile3GPP::pPrimaryID

8.298.2.25 uint8_t* LibpackProfile3GPP::pProfilename

8.298.2.26 uint16_t* LibpackProfile3GPP::pProfilenameSize

8.298.2.27 LibPackQosClassID* LibpackProfile3GPP::pQosClassID

8.298.2.28 uint32_t* LibpackProfile3GPP::pSecDNSIPv4AddPref

8.298.2.29 uint16_t* LibpackProfile3GPP::pSecDNSIPv6addpref

8.298.2.30 uint8_t* LibpackProfile3GPP::pSecondaryFlag

8.298.2.31 LibPackTFTIDParams* LibpackProfile3GPP::pTFTID1Params

8.298.2.32 LibPackTFTIDParams* LibpackProfile3GPP::pTFTID2Params

8.298.2.33 LibPackUMTSQoS* LibpackProfile3GPP::pUMTSMinQoS

8.298.2.34 LibPackUMTSReqQoSSigInd* LibpackProfile3GPP::pUMTSMinQosSigInd

8.298.2.35 LibPackUMTSQoS* LibpackProfile3GPP::pUMTSReqQoS

8.298.2.36 LibPackUMTSReqQoSSigInd* LibpackProfile3GPP::pUMTSReqQoSSigInd

8.298.2.37 uint8_t* LibpackProfile3GPP::pUsername

8.298.2.38 uint16_t* LibpackProfile3GPP::pUsernameSize

8.299 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

8.299.1 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 – 1 - (TRUE)implies request DNS addresses from the PDSN – 0 - (FALSE)implies do not request DNS addresses from the PDSN – Default value is 1 (TRUE)
pPppSessClose-TimerDO	• PPP Session Close Timer for DO – 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 – Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down
pAllowLinger	• Allow/disallow lingering of interface – 1 -(TRUE) implies allow lingering – 0 -(FALSE) implies do not allow lingering
pLcpAckTimeout	• LCP ACK Timeout – Value of LCP ACK Timeout in milliseconds
pIpcpAck-Timeout	• IPCP ACK Timeout – Value of IPCP ACK Timeout in milliseconds
pAuthTimeout	• AUTH Timeout – 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 – 1 - PAP – 2 - CHAP – 3 - PAP or CHAP
pUserId	• User ID to be used during data network authentication • maximum length allowed is 127 uint8_ts; • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
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; • maximum length allowed is 127 uint8_ts • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
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 – 0 - Low (Low speed Service Options (SO15) only) – 1 - Medium (SO33 + low R-SCH) – 2 - High (SO33 + high R-SCH) – Default is 2
pAppType	• Application Type: – 0x00000001 - Default Application Type – 0x00000020 - LBS Application Type – 0x00000040 - Tethered Application Type – This parameter is not used while creating/modifying a profile

pDataMode	• Data Mode to use: – 0 - CDMA or HDR (Hybrid 1X/1xEV-DO) – 1 - CDMA Only (1X only) – 2 - HDR Only (1xEV-DO only) – Default is 0
pAppPriority	• Application Priority – Numerical 1 uint8_t value defining the application priority; higher value implies higher priority – This parameter is not used while creating/modifying a profile
pApnString	• String representing the Access Point Name • maximum length allowed is 100 uint8_ts • QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.
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: – 0 - IPv4 PDN Type – 1 - IPv6 PDN Type – 2 - IPv4 or IPv6 PDN Type – 3 - Unspecified PDN Type (implying no preference)
pIsPcscf-AddressNedded	• This boolean value is used to control if PCSCF address is requested from PDSN – 1 -(TRUE) implies request for PCSCF value from the PDSN – 0 -(FALSE) implies do not request for PCSCF value from the PDSN
pPrimaryV4Dns-Address	• IPv4 Primary DNS address – The Primary IPv4 DNS address that can be statically assigned to the UE
pSecondaryV4-DnsAddress	• IPv4 Secondary DNS address – The Secondary IPv4 DNS address that can be statically assigned to the UE
pPriV6Dns-Address	• Primary IPv6 DNS address – The Primary IPv6 DNS address that can be statically assigned to the UE
pSecV6Dns-Address	• Secondary IPv6 DNS address – The Secondary IPv6 DNS address that can be statically assigned to the UE

pRATType	• Optional 1 uint8_t Flag indicating RAT Type • Values: – 1 - HRPD – 2 - EHRPD – 3 - HRPD_EHRPD • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pAPNEnabled3GPP2	• Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If disabled, the profile can not be used for making data calls • Values: – 0 - Disabled – 1 - Enabled(default value) • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pPDNInactivTimeout3GPP2	• Optional 4 uint8_ts indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pAPNClass3GPP2	• Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().

8.299.2 Field Documentation

8.299.2.1 uint8_t* LibpackProfile3GPP2::pAllowLinger

8.299.2.2 uint8_t* LibpackProfile3GPP2::pAPNClass3GPP2

8.299.2.3 uint8_t* LibpackProfile3GPP2::pAPNEnabled3GPP2

8.299.2.4 uint8_t* LibpackProfile3GPP2::pApnString

8.299.2.5 uint16_t* LibpackProfile3GPP2::pApnStringSize

8.299.2.6 uint8_t* LibpackProfile3GPP2::pAppPriority

8.299.2.7 uint32_t* LibpackProfile3GPP2::pAppType

8.299.2.8 uint8_t* LibpackProfile3GPP2::pAuthPassword

8.299.2.9 uint16_t* LibpackProfile3GPP2::pAuthPasswordSize

8.299.2.10 uint8_t* LibpackProfile3GPP2::pAuthProtocol

8.299.2.11 uint8_t* LibpackProfile3GPP2::pAuthRetryCount

8.299.2.12 uint16_t* LibpackProfile3GPP2::pAuthTimeout

8.299.2.13 uint8_t* LibpackProfile3GPP2::pDataMode

8.299.2.14 uint8_t* LibpackProfile3GPP2::pDataRate

8.299.2.15 uint16_t* LibpackProfile3GPP2::pIpcpAckTimeout

8.299.2.16 uint8_t* LibpackProfile3GPP2::pIpcpCreqRetryCount

8.299.2.17 uint8_t* LibpackProfile3GPP2::pIsPcscfAddressNedded

8.299.2.18 uint16_t* LibpackProfile3GPP2::pLcpAckTimeout

8.299.2.19 uint8_t* LibpackProfile3GPP2::pLcpCreqRetryCount

8.299.2.20 uint8_t* LibpackProfile3GPP2::pNegoDnsSrvrPref

8.299.2.21 uint32_t* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2

8.299.2.22 uint8_t* LibpackProfile3GPP2::pPdnType

8.299.2.23 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimer1x

8.299.2.24 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimerDO

8.299.2.25 uint32_t* LibpackProfile3GPP2::pPrimaryV4DnsAddress

8.299.2.26 uint16_t* LibpackProfile3GPP2::pPriV6DnsAddress

8.299.2.27 uint8_t* LibpackProfile3GPP2::pRATType

8.299.2.28 uint32_t* LibpackProfile3GPP2::pSecondaryV4DnsAddress

8.299.2.29 uint16_t* LibpackProfile3GPP2::pSecV6DnsAddress

8.299.2.30 uint8_t* LibpackProfile3GPP2::pUserId

8.299.2.31 uint16_t* LibpackProfile3GPP2::pUserIdSize

8.300 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

8.300.1 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 – 0x00 - PDP-IP (IPv4) – 0x01 - PDP-PPP – 0x02 - PDP-IPV6 – 0x03 - PDP-IPV4V6

pPdpHdrCompType	• PDP header compression type – 0 - PDP header compression is OFF – 1 - Manufacturer preferred compression – 2 - PDP header compression based on RFC 1144 – 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095
pPdpDataCompType	• PDP data compression type – 0 - PDP data compression is OFF – 1 - Manufacturer preferred compression – 2 - V.42BIS data compression – 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 – Bit map that indicates the authentication algorithm preference * Bit 0 - PAP preference · 0 - PAP is never performed · 1 - PAP may be performed * Bit 1 - CHAP preference · 0 - CHAP is never performed · 1 - CHAP may be performed * If more than one bit is set, then the device decides which authentication procedure is performed while setting up the data session. For example, the device may have a policy to select the most secure authentication mechanism.
pIPv4AddrPref	• IPv4 Address Preference
pPcscfAddr-UsingPCO	• P-CSCF Address using PCO Flag – 1 - (TRUE) implies request PCSCF address using PCO – 0 - (FALSE) implies do not request By default, this value is 0
pPdpAccess-ConFlag	• PDP access control flag – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission
pPcscfAddr-UsingDhcp	• P-CSCF address using DHCP – 1 - (TRUE) implies Request PCSCF address using DHCP – 0 - (FALSE) implies do not request By default, value is 0
pImCnFlag	• IM CN flag – 1 - (TRUE) implies request IM CN flag for this profile – 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 – 1 - (TRUE) implies this is secondary profile – 0 - (FALSE) implies this is not secondary profile

pPrimaryID	• PDP context primary ID • 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 – 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 – This enumerated value may be used to indicate the address allocation preference * 0 - NAS signaling is used for address allocation * 1 - DHCP is used for address allocation
pQosClassID	• 3GPP LTE QoS parameters
pAPNDisabled-Flag	• Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If set, the profile can not be used for making data calls • Any data call is failed locally • Values: – 0 - FALSE(default) – 1 - True • 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 • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

pAPNClass	• Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

8.300.2 Field Documentation

8.300.2.1 uint8_t* LibPackprofile_3GPP::pAddrAllocPref

8.300.2.2 uint8_t* LibPackprofile_3GPP::pAPNClass

8.300.2.3 uint8_t* LibPackprofile_3GPP::pAPNDisabledFlag

8.300.2.4 uint8_t* LibPackprofile_3GPP::pAPNName

8.300.2.5 uint16_t* LibPackprofile_3GPP::pAPNnameSize

8.300.2.6 uint8_t* LibPackprofile_3GPP::pAuthenticationPref

8.300.2.7 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSMinimumQoS

8.300.2.8 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSRequestedQos

8.300.2.9 uint8_t* LibPackprofile_3GPP::pImCnFlag

8.300.2.10 uint32_t* LibPackprofile_3GPP::pIPv4AddrPref

8.300.2.11 uint16_t* LibPackprofile_3GPP::pIPv6AddPref

8.300.2.12 uint8_t* LibPackprofile_3GPP::pPassword

8.300.2.13 uint16_t* LibPackprofile_3GPP::pPasswordSize

8.300.2.14 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingDhcp

8.300.2.15 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingPCO

8.300.2.16 uint32_t* LibPackprofile_3GPP::pPDNInactivTimeout

8.300.2.17 uint8_t* LibPackprofile_3GPP::pPdpAccessConFlag

8.300.2.18 uint8_t* LibPackprofile_3GPP::pPdpContext

8.300.2.19 uint8_t* LibPackprofile_3GPP::pPdpDataCompType

8.300.2.20 uint8_t* LibPackprofile_3GPP::pPdpHdrCompType

8.300.2.21 uint8_t* LibPackprofile_3GPP::pPDPtype

8.300.2.22 uint32_t* LibPackprofile_3GPP::pPriDNSIPv4AddPref

8.300.2.23 uint16_t* LibPackprofile_3GPP::pPriDNSIPv6addpref

8.300.2.24 uint8_t* LibPackprofile_3GPP::pPrimaryID

8.300.2.25 uint8_t* LibPackprofile_3GPP::pProfilename

8.300.2.26 uint16_t* LibPackprofile_3GPP::pProfilenameSize

8.300.2.27 **LibPackQosClassID*** LibPackprofile_3GPP::pQosClassID

8.300.2.28 uint32_t* LibPackprofile_3GPP::pSecDNSIPv4AddPref

8.300.2.29 uint16_t* LibPackprofile_3GPP::pSecDNSIPv6addpref

8.300.2.30 uint8_t* LibPackprofile_3GPP::pSecondaryFlag

8.300.2.31 **LibPackTFTIDParams*** LibPackprofile_3GPP::pTFTID1Params

8.300.2.32 **LibPackTFTIDParams*** LibPackprofile_3GPP::pTFTID2Params

8.300.2.33 **LibPackUMTSQoS*** LibPackprofile_3GPP::pUMTSMinQoS

8.300.2.34 **LibPackUMTSReqQoSSigInd*** LibPackprofile_3GPP::pUMTSMinQosSigInd

8.300.2.35 **LibPackUMTSQoS*** LibPackprofile_3GPP::pUMTSReqQoS

8.300.2.36 **LibPackUMTSReqQoSSigInd*** LibPackprofile_3GPP::pUMTSReqQoSSigInd

8.300.2.37 uint8_t* LibPackprofile_3GPP::pUsername

8.300.2.38 uint16_t* LibPackprofile_3GPP::pUsernameSize

8.301 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

8.301.1 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 – 1 - (TRUE)implies request DNS addresses from the PDSN – 0 - (FALSE)implies do not request DNS addresses from the PDSN – Default value is 1 (TRUE)
pPppSessClose-TimerDO	• PPP Session Close Timer for DO – 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 – Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down
pAllowLinger	• Allow/disallow lingering of interface – 1 -(TRUE) implies allow lingering – 0 -(FALSE) implies do not allow lingering
pLcpAckTimeout	• LCP ACK Timeout – Value of LCP ACK Timeout in milliseconds
pIpcpAck-Timeout	• IPCP ACK Timeout – Value of IPCP ACK Timeout in milliseconds
pAuthTimeout	• AUTH Timeout – 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 – 1 - PAP – 2 - CHAP – 3 - PAP or CHAP
pUserId	• User ID to be used during data network authentication • maximum length allowed is 127 bytes; • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
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; • maximum length allowed is 127 bytes • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
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 – 0 - Low (Low speed Service Options (SO15) only) – 1 - Medium (SO33 + low R-SCH) – 2 - High (SO33 + high R-SCH) – Default is 2
pAppType	• Application Type: – 0x00000001 - Default Application Type – 0x00000020 - LBS Application Type – 0x00000040 - Tethered Application Type – This parameter is not used while creating/modifying a profile

pDataMode	• Data Mode to use: – 0 - CDMA or HDR (Hybrid 1X/1xEV-DO) – 1 - CDMA Only (1X only) – 2 - HDR Only (1xEV-DO only) – Default is 0
pAppPriority	• Application Priority – Numerical 1 uint8_t value defining the application priority; higher value implies higher priority – This parameter is not used while creating/modifying a profile
pApnString	• String representing the Access Point Name • maximum length allowed is 100 bytes • QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.
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: – 0 - IPv4 PDN Type – 1 - IPv6 PDN Type – 2 - IPv4 or IPv6 PDN Type – 3 - Unspecified PDN Type (implying no preference)
pIsPcscf-AddressNedded	• This boolean value is used to control if PCSCF address is requested from PDSN – 1 -(TRUE) implies request for PCSCF value from the PDSN – 0 -(FALSE) implies do not request for PCSCF value from the PDSN
pPrimaryV4Dns-Address	• IPv4 Primary DNS address – The Primary IPv4 DNS address that can be statically assigned to the UE
pSecondaryV4-DnsAddress	• IPv4 Secondary DNS address – The Secondary IPv4 DNS address that can be statically assigned to the UE
pPriV6Dns-Address	• Primary IPv6 DNS address – The Primary IPv6 DNS address that can be statically assigned to the UE
pSecV6Dns-Address	• Secondary IPv6 DNS address – The Secondary IPv6 DNS address that can be statically assigned to the UE

pRATType	• Optional 1 uint8_t Flag indicating RAT Type • Values: – 1 - HRPD – 2 - EHRPD – 3 - HRPD_EHRPD • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pAPNEnabled3GPP2	• Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If disabled, the profile can not be used for making data calls • Values: – 0 - Disabled – 1 - Enabled(default value) • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pPDNInactivTimeout3GPP2	• Optional 4 Bytes indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pAPNClass3GPP2	• Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().

8.301.2 Field Documentation

8.301.2.1 uint8_t* LibPackprofile_3GPP2::pAllowLinger

8.301.2.2 uint8_t* LibPackprofile_3GPP2::pAPNClass3GPP2

8.301.2.3 uint8_t* LibPackprofile_3GPP2::pAPNEnabled3GPP2

8.301.2.4 uint8_t* LibPackprofile_3GPP2::pApnString

8.301.2.5 uint16_t* LibPackprofile_3GPP2::pApnStringSize

8.301.2.6 uint8_t* LibPackprofile_3GPP2::pAppPriority

8.301.2.7 uint32_t* LibPackprofile_3GPP2::pAppType

8.301.2.8 uint8_t* LibPackprofile_3GPP2::pAuthPassword

8.301.2.9 uint16_t* LibPackprofile_3GPP2::pAuthPassword_tSize

8.301.2.10 uint8_t* LibPackprofile_3GPP2::pAuthProtocol

8.301.2.11 uint8_t* LibPackprofile_3GPP2::pAuthRetryCount

8.301.2.12 uint16_t* LibPackprofile_3GPP2::pAuthTimeout

8.301.2.13 uint8_t* LibPackprofile_3GPP2::pDataMode

8.301.2.14 uint8_t* LibPackprofile_3GPP2::pDataRate

8.301.2.15 uint16_t* LibPackprofile_3GPP2::pIpcpAckTimeout

8.301.2.16 uint8_t* LibPackprofile_3GPP2::pIpcpCreqRetryCount

8.301.2.17 uint8_t* LibPackprofile_3GPP2::pIsPcscfAddressNedded

8.301.2.18 uint16_t* LibPackprofile_3GPP2::pLcpAckTimeout

8.301.2.19 uint8_t* LibPackprofile_3GPP2::pLcpCreqRetryCount

8.301.2.20 uint8_t* LibPackprofile_3GPP2::pNegoDnsSrvrPref

8.301.2.21 uint32_t* LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2

8.301.2.22 uint8_t* LibPackprofile_3GPP2::pPdnType

8.301.2.23 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimer1x

8.301.2.24 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimerDO

8.301.2.25 uint32_t* LibPackprofile_3GPP2::pPrimaryV4DnsAddress

8.301.2.26 uint16_t* LibPackprofile_3GPP2::pPriV6DnsAddress

8.301.2.27 uint8_t* LibPackprofile_3GPP2::pRATType

8.301.2.28 uint32_t* LibPackprofile_3GPP2::pSecondaryV4DnsAddress

8.301.2.29 uint16_t* LibPackprofile_3GPP2::pSecV6DnsAddress

8.301.2.30 uint8_t* LibPackprofile_3GPP2::pUserId

8.301.2.31 uint16_t* LibPackprofile_3GPP2::pUserIdSize

8.302 LibPackQosClassID Struct Reference

Data Fields

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

8.302.1 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

8.302.2 Field Documentation

8.302.2.1 uint8_t LibPackQosClassID::gDlBitRate

8.302.2.2 uint32_t LibPackQosClassID::gUlBitRate

8.302.2.3 uint32_t LibPackQosClassID::maxDlBitRate

8.302.2.4 uint32_t LibPackQosClassID::maxUlBitRate

8.302.2.5 uint8_t LibPackQosClassID::QCI

8.303 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

8.303.1 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 – 4 - IPv4 – 6 - IPv6
sourceIP	• Source IP address – IPv4 - Fill the first 4 uint8_ts – 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

8.303.2 Field Documentation

8.303.2.1 uint16_t LibPackTFTIDParams::destPortRangeEnd

8.303.2.2 uint32_t LibPackTFTIDParams::destPortRangeStart

8.303.2.3 uint8_t LibPackTFTIDParams::eValid

8.303.2.4 uint8_t LibPackTFTIDParams::filterId

8.303.2.5 uint32_t LibPackTFTIDParams::flowLabel

8.303.2.6 uint32_t LibPackTFTIDParams::IPSECSPI

8.303.2.7 uint8_t LibPackTFTIDParams::ipVersion

8.303.2.8 uint8_t LibPackTFTIDParams::nextHeader

8.303.2.9 uint16_t* LibPackTFTIDParams::pSourceIP

8.303.2.10 uint8_t LibPackTFTIDParams::sourceIPMask

8.303.2.11 uint16_t LibPackTFTIDParams::srcPortRangeEnd

8.303.2.12 uint16_t LibPackTFTIDParams::srcPortRangeStart

8.303.2.13 uint16_t LibPackTFTIDParams::tosMask

8.304 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

8.304.1 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 • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 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 • 0x00 - Subscribe • 0x01 - delivery order on • 0x02 - delivery order off
maxSDUSize	• Maximum SDU size
sduErrorRatio	- SDU error ratio • Target value for fraction of SDUs lost or detected as erroneous. • 0x00 - Subscribe • 0x01 - 1∗10^(-2) • 0x02 - 7∗10^(-3) • 0x03 - 1∗10^(-3) • 0x04 - 1∗10^(-4) • 0x05 - 1∗10^(-5) • 0x06 - 1∗10^(-6) • 0x07 - 1∗10^(-1)
resBerRatio	- Residual bit error ratio • Target value for undetected bit error ratio in in the delivered SDUs. • 0x00 - Subscribe • 0x01 - 5∗10^(-2) • 0x02 - 1∗10^(-2) • 0x03 - 5∗10^(-3) • 0x04 - 4∗10^(-3) • 0x05 - 1∗10^(-3) • 0x06 - 1∗10^(-4) • 0x07 - 1∗10^(-5) • 0x08 - 1∗10^(-6) • 0x09 - 1∗10^(-8)

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

8.304.2 Field Documentation

8.304.2.1 uint8_t LibPackUMTSQoS::deliveryErrSDU

8.304.2.2 uint32_t LibPackUMTSQoS::grntDownlinkBitrate

8.304.2.3 uint32_t LibPackUMTSQoS::grntUplinkBitrate

8.304.2.4 uint32_t LibPackUMTSQoS::maxDownlinkBitrate

8.304.2.5 uint32_t LibPackUMTSQoS::maxSDUSize

8.304.2.6 uint32_t LibPackUMTSQoS::maxUplinkBitrate

8.304.2.7 uint8_t LibPackUMTSQoS::qosDeliveryOrder

8.304.2.8 uint8_t LibPackUMTSQoS::resBerRatio

8.304.2.9 uint8_t LibPackUMTSQoS::sduErrorRatio

8.304.2.10 uint8_t LibPackUMTSQoS::trafficClass

8.304.2.11 uint32_t LibPackUMTSQoS::trafficPriority

8.304.2.12 uint32_t LibPackUMTSQoS::transferDelay

8.305 LibPackUMTSReqQoSSigInd Struct Reference

Data Fields

- LibPackUMTSQoS UMTSReqQoS
- uint8_t SigInd

8.305.1 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 • TRUE - Signaling indication ON • FALSE - Signaling indication OFF

8.305.2 Field Documentation

8.305.2.1 uint8_t LibPackUMTSReqQoSSigInd::SigInd

8.305.2.2 LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS

8.306 lineCtrlInfo Struct Reference

Data Fields

- BYTE polarityIncluded
- BYTE toggleMode
- BYTE revPolarity
- BYTE pwrDenialTime

8.306.1 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

8.306.2 Field Documentation

8.306.2.1 BYTE lineCtrlInfo::polarityIncluded

8.306.2.2 BYTE lineCtrlInfo::pwrDenialTime

8.306.2.3 BYTE lineCtrlInfo::revPolarity

8.306.2.4 BYTE lineCtrlInfo::toggleMode

8.307 loc_BdsSV Struct Reference

Data Fields

- uint16_t id
- uint8_t mask

8.307.1 Detailed Description

This structure contains the BDS SV Info

Parameters

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

8.307.2 Field Documentation

8.307.2.1 uint16_t loc_BdsSV::id

8.307.2.2 uint8_t loc_BdsSV::mask

8.308 loc_BdsSVInfo Struct Reference

Data Fields

- uint8_t len
- loc_BdsSV * pSV

8.308.1 Detailed Description

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

Parameters

len	• Number of sets of the following elements: – gnssSvId – deleteSvInfoMask
pSV	• Pointer to struct loc_BdsSV. See loc_BdsSV for more information

8.308.2 Field Documentation

8.308.2.1 uint8_t loc_BdsSVInfo::len

8.308.2.2 loc_BdsSV* loc_BdsSVInfo::pSV

8.309 loc_CellDb Struct Reference

Data Fields

- uint32_t mask

8.309.1 Detailed Description

This structure contains the cell database

Parameters

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

8.309.2 Field Documentation

8.309.2.1 uint32_t loc_CellDb::mask

8.310 loc_ClkInfo Struct Reference

Data Fields

- uint32_t mask

8.310.1 Detailed Description

This structure contains the clock Info

Parameters

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

8.310.2 Field Documentation

8.310.2.1 uint32_t loc_ClkInfo::mask

8.311 loc_GnssData Struct Reference

Data Fields

- uint64_t mask

8.311.1 Detailed Description

This structure contains the GNSS data

Parameters

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

8.311.2 Field Documentation

8.311.2.1 uint64_t loc_GnssData::mask

8.312 loc_gpsTime Struct Reference

Data Fields

- uint16_t gpsWeek
- uint32_t gpsTimeOfWeekMs

8.312.1 Detailed Description

This structure contains GPS Time info.

Parameters

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

8.312.2 Field Documentation

8.312.2.1 uint32_t loc_gpsTime::gpsTimeOfWeekMs

8.312.2.2 uint16_t loc_gpsTime::gpsWeek

8.313 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

8.313.1 Detailed Description

This structure contains the Application Information

Parameters

appProvider-Length	• Length of the Application Provider
pAppProvider	• Application Provider • Depends upon the Length of application Provider
appNameLength	• Length of Application Name

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

8.313.2 Field Documentation

8.313.2.1 uint8_t loc_LocApplicationInfo::appNameLength

8.313.2.2 uint8_t loc_LocApplicationInfo::appProviderLength

8.313.2.3 uint8_t loc_LocApplicationInfo::appVersionLength

8.313.2.4 uint8_t loc_LocApplicationInfo::appVersionValid

8.313.2.5 uint8_t* loc_LocApplicationInfo::pAppName

8.313.2.6 uint8_t* loc_LocApplicationInfo::pAppProvider

8.313.2.7 uint8_t* loc_LocApplicationInfo::pAppVersion

8.314 loc_precisionDilution Struct Reference

Data Fields

- uint32_t PDOP
- uint32_t HDOP
- uint32_t VDOP

8.314.1 Detailed Description

This structure contains Dilution of precision associated with this position.

Parameters

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

HDOP	• Horizontal dilution of precision. • Range - 1 (highest accuracy) to 50 (lowest accuracy)
VDOP	• Vertical dilution of precision. • Range- 1 (highest accuracy) to 50 (lowest accuracy)

8.314.2 Field Documentation

8.314.2.1 uint32_t loc_precisionDilution::HDOP

8.314.2.2 uint32_t loc_precisionDilution::PDOP

8.314.2.3 uint32_t loc_precisionDilution::VDOP

8.315 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

8.315.1 Detailed Description

Contain fields in struct loc_satelliteInfo

Parameters

svListLen	• number of sets of the following elements: – validMask – system – gnssSvid – healthStatus – svStatus – svInfoMask – elevation – azimuth – snr

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

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

8.315.2 Field Documentation

8.315.2.1 float loc_satelliteInfo::azimuth

8.315.2.2 float loc_satelliteInfo::elevation

8.315.2.3 uint16_t loc_satelliteInfo::gnssSvId

8.315.2.4 uint8_t loc_satelliteInfo::healthStatus

8.315.2.5 float loc_satelliteInfo::snr

8.315.2.6 uint8_t loc_satelliteInfo::svInfoMask

8.315.2.7 uint8_t loc_satelliteInfo::svListLen

8.315.2.8 uint32_t loc_satelliteInfo::svStatus

8.315.2.9 uint32_t loc_satelliteInfo::system

8.315.2.10 uint32_t loc_satelliteInfo::validMask

8.316 loc_sensorDataUsage Struct Reference

Data Fields

- uint32_t usageMask
- uint32_t aidingIndicatorMask

8.316.1 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

8.316.2 Field Documentation

8.316.2.1 uint32_t loc_sensorDataUsage::aidingIndicatorMask

8.316.2.2 uint32_t loc_sensorDataUsage::usageMask

8.317 loc_SV Struct Reference

Data Fields

- uint16_t id
- uint32_t system
- uint8_t mask

8.317.1 Detailed Description

This structure contains the Delete LOC SV Info

Parameters

id	• LOC SV ID of the satellite whose data is to be deleted • Range: – For GPS: 1 to 32 – For SBAS: 33 to 64 – For GLONASS: 65 to 96
system	• Indicates to which constellation this loc_SV belongs • Valid values: – eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite – eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite – eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite – eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite – eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite – eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite

mask	• Indicates if the ephemeris or almanac for a satellite is to be deleted • Valid values: – 0x01 - DELETE_EPHEMERIS – 0x02 - DELETE_ALMANAC

8.317.2 Field Documentation

8.317.2.1 uint16_t loc_SV::id

8.317.2.2 uint8_t loc_SV::mask

8.317.2.3 uint32_t loc_SV::system

8.318 loc_SVInfo Struct Reference

Data Fields

- uint8_t len
- loc_SV * pSV

8.318.1 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: – gnssSvId – system – deleteSvInfoMask
pSV	• Pointer to struct loc_SV. See loc_SV for more information

8.318.2 Field Documentation

8.318.2.1 uint8_t loc_SVInfo::len

8.318.2.2 loc_SV* loc_SVInfo::pSV

8.319 loc_svUsedforFix Struct Reference

Data Fields

- uint8_t gnssSvUsedList_len
- uint16_t gnssSvUsedList [255]

8.319.1 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. • Following information is associated with each SV ID: – GPS - 1 to 32 – SBAS - 33 to 64 – GLONASS - 65 to 96 – QZSS - 193 to 197 – BDS - 201 to 237

8.319.2 Field Documentation

8.319.2.1 uint16_t loc_svUsedforFix::gnssSvUsedList[255]

8.319.2.2 uint8_t loc_svUsedforFix::gnssSvUsedList_len

8.320 LocApplicationInfo Struct Reference

Data Fields

- BYTE appProviderLength
- CHAR * pAppProvider
- BYTE appNameLength
- CHAR * pAppName
- BYTE appVersionValid
- CHAR appVersionLength
- CHAR * pAppVersion

8.320.1 Detailed Description

This structure contains the Application Information

Parameters

appProvider-Length	• Length of the Application Provider
pAppProvider	• Application Provider • Depends upon the Length of application Provider
appNameLength	• Length of Application Name

pAppName	• Application Name • Depends upon the Length of application Name
appVersionValid	• Specifies whether the application version string contains a valid value • 0x00 (FALSE) – Application version string is invalid • 0x01 (TRUE) – Application version string is valid
appVersion-Length	• Length of Application Version
pAppVersion	• Application Version • Depends upon the Length of application Version

8.320.2 Field Documentation

8.320.2.1 BYTE LocApplicationInfo::appNameLength

8.320.2.2 BYTE LocApplicationInfo::appProviderLength

8.320.2.3 CHAR LocApplicationInfo::appVersionLength

8.320.2.4 BYTE LocApplicationInfo::appVersionValid

8.320.2.5 CHAR* LocApplicationInfo::pAppName

8.320.2.6 CHAR* LocApplicationInfo::pAppProvider

8.320.2.7 CHAR* LocApplicationInfo::pAppVersion

8.321 LocDelAssDataReq Struct Reference

Data Fields

- SVInfo * pSVInfo
- GnssData * pGnssData
- CellDb * pCellDb
- ClkInfo * pClkInfo
- BdsSVInfo * pBdsSVInfo

8.321.1 Detailed Description

This structure contains LOC delete assist data request

Parameters

pSVInfo[IN]	• Optional parameter • Pointer to struct SVInfo. See SVInfo for more information

pGnssData[IN]	• Optional parameter • Pointer to struct GnssData. See GnssData for more information
pCellDb[IN]	• Optional parameter • Pointer to struct CellDb. See CellDb for more information
pClkInfo[IN]	• Optional parameter • Pointer to struct ClkInfo. See ClkInfo for more information
pBdsSVInfo[IN]	• Optional parameter • Pointer to struct BdsSVInfo. See BdsSVInfo for more information

8.321.2 Field Documentation

8.321.2.1 BdsSVInfo* LocDelAssDataReq::pBdsSVInfo

8.321.2.2 CellDb* LocDelAssDataReq::pCellDb

8.321.2.3 ClkInfo* LocDelAssDataReq::pClkInfo

8.321.2.4 GnssData* LocDelAssDataReq::pGnssData

8.321.2.5 SVInfo* LocDelAssDataReq::pSVInfo

8.322 LOCEventRegisterReqResp Struct Reference

Data Fields

- ULONGLONG eventRegister

8.322.1 Detailed Description

This structure contains the Parameter for RegisterEvents

Parameters

pEventRegMask	• Specifies the events that the control point is interested in receiving. -Values – 0x00000001 - to receive position report event indications – 0x00000002 - to receive satellite report event indications.These reports are sent at a 1 Hz rate. – 0x00000004 - to receive NMEA reports for position and satellites in view.The report is at a 1 Hz rate. – 0x00000008 - to receive NI Notify/Verify request event indications – 0x00000010 - to receive time injection request event indications. – 0x00000020 - to receive predicted orbits request event indications. – 0x00000040 - to receive position injection request event indications. – 0x00000080 - to receive engine state report event indications. – 0x00000100 - to receive fix session status report event indications. – 0x00000200 - to receive Wi-Fi position request event indications. – 0x00000400 - to receive notifications from the location engine indicating its readiness to accept data from the sensors(accelerometer, gyroscope, etc.). – 0x00000800 - to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock. – 0x00001000 - to receive Stationary Position Indicator (SPI) streaming report indications. – 0x00002000 - to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server. – 0x00004000 - to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited. – 0x00008000 - to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, e.g., if GPS is turned off or if the network is unavailable. – 0x00010000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence. – 0x00020000 - to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports. – 0x00040000 - to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data. – 0x00080000 - to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session. – 0x00100000 - to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing. – 0x00200000 - to receive Wi-Fi Access Point (AP) data inject request event indications. – 0x00400000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification. – 0x00800000 - to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.). – 0x01000000 - to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). – 0x02000000 - to receive satellite position reports as polynomials.Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_-CONSTELL_REPORT_CONFIG.

Note

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

8.322.2 Field Documentation

8.322.2.1 ULONGLONG LOCEventRegisterReqResp::eventRegister

8.323 LOCExtPowerStateReqResp Struct Reference

Data Fields

- ULONG extPowerState

8.323.1 Detailed Description

This structure contains the Parameter External Power Source State.

Parameters

pLOCEvent-RegisterReq-Resp	• Specifies the Power state; injected by the control point. • Values – 0 - Device is not connected to an external power source – 1 - Device is connected to an external power source – 2 - Unknown external power state

8.323.2 Field Documentation

8.323.2.1 ULONG LOCExtPowerStateReqResp::extPowerState

8.324 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

8.324.1 Detailed Description

This structure contains LOC Inject Position parameters

Parameters

pLatitude	• Optional parameter • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude • Note - This field must be specified together with pLongitude and pHorUncCircular.
pLongitude	• Optional parameter • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude • Note - This field must be specified together with pLatitude and pHorUncCircular.
pHorUncCircular	• Optional parameter • Horizontal position uncertainty. • Units - Meters • Note - This field must be specified together with pLatitude and pLongitude.
pHorConfidence	• Optional parameter • Horizontal confidence. • Units - Percent • Values – Valid Values - 1 to 99 – Invalid Values - 0, 101 to 255 – If 100 is received, reinterpret to 99 • 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 • Altitude With Respect to Ellipsoid. • Units - Meters • Values – Positive - height – Negative = depth
pAltitudeWrt-MeanSeaLevel	• Optional parameter • Altitude With Respect to Sea Level. • Units - Meters
pVertUnc	• Optional parameter • Vertical uncertainty. • Units - Meters • Note - This is mandatory if either pAltitudeWrtEllipsoid or pAltitudeWrtMeanSeaLevel is specified.
pVertConfidence	• Optional parameter • Vertical confidence. • Units - Percentage • Values – Valid Values - 0 to 99 – Invalid Values - 0, 100-256 – If 100 is received, reinterpret to 99 • 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 • UTC timestamp • Units - Milliseconds since Jan. 1, 1970
pTimestampAge	• Optional parameter • Position age, which is an estimate of how long ago this fix was made. • Units - Milliseconds
pPositionSrc	• Optional parameter • Source from which this position was obtained • Valid values – 0 - Position source is GNSS – 1 - Position source is Cell ID – 2 - Position source is Enhanced Cell ID – 3 - Position source is Wi-Fi – 4 - Position source is Terrestrial – 5 - Position source is GNSS Terrestrial Hybrid – 6 - Other sources • 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. – 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 • Horizontal position uncertainty (circular) without any optimization. • Units - Meters
pRawHor-Confidence	• Optional parameter • Horizontal confidence associated with raw horizontal uncertainty • Units: Percent • Values – Valid values - 1 to 99 – Invalid values - 0, 101 to 255 – If 100 is received, reinterpret to 99 • Note - This field must be specified together with raw horizontal uncertainty. If not specified when rawHorUncCircular is set, the default value is 50.

8.324.2 Field Documentation

8.324.2.1 altitudeSrcInfo∗ LocInjectPositionReq::pAltitudeSrcInfo

8.324.2.2 ULONG∗ LocInjectPositionReq::pAltitudeWrtEllipsoid

8.324.2.3 ULONG∗ LocInjectPositionReq::pAltitudeWrtMeanSeaLevel

8.324.2.4 BYTE* LocInjectPositionReq::pHorConfidence

8.324.2.5 ULONG* LocInjectPositionReq::pHorReliability

8.324.2.6 ULONG* LocInjectPositionReq::pHorUncCircular

8.324.2.7 ULONGLONG* LocInjectPositionReq::pLatitude

8.324.2.8 ULONGLONG* LocInjectPositionReq::pLongitude

8.324.2.9 ULONG* LocInjectPositionReq::pPositionSrc

8.324.2.10 BYTE* LocInjectPositionReq::pRawHorConfidence

8.324.2.11 ULONG* LocInjectPositionReq::pRawHorUncCircular

8.324.2.12 ULONG* LocInjectPositionReq::pTimestampAge

8.324.2.13 ULONGLONG* LocInjectPositionReq::pTimestampUtc

8.324.2.14 BYTE* LocInjectPositionReq::pVertConfidence

8.324.2.15 ULONG* LocInjectPositionReq::pVertReliability

8.324.2.16 ULONG* LocInjectPositionReq::pVertUnc

8.325 LocInjectSensorDataReq Struct Reference

Data Fields

- ULONG * pOpaqueIdentifier
- sensorData * pAcceleroData
- sensorData * pGyroData
- ULONG * pAcceleroTimeSrc
- ULONG * pGyroTimeSrc
- tempratureData * pAcceleroTempData
- tempratureData * pGyroTempData

8.325.1 Detailed Description

This structure contains parameters to inject sensor data into the GNSS location engine

Parameters

pOpaque-Identifier	• Opaque Identifier (Optional parameter) • 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 • Pointer to struct sensorData. See sensorData for more information

pGyroData	• Optional parameter • Pointer to struct sensorData. See sensorData for more information
pAcceleroTime-Src	• 3-Axis Accelerometer Data Time Source (Optional parameter) • The location service uses this field to identify the time reference used in the accelerometer data time stamps. • If not specified, the location service assumes that the time source for the accelerometer data is unknown. • Valid values – 0 - Sensor time source is unspecified – 1 - Time source is common between the sensors and the location engine
pGyroTimeSrc	• 3-Axis Gyroscope Data Time Source (Optional) • The location service uses this field to identify the time reference used in the gyroscope data time stamps. • If not specified, the location service assumes that the time source for the gyroscope data is unknown. • Valid values – 0 - Sensor time source is unspecified – 1 - Time source is common between the sensors and the location engine
pAcceleroTemp-Data	• Optional parameter • Pointer to struct tempratureData. See tempratureData for more information
pGyroTempData	• Optional parameter • Pointer to struct tempratureData. See tempratureData for more information

8.325.2 Field Documentation

8.325.2.1 sensorData* LocInjectSensorDataReq::pAcceleroData

8.325.2.2 tempratureData* LocInjectSensorDataReq::pAcceleroTempData

8.325.2.3 ULONG* LocInjectSensorDataReq::pAcceleroTimeSrc

8.325.2.4 sensorData* LocInjectSensorDataReq::pGyroData

8.325.2.5 tempratureData* LocInjectSensorDataReq::pGyroTempData

8.325.2.6 ULONG* LocInjectSensorDataReq::pGyroTimeSrc

8.325.2.7 ULONG* LocInjectSensorDataReq::pOpaqueIdentifier

8.326 LocSetCradleMountReq Struct Reference

Data Fields

- ULONG state
- BYTE * pConfidence

8.326.1 Detailed Description

This structure contains parameters to set current cradle mount configuration

Parameters

state	• Cradle Mount State • Valid values: – 0 - Device is mounted on the cradle – 1 - Device is not mounted on the cradle – 2 - Unknown cradle mount state
pConfidence	• Cradle Mount Confidence (Optional) • Confidence in the Cradle Mount state expressed as a percentage. • Range - 0 to 100

8.326.2 Field Documentation

8.326.2.1 BYTE* LocSetCradleMountReq::pConfidence

8.326.2.2 ULONG LocSetCradleMountReq::state

8.327 LOCStartReq Struct Reference

Data Fields

- BYTE SessionId
- ULONG * pRecurrenceType
- ULONG * pHorizontalAccuracyLvl
- ULONG * pIntermediateReportState
- ULONG * pMinIntervalTime
- struct LocApplicationInfo * pApplicationInfo
- ULONG * pConfigAltitudeAssumed

8.327.1 Detailed Description

This structure contains the LOC Start Request

Parameters

SessionId[IN]	• ID of the session as identified by the control point. • Range: 0 to 255

pRecurrence-Type[IN]	• Optional Parameter • Specifies the type of session in which the control point is interested. • Defaults to SINGLE. -Values – 1 - Request periodic position fixes – 2 - Request a single position fix
pHorizontal-AccuracyLvl[IN]	• Optional Parameter • Specifies the horizontal accuracy level required by the control point. • Defaults to LOW • Values – 1 - Low accuracy – 2 - Medium accuracy – 3 - High accuracy
pIntermediate-ReportState[IN]	• Optional Parameter • Specifies if the control point is interested in receiving intermediate reports. • ON by default. • Values – 1 - Intermediate reports are turned on – 2 - Intermediate reports are turned off
pMinInterval-Time[IN]	• Optional Parameter • Minimum time interval, specified by the control point, that must elapse between position reports. • Units - Milliseconds • Default - 1000 ms
LocApplication-Info[IN]	• Optional Parameter • LOC Application Parameters • See LocApplicationInfo for more information
pConfigAltitude-Assumed[IN]	• Optional Parameter • Configuration for Altitude Assumed Info in GNSS SV Info Event • Defaults to ENABLED. • Values – 1 - Enable Altitude Assumed information in GNSS SV Info Event – 2 - Disable Altitude Assumed information in GNSS SV Info Event

8.327.2 Field Documentation

8.327.2.1 struct LocApplicationInfo* LOCStartReq::pApplicationInfo

8.327.2.2 ULONG* LOCStartReq::pConfigAltitudeAssumed

8.327.2.3 ULONG* LOCStartReq::pHorizontalAccuracyLvl

8.327.2.4 ULONG* LOCStartReq::pIntermediateReportState

8.327.2.5 ULONG* LOCStartReq::pMinIntervalTime

8.327.2.6 ULONG* LOCStartReq::pRecurrenceType

8.327.2.7 BYTE LOCStartReq::SessionId

8.328 LOCStopReq Struct Reference

Data Fields

- BYTE sessionId

8.328.1 Detailed Description

This structure contains the LOC Stop Request

Parameters

sessionId	• ID of the session as identified by the control point. • Range: 0 to 255

8.328.2 Field Documentation

8.328.2.1 BYTE LOCStopReq::sessionId

8.329 LteCQIParm Struct Reference

Data Fields

- BYTE ValidityCW0
- BYTE CQIValueCW0
- BYTE ValidityCW1
- BYTE CQIValueCW1

8.329.1 Detailed Description

This structure contains information about the SLQSSwiGetLteCQI response parameters.

Parameters

pValidityCW0[OUT]	• Values – 0- Invalid. – 1- Valid.

pCQIValueCW0[OUT]	• Values – Range 0∼15
pValidityCW1[OUT]	• Values – 0- Invalid. – 1- Valid.
pCQIValueCW1[OUT]	• Values – Range 0∼15

8.329.2 Field Documentation

8.329.2.1 BYTE LteCQIParm::CQIValueCW0

8.329.2.2 BYTE LteCQIParm::CQIValueCW1

8.329.2.3 BYTE LteCQIParm::ValidityCW0

8.329.2.4 BYTE LteCQIParm::ValidityCW1

8.330 lteEARFCN Struct Reference

Data Fields

- BYTE status
- ULONG earfcn0
- ULONG earfcn1

8.330.1 Detailed Description

This structure contains the parameters for WCDMA UARCFN.

Parameters

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

8.330.2 Field Documentation

8.330.2.1 ULONG lteEARFCN::earfcn0

8.330.2.2 ULONG lteEARFCN::earfcn1

8.330.2.3 BYTE lteEARFCN::status

8.331 lteGsmCellInfo Struct Reference

Data Fields

- BYTE cellReselPriority
- BYTE threshGsmHigh
- BYTE threshGsmLow
- BYTE nccPermitted
- BYTE cells_len
- gsmCellInfo GsmCellInfo [255]

8.331.1 Detailed Description

This structure contains information about the LTE GSM Cell.

Parameters

cellReselPriority	• Priority of this frequency group. • Range: 0 to 7. • This field is only valid when ue_in_idle is TRUE.
threshGsmHigh	• Reselection threshold for high priority layers. • Range: 0 to 31. • This field is only valid when ue_in_idle is TRUE.
threshGsmLow	• Reselection threshold for low priority layers. • Range: 0 to 31. • 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. • Range: 0 to 255. • 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. • 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.

8.331.2 Field Documentation

8.331.2.1 BYTE lteGsmCellInfo::cellReselPriority

8.331.2.2 BYTE lteGsmCellInfo::cells_len

8.331.2.3 gsmCellInfo lteGsmCellInfo::GsmCellInfo[255]

8.331.2.4 BYTE lteGsmCellInfo::nccPermitted

8.331.2.5 BYTE lteGsmCellInfo::threshGsmHigh

8.331.2.6 BYTE lteGsmCellInfo::threshGsmLow

8.332 LTEInfo Struct Reference

Data Fields

- BYTE band
- BYTE bandwidth
- WORD RXChan
- WORD TXChan
- BYTE emmState
- BYTE emmSubState
- BYTE emmConnState

8.332.1 Detailed Description

Structure for storing the LTE information for the device.

Parameters

band	• LTE Band – 1 ∼ 41 (Band in decimal) – 0xFF - Invalid
bandwidth	• BandWidth. – 0x00 - 1.4 MHz – 0x01 - 3 MHz – 0x02 - 5 MHz – 0x03 - 10 MHz – 0x04 - 15 MHz – 0x05 - 20 MHz – 0x06 - Invalid – 0xFF - Unknown
RXChan	• RX channel number in decimal – 0xFFFF - Not Available

TXChan	• TX channel number in decimal – 0xFFFF - Not Available
emmState	• EMM State. – 0x00 - Deregistered – 0x01 - Reg Initiated – 0x02 - Registered – 0x03 - TAU Initiated – 0x04 - SR Initiated – 0x05 - Dereg Initiated – 0x06 - Invalid – 0xFF - Unknown
emmSubState	• EMM Sub State. – 0xFF - NOT Applicable • When EMM_state is 0x00: – 0x00 - No IMSI – 0x01 - PLMN Search – 0x02 - Attach Needed – 0x03 - No Cell – 0x04 - Attaching – 0x05 - Normal Service – 0x06 - Limited Service – 0x07 - Waiting for PDN • When EMM_state is 0x01: – 0x00 - Waiting for NW – 0x01 - Waiting for ESM • When EMM_state is 0x02: – 0x00 - Normal Service – 0x01 - Update Needed – 0x02 - Attempt Update – 0x03 - No Cell – 0x04 - PLMN Search – 0x05 - Limited Service – 0x06 - MM Update – 0x07 - IMSI Detach – 0x08 - Waiting for ESM
emmConnState	• EMM Connected Mode State. – 0x00 - RRC Idle – 0x01 - Waiting RRC Cfm – 0x02 - RRC Connected – 0x03 - RRC Releasing – 0xFF - Unknown

8.332.2 Field Documentation

8.332.2.1 BYTE LTEInfo::band

8.332.2.2 BYTE LTEInfo::bandwidth

8.332.2.3 BYTE LTEInfo::emmConnState

8.332.2.4 BYTE LTEInfo::emmState

8.332.2.5 BYTE LTEInfo::emmSubState

8.332.2.6 WORD LTEInfo::RXChan

8.332.2.7 WORD LTEInfo::TXChan

8.333 LTEInfoInterfreq Struct Reference

Data Fields

- BYTE ueInIdle
- BYTE freqsLen
- infoInterFreq InfoInterfreq [255]

8.333.1 Detailed Description

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

Parameters

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

8.333.2 Field Documentation

8.333.2.1 BYTE LTEInfoInterfreq::freqsLen

8.333.2.2 infoInterFreq LTEInfoInterfreq::InfoInterfreq[255]

8.333.2.3 BYTE LTEInfoInterfreq::ueInIdle

8.334 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]

8.334.1 Detailed Description

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

Parameters

ueInIdle	• TRUE if the UE is in Idle mode, otherwise FALSE. – 0xFF - Not Available
plmn[PLMN_LENGTH]	• PLMN ID coded as octet 3, 4, and 5.
tac	• Tracking area code. – 0xFFFF - Not Available
globalCellId	• Global cell ID in the system information block. – 0xFFFFFFFF - Not Available
earfcn	• E-UTRA absolute radio frequency channel number of the serving cell. • Range: 0 to 65535. – 0xFFFF - Not Available
servingCellId	• LTE serving cell ID. • Range: 0 to 503. • This is the cell ID of the serving cell and can be found in the cell list. – 0xFFFF - Not Available
cellReselPriority	• Priority for serving frequency. • Range: 0 to 7. • This field is only valid when ue_in_idle is TRUE. – 0xFF - Not Available

sNonIntraSearch	• S non-intra search threshold to control non-intrafrequency searches. • Range: 0 to 31. • This field is only valid when ue_in_idle is TRUE. – 0xFF - Not Available
threshServing-Low	• Serving cell low threshold. • Range: 0 to 31. • This field is only valid when ue_in_idle is TRUE. – 0xFF - Not Available
sIntraSearch	• S Intra search threshold. • Range: 0 to 31. • The current cell measurement must fall below this threshold to consider intrafrequency for reselection. • This field is only valid when ue_in_idle is TRUE. – 0xFF - Not Available
cellsLen	• Provides the number of set of cell params. • If 0(zero), then no information follows it.
CellParams[MAX_DESCRIPTION_LENGTH]	• See cellParams for more information.

8.334.2 Field Documentation

8.334.2.1 cellParams LTEInfoIntrafreq::CellParams[255]

8.334.2.2 BYTE LTEInfoIntrafreq::cellReselPriority

8.334.2.3 BYTE LTEInfoIntrafreq::cellsLen

8.334.2.4 WORD LTEInfoIntrafreq::earfcn

8.334.2.5 ULONG LTEInfoIntrafreq::globalCellId

8.334.2.6 BYTE LTEInfoIntrafreq::plmn[3]

8.334.2.7 WORD LTEInfoIntrafreq::servingCellId

8.334.2.8 BYTE LTEInfoIntrafreq::sIntraSearch

8.334.2.9 BYTE LTEInfoIntrafreq::sNonIntraSearch

8.334.2.10 WORD LTEInfoIntrafreq::tac

8.334.2.11 BYTE LTEInfoIntrafreq::threshServingLow

8.334.2.12 BYTE LTEInfoIntrafreq::ueInIdle

8.335 LTEInfoNeighboringGSM Struct Reference

Data Fields

- BYTE ueInIdle
- BYTE freqsLen
- lteGsmCellInfo LteGsmCellInfo [255]

8.335.1 Detailed Description

This structure contains information about the LTE Neighboring GSM Network.

Parameters

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

8.335.2 Field Documentation

8.335.2.1 BYTE LTEInfoNeighboringGSM::freqsLen

8.335.2.2 lteGsmCellInfo LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.335.2.3 BYTE LTEInfoNeighboringGSM::ueInIdle

8.336 LTEInfoNeighboringWCDMA Struct Reference

Data Fields

- BYTE ueInIdle
- BYTE freqsLen
- lteWcdmaCellInfo LTEWCDMACellInfo [255]

8.336.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA Network.

Parameters

ueInIdle	• TRUE if the UE is in Idle mode, otherwise FALSE. – 0xFF - Not Available
freqsLen	• Provides the number of set of LTE WCDMA cell information. • If 0(zero), then no information follows it.
LTEWCDMA-CellInfo[MAX_D-ESCRIPTION_L-ENGTH]	• See lteWcdmaCellInfo for more information.

8.336.2 Field Documentation

8.336.2.1 BYTE LTEInfoNeighboringWCDMA::freqsLen

8.336.2.2 lteWcdmaCellInfo LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.336.2.3 BYTE LTEInfoNeighboringWCDMA::ueInIdle

8.337 LteNasReleaseInfo_s Struct Reference

Data Fields

- BYTE nas_release
- BYTE nas_major
- BYTE nas_minor

8.337.1 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

8.337.2 Field Documentation

8.337.2.1 BYTE LteNasReleaseInfo_s::nas_major

8.337.2.2 BYTE LteNasReleaseInfo_s::nas_minor

8.337.2.3 BYTE LteNasReleaseInfo_s::nas_release

8.338 ltePCI Struct Reference

Data Fields

- BYTE status
- ULONG earfcn
- ULONG pci

8.338.1 Detailed Description

This structure contains the parameters for WCDMA UARCFN.

Parameters

status	• 0 - Disable • 1 - Enable
earfcn	• UARFCN to which UE is locked
pci	• PCI to which the UE is locked

8.338.2 Field Documentation

8.338.2.1 ULONG ltePCI::earfcn

8.338.2.2 ULONG ltePCI::pci

8.338.2.3 BYTE ltePCI::status

8.339 lteRsrpinformation Struct Reference

Data Fields

- SHORT rsrplevel

8.339.1 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).

8.339.2 Field Documentation

8.339.2.1 SHORT lteRsrpinformation::rsrplevel

8.340 LTERSRPThresh Struct Reference

Data Fields

- BYTE LTERSRPThreshListLen
- WORD * pLTERSRPThreshList

8.340.1 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) • Maximum of 32 values • Range for RSRP values: -140 to -44 (in dBm).

8.340.2 Field Documentation

8.340.2.1 BYTE LTERSRPThresh::LTERSRPThreshListLen

8.340.2.2 WORD* LTERSRPThresh::pLTERSRPThreshList

8.341 LTERSRQThresh Struct Reference

Data Fields

- BYTE LTERSRQThreshListLen
- WORD * pLTERSRQThreshList

8.341.1 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) • Maximum of 32 values. • Range for RSRQ values: -20 to -3 (in dBm)

8.341.2 Field Documentation

8.341.2.1 BYTE LTERSRQThresh::LTERSRQThreshListLen

8.341.2.2 WORD* LTERSRQThresh::pLTERSRQThreshList

8.342 LTERSSIThresh Struct Reference

Data Fields

- BYTE LTERSSIThreshListLen
- WORD * pLTERSSIThreshList

8.342.1 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) • Maximum of 32 values. • Range for RSSI values: -120 to 0 (in dBm)

8.342.2 Field Documentation

8.342.2.1 BYTE LTERSSIThresh::LTERSSIThreshListLen

8.342.2.2 WORD* LTERSSIThresh::pLTERSSIThreshList

8.343 LteSccRxInfoResp Struct Reference

Data Fields

- SccRxInfo * pSccRxInfo

8.343.1 Detailed Description

This structure contains information about the SLQSSwiGetLteSccRxInfo response parameters.

Parameters

pSccRxInfo	[Optional] • See SccRxInfo for more information

8.343.2 Field Documentation

8.343.2.1 SccRxInfo* LteSccRxInfoResp::pSccRxInfo

8.344 LTESigRptCfg Struct Reference

Data Fields

- BYTE rptRate
- BYTE avgPeriod

8.344.1 Detailed Description

This structure contains LTE Signal Report Config parameters.

Parameters

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

8.344.2 Field Documentation

8.344.2.1 BYTE LTESigRptCfg::avgPeriod

8.344.2.2 BYTE LTESigRptCfg::rptRate

8.345 LTESigRptConfig Struct Reference

Data Fields

- BYTE rptRate
- BYTE avgPeriod

8.345.1 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 • 0 - Report using the default configuration • 1 - Report every 1 sec • 2 - Report every 2 sec • 3 - Report every 3 sec • 4 - Report every 4 sec • 5 - Report every 5 sec
avgPeriod	• Averaging period to be used for the LTE signal. • Values – 0 - Average using the default configuration – 1 - Average over 1 sec – 2 - Average over 2 sec – 3 - Average over 3 sec – 4 - Average over 4 sec – 5 - Average over 5 sec – 6 - Average over 6 sec – 7 - Average over 7 sec – 8 - Average over 8 sec – 9 - Average over 9 sec – 10 - Average over 10 sec

8.345.2 Field Documentation

8.345.2.1 BYTE LTESigRptConfig::avgPeriod

8.345.2.2 BYTE LTESigRptConfig::rptRate

8.346 lteSnrinformation Struct Reference

Data Fields

- SHORT snrlevel

8.346.1 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.

8.346.2 Field Documentation

8.346.2.1 SHORT lteSnrinformation::snrlevel

8.347 LTESNRThresh Struct Reference

Data Fields

- BYTE LTESNRThresListLen
- SHORT * pLTESNRThresList

8.347.1 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 • 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 – For LTE, each SNR threshold value is a signed 2 Byte value – Maximum number of threshold values is 16 – At least one value must be specified – 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

8.347.2 Field Documentation

8.347.2.1 BYTE LTESNRThresh::LTESNRThresListLen

8.347.2.2 SHORT* LTESNRThresh::pLTESNRThresList

8.348 LTESNRThreshold Struct Reference

Data Fields

- BYTE LTESNRThreshListLen
- WORD * pLTESNRThreshList

8.348.1 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) • Maximum of 32 values • Range for SNR values: -20 to 30 (in dB).

8.348.2 Field Documentation

8.348.2.1 BYTE LTESNRThreshold::LTESNRThreshListLen

8.348.2.2 WORD* LTESNRThreshold::pLTESNRThreshList

8.349 LTESSInfo Struct Reference

Data Fields

- INT8 rssi
- INT8 rsrq
- SHORT rsrp
- SHORT snr

8.349.1 Detailed Description

This structure contains the parameters for LTE Signal Strength Information

Parameters

rssi	• RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal. – For CDMA and UMTS, this indicates forward link pilot Ec – For GSM, this indicates received signal strength
rsrq	• RSRQ value in dB (signed integer value) as measured by L1. • Range: -3 to -20 (-3 means -3 dB, -20 means -20 dB).
rsrp	• Current RSRP in dBm as measured by L1. • 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,

8.349.2 Field Documentation

8.349.2.1 SHORT LTESSInfo::rsrp

8.349.2.2 INT8 LTESSInfo::rsrq

8.349.2.3 INT8 LTESSInfo::rssi

8.349.2.4 SHORT LTESSInfo::snr

8.350 lteSSInfo Struct Reference

Data Fields

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

8.350.1 Detailed Description

Parameters

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

8.350.2 Field Documentation

8.350.2.1 int16_t lteSSInfo::rsrp

8.350.2.2 int8_t lteSSInfo::rsrq

8.350.2.3 int8_t lteSSInfo::rssi

8.350.2.4 int16_t lteSSInfo::snr

8.351 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

8.351.1 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.. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
lac	• Location area code. • Only applies to 3GPP. – 0xFFFF - Not Available
cellIdValid	• Indicates whether the cell ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
cellId	• Cell ID. – 0xFFFFFFFF - Not Available
regRejectInfo-Valid	• Indicates whether the registration reject information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
rejectSrvDomain	• Type of service domain in which the registration is rejected. – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
rejCause	• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. – 0xFF - Not Available

networkIdValid	• Indicates whether the network ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
MCC[PLMN_LENGTH]	• Mobile Country Code. • MCC digits in ASCII characters
MNC[PLMN_LENGTH]	• Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.
tacValid	• Indicates whether tracking area code is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
tac	• Tracking area code. • Only applicable for LTE. – 0xFFFF - Not Available

8.351.2 Field Documentation

8.351.2.1 ULONG LTESysInfo::cellId

8.351.2.2 BYTE LTESysInfo::cellIdValid

8.351.2.3 WORD LTESysInfo::lac

8.351.2.4 BYTE LTESysInfo::lacValid

8.351.2.5 BYTE LTESysInfo::MCC[3]

8.351.2.6 BYTE LTESysInfo::MNC[3]

8.351.2.7 BYTE LTESysInfo::networkIdValid

8.351.2.8 BYTE LTESysInfo::regRejectInfoValid

8.351.2.9 BYTE LTESysInfo::rejCause

8.351.2.10 BYTE LTESysInfo::rejectSrvDomain

8.351.2.11 sysInfoCommon LTESysInfo::sysInfoLTE

8.351.2.12 WORD LTESysInfo::tac

8.351.2.13 BYTE LTESysInfo::tacValid

8.352 lteWcdmaCellInfo Struct Reference

Data Fields

- WORD uarfcn
- BYTE cellReselPriority
- WORD threshXhigh
- WORD threshXlow
- BYTE cellsLen
- wcdmaCellInfo WCDMACellInfo [255]

8.352.1 Detailed Description

This structure contains information about the LTE WCDMA Cell.

Parameters

uarfcn	• WCDMA layer frequency. • Range: 0 to 16383.
cellReselPriority	• Cell re-selection priority. • Range: 0 to 7. • This field is only valid when ue_in_idle is TRUE.
threshXhigh	• Re-selection low threshold. • Range: 0 to 31. • This field is only valid when ue_in_idle is TRUE.
threshXlow	• Re-selection high threshold. • Range: 0 to 31. • 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.

8.352.2 Field Documentation

8.352.2.1 BYTE lteWcdmaCellInfo::cellReselPriority

8.352.2.2 BYTE lteWcdmaCellInfo::cellsLen

8.352.2.3 WORD lteWcdmaCellInfo::threshXhigh

8.352.2.4 WORD lteWcdmaCellInfo::threshXlow

8.352.2.5 WORD lteWcdmaCellInfo::uarfcn

8.352.2.6 wcdmaCellInfo lteWcdmaCellInfo::WCDMACellInfo[255]

8.353 messageModeTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSMessageModeInfo MessageModeInfo

8.353.1 Detailed Description

Parameters

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

8.353.2 Field Documentation

8.353.2.1 sMSMessageModeInfo messageModeTlv::MessageModeInfo

8.353.2.2 uint8_t messageModeTlv::TlvPresent

8.354 messageWaitingInfoContent Struct Reference

Data Fields

- BYTE msgType
- BYTE activeInd
- BYTE msgCount

8.354.1 Detailed Description

This structure contains message waiting information per instance

Parameters

msgType	• Message type – 0x00 - MWI_MESSAGE_TYPE_VOICMAIL - Voicemail – 0x01 - MWI_MESSAGE_TYPE_FAX - Fax – 0X02 - MWI_MESSAGE_TYPE_EMAIL - Email – 0x03 - MWI_MESSAGE_TYPE_OTHER - Other – 0x04 - MWI_MESSAGE_TYPE_VIDEOMAIL - Videomail
activeInd	• Indicates whether the indication is active – 0x00 - Inactive – 0x01 - Active
msgCount	• Number of messages

8.354.2 Field Documentation

8.354.2.1 BYTE messageWaitingInfoContent::activeInd

8.354.2.2 BYTE messageWaitingInfoContent::msgCount

8.354.2.3 BYTE messageWaitingInfoContent::msgType

8.355 minBasedIMSI Struct Reference

Data Fields

- BYTE mccM [3]
- WORD imsiM1112
- BYTE imsiMS1 [7]
- BYTE imsiMS2 [3]

8.355.1 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 – 0xFFFF - Not Available
imsiMS1	• ASCII character representation of IMSI_M_S1 value
imsiMS2	• ASCII character representation of IMSI_M_S2 value

8.355.2 Field Documentation

8.355.2.1 WORD minBasedIMSI::imsiM1112

8.355.2.2 BYTE minBasedIMSI::imsiMS1[7]

8.355.2.3 BYTE minBasedIMSI::imsiMS2[3]

8.355.2.4 BYTE minBasedIMSI::mccM[3]

8.356 mitigationDevList Struct Reference

Data Fields

- BYTE mitigationDevIdLen
- CHAR mitigationDevId [255]
- BYTE maxMitigationLevel

8.356.1 Detailed Description

This structure contains mitigation devices list

Parameters

mitigationDevId-Len	• Number of sets of the following elements – mitigation_dev_id
mitigationDevId	• Mitigation device ID.
maxMitigation-Level	• Maximum valid mitigation level. • Valid range - 0 to max_mitigation_level.

8.356.2 Field Documentation

8.356.2.1 BYTE mitigationDevList::maxMitigationLevel

8.356.2.2 CHAR mitigationDevList::mitigationDevId[255]

8.356.2.3 BYTE mitigationDevList::mitigationDevIdLen

8.357 MNRInfo Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- ULONG rat

8.357.1 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. – 0x04 - RADIO_IF_GSM – 0x05 - RADIO_IF_UMTS – 0x08 - RADIO_IF_LTE

8.357.2 Field Documentation

8.357.2.1 WORD MNRInfo::mcc

8.357.2.2 WORD MNRInfo::mnc

8.357.2.3 ULONG MNRInfo::rat

8.358 ModifyProfileIn Struct Reference

Data Fields

- BYTE * pProfileID
- BYTE * pProfileType
- QmiProfileInfo curProfile

8.358.1 Detailed Description

This structure contains input parameters for SLQSModifyProfile

Parameters

ProfileID	• 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
ProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2 – NULL is not allowed

curProfile	• Contains Union of profile(3GPP/3GPP2) structures

8.358.2 Field Documentation

8.358.2.1 QmiProfileInfo ModifyProfileIn::curProfile

8.358.2.2 BYTE* ModifyProfileIn::pProfileID

8.358.2.3 BYTE* ModifyProfileIn::pProfileType

8.359 ModifyProfileOut Struct Reference

Data Fields

- USHORT * pExtErrorCode

8.359.1 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. • Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERN-AL is returned by device. • See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description.

8.359.2 Field Documentation

8.359.2.1 USHORT* ModifyProfileOut::pExtErrorCode

8.360 msgWaitingInfo Struct Reference

Data Fields

- BYTE numInstances
- messageWaitingInfoContent msgWaitInfo [0xFF]

8.360.1 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. – See messageWaitingInfoContent for more information.

8.360.2 Field Documentation

8.360.2.1 messageWaitingInfoContent msgWaitingInfo::msgWaitInfo[0xFF]

8.360.2.2 BYTE msgWaitingInfo::numInstances

8.361 namName Struct Reference

Data Fields

- BYTE namNameLen
- BYTE namName [12]

8.361.1 Detailed Description

This structure contains the parameters for NAM Name Information

Parameters

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

8.361.2 Field Documentation

8.361.2.1 BYTE namName::namName[12]

8.361.2.2 BYTE namName::namNameLen

8.362 nas_acqOrderPref Struct Reference

Data Fields

- uint8_t acqOrdeLen
- uint8_t * pAcqOrder

8.362.1 Detailed Description

Contain the Acquisition Order Preference.

Parameters

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

8.362.2 Field Documentation

8.362.2.1 uint8_t nas_acqOrderPref::acqOrdeLen

8.362.2.2 uint8_t* nas_acqOrderPref::pAcqOrder

8.363 nas_ActPilotPNElement Struct Reference

Data Fields

- uint16_t ActSetPilotPN
- uint8_t ActSetPilotPNStrength

8.363.1 Detailed Description

Parameters

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

8.363.2 Field Documentation

8.363.2.1 uint16_t nas_ActPilotPNElement::ActSetPilotPN

8.363.2.2 uint8_t nas_ActPilotPNElement::ActSetPilotPNStrength

8.364 nas_AddCDMASysInfo Struct Reference

Data Fields

- uint16_t geoSysIdx
- uint16_t regPrd

8.364.1 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. • When the system index is not known, 0xFFFF is used.
regPrd	• Registration period after the CDMA system is acquired. • When the CDMA registration period is not valid, 0xFFFF is used.

8.364.2 Field Documentation

8.364.2.1 uint16_t nas_AddCDMASysInfo::geoSysIdx

8.364.2.2 uint16_t nas_AddCDMASysInfo::regPrd

8.365 nas_AddSysInfo Struct Reference

Data Fields

- uint16_t geoSysIdx
- uint32_t cellBroadcastCap

8.365.1 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. • When the system index is not known, 0xFFFF is used.
cellBroadcast-Cap	• Cell broadcast capability of the serving system. • When the CDMA registration period is not valid, 0xFFFF is used. – NAS_CELL_BROADCAST_CAP_UNKNOWN - Cell broadcast support is unknown – NAS_CELL_BROADCAST_CAP_OFF - Cell broadcast is not supported – NAS_CELL_BROADCAST_CAP_ON - Cell broadcast is supported

8.365.2 Field Documentation

8.365.2.1 uint32_t nas_AddSysInfo::cellBroadcastCap

8.365.2.2 uint16_t nas_AddSysInfo::geoSysIdx

8.366 nas_CallBarringSysInfo Struct Reference

Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

8.366.1 Detailed Description

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

Parameters

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

8.366.2 Field Documentation

8.366.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

8.366.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

8.367 nas_callBarStatus Struct Reference

Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

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

8.367.2 Field Documentation

8.367.2.1 uint32_t nas_callBarStatus::csBarStatus

8.367.2.2 uint32_t nas_callBarStatus::psBarStatus

8.368 nas_CDMAChannel Struct Reference

Data Fields

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

8.368.1 Detailed Description

Parameters

priChA	• A Channel number for the primary carrier. – 0xFFFF - Not Available
priChB	• B Channel number for the primary carrier. – 0xFFFF - Not Available
secChA	• A Channel number for the secondary carrier. – 0xFFFF - Not Available

secChB	• B Channel number for the secondary carrier. – 0xFFFF - Not Available

8.368.2 Field Documentation

8.368.2.1 uint16_t nas_CDMAChannel::priChA

8.368.2.2 uint16_t nas_CDMAChannel::priChB

8.368.2.3 uint16_t nas_CDMAChannel::secChA

8.368.2.4 uint16_t nas_CDMAChannel::secChB

8.369 nas_CDMAECIOThresh Struct Reference

Data Fields

- uint8_t CDMAECIOThreshListLen
- int16_t * pCDMAECIOThreshList

8.369.1 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) • Maximum of 32 values. Range for ECIO values: -31.5 to 0 (in dB)

8.369.2 Field Documentation

8.369.2.1 uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen

8.369.2.2 int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList

8.370 nas_CDMAInfo Struct Reference

Data Fields

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

- uint32_t baseLong

8.370.1 Detailed Description

This structure contains information about the CDMA Network.

Parameters

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

8.370.2 Field Documentation

8.370.2.1 uint16_t nas_CDMAInfo::baseId

8.370.2.2 uint32_t nas_CDMAInfo::baseLat

8.370.2.3 uint32_t nas_CDMAInfo::baseLong

8.370.2.4 uint16_t nas_CDMAInfo::nid

8.370.2.5 uint16_t nas_CDMAInfo::refpn

8.370.2.6 uint16_t nas_CDMAInfo::sid

8.371 nas_CDMARSSIThresh Struct Reference

Data Fields

- uint8_t CDMARSSIThreshListLen
- int16_t * pCDMARSSIThreshList

8.371.1 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) • maximum of 32 values. • Range for RSSI values:-105 to -21 (in dBm).

8.371.2 Field Documentation

8.371.2.1 uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen

8.371.2.2 int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList

8.372 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]

8.372.1 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. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
isSysPrlMatch	• Indicates whether the system is in a PRL. • Only applies to CDMA/HDR. – 0x00 - System is not in a PRL – 0x01 - System is in a PRL – 0xFF - Not Available • If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL. • If the system is in a PRL, roam_status is set to the value based on the standard specification.
pRevInUseValid	• Indicates whether the P_Rev in use is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
pRevInUse	• The lesser of the base station P_Rev and mobile P_Rev • Only applicable for CDMA. – 0xFF - Not Available
bsPRevValid	• Indicates whether the base station P_Rev is valid – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
bsPRev	• Base station P_Rev. • Only applicable for CDMA. – 0xFF - Not Available
ccsSupported-Valid	• Indicates whether the supported concurrent service is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

ccsSupported	• Whether concurrent service is supported. • Only applicable for CDMA. – 0x00 - Not supported – 0x01 - Supported – 0xFF - Not Available
cdmaSysIdValid	• Indicates whether the CDMA system ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
systemID	• System ID. – 0xFFFF - Not Available
networkID	• Network ID. – 0xFFFF - Not Available
bsInfoValid	• Indicates whether the base station information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
baseLat	• Base station latitude in units of 0.25 sec. • Expressed as a two's complement signed number with positive numbers signifying North latitudes. – 0xFFFFFFFF - Not Available
baseLong	• Base station longitude in units of 0.25 sec. • Expressed as a two's complement signed number with positive numbers signifying East latitudes. – 0xFFFFFFFF - Not Available
packetZoneValid	• Indicates whether the packet zone is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
packetZone	• Packet zone (8-bit). – 0xFFFF indicates no packet zone. • Only applicable for CDMA.

networkIdValid	• Indicates whether the network ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
MCC[NAS_PLMN_LENGTH]	• Mobile Country Code. • MCC digits in ASCII characters • MCC wildcard value is returned as {'3', 0xFF, 0xFF}.
MNC[NAS_PLMN_LENGTH]	• Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • MNC wildcard value is returned as {'7', 0xFF, 0xFF}.

8.372.2 Field Documentation

8.372.2.1 uint16_t nas_CDMASysInfo::baseId

8.372.2.2 uint32_t nas_CDMASysInfo::baseLat

8.372.2.3 uint32_t nas_CDMASysInfo::baseLong

8.372.2.4 uint8_t nas_CDMASysInfo::bsInfoValid

8.372.2.5 uint8_t nas_CDMASysInfo::bsPRev

8.372.2.6 uint8_t nas_CDMASysInfo::bsPRevValid

8.372.2.7 uint8_t nas_CDMASysInfo::ccsSupported

8.372.2.8 uint8_t nas_CDMASysInfo::ccsSupportedValid

8.372.2.9 uint8_t nas_CDMASysInfo::cdmaSysIdValid

8.372.2.10 uint8_t nas_CDMASysInfo::isSysPrlMatch

8.372.2.11 uint8_t nas_CDMASysInfo::isSysPrlMatchValid

8.372.2.12 uint8_t nas_CDMASysInfo::MCC[3]

8.372.2.13 uint8_t nas_CDMASysInfo::MNC[3]

8.372.2.14 uint16_t nas_CDMASysInfo::networkID

8.372.2.15 uint8_t nas_CDMASysInfo::networkIdValid

8.372.2.16 uint16_t nas_CDMASysInfo::packetZone

8.372.2.17 uint8_t nas_CDMASysInfo::packetZoneValid

8.372.2.18 uint8_t nas_CDMASysInfo::pRevInUse

8.372.2.19 uint8_t nas_CDMASysInfo::pRevInUseValid

8.372.2.20 nas_sysInfoCommon nas_CDMASysInfo::sysInfoCDMA

8.372.2.21 uint16_t nas_CDMASysInfo::systemID

8.373 nas_CDMASysInfoExt Struct Reference

Data Fields

- uint16_t MCC
- uint8_t imsi_11_12

8.373.1 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

8.373.2 Field Documentation

8.373.2.1 uint8_t nas_CDMASysInfoExt::imsi_11_12

8.373.2.2 uint16_t nas_CDMASysInfoExt::MCC

8.374 nas_cellParams Struct Reference

Data Fields

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

8.374.1 Detailed Description

This structure contains information about the Cell parameters.

Parameters

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

8.374.2 Field Documentation

8.374.2.1 uint16_t nas_cellParams::pci

8.374.2.2 int16_t nas_cellParams::rsrp

8.374.2.3 int16_t nas_cellParams::rsrq

8.374.2.4 int16_t nas_cellParams::rssi

8.374.2.5 int16_t nas_cellParams::srxlev

8.375 nas_CommInfo Struct Reference

Data Fields

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

8.375.1 Detailed Description

Structure for storing the common information for the device.

Parameters

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

8.375.2 Field Documentation

8.375.2.1 uint8_t nas_CommInfo::imsRegState

8.375.2.2 uint8_t nas_CommInfo::modemMode

8.375.2.3 uint8_t nas_CommInfo::psState

8.375.2.4 uint8_t nas_CommInfo::systemMode

8.375.2.5 int8_t nas_CommInfo::temperature

8.376 nas_CSGID Struct Reference

Data Fields

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

8.376.1 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 • 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: – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE – 0x09 - RADIO_IF_TDSCDMA - TDS

8.376.2 Field Documentation

8.376.2.1 uint32_t nas_CSGID::id

8.376.2.2 uint16_t nas_CSGID::mcc

8.376.2.3 uint16_t nas_CSGID::mnc

8.376.2.4 uint8_t nas_CSGID::mncPcsDigits

8.376.2.5 uint8_t nas_CSGID::rat

8.377 nas_currentPLMN Struct Reference

Data Fields

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

8.377.1 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 – A 16 bit representation of MCC – Range 0 to 999
MNC	• mobile network code – A 16 bit representation of MNC – Range 0 to 999
netDescrLength	• Length of Network description field • Defaults to zero
netDescr	• Network Description – optional string containing network name or description

8.377.2 Field Documentation

8.377.2.1 uint16_t nas_currentPLMN::MCC

8.377.2.2 uint16_t nas_currentPLMN::MNC

8.377.2.3 uint8_t nas_currentPLMN::netDescr[255]

8.377.2.4 uint8_t nas_currentPLMN::netDescrLength

8.378 nas_dataSrvCapabilities Struct Reference

Data Fields

- uint8_t dataCapabilitiesLen
- uint8_t dataCapabilities [32]

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

8.378.2 Field Documentation

8.378.2.1 uint8_t nas_dataSrvCapabilities::dataCapabilities[32]

8.378.2.2 uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen

8.379 nas_DataStatusDetail Struct Reference

Data Fields

- uint32_t IPAddress
- uint8_t LastErrCode

8.379.1 Detailed Description

Parameters

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

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

8.379.2 Field Documentation

8.379.2.1 uint32_t nas_DataStatusDetail::IPAddress

8.379.2.2 uint8_t nas_DataStatusDetail::LastErrCode

8.380 nas_detailSvcInfo Struct Reference

Data Fields

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

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

isSysForbidden	• Forbidden system information • Values: – 0x00 - System is not a forbidden system – 0x01 - System is a forbidden system

8.380.2 Field Documentation

8.380.2.1 uint8_t nas_detailSvcInfo::hdrHybrid

8.380.2.2 uint8_t nas_detailSvcInfo::hdrSrvStatus

8.380.2.3 uint8_t nas_detailSvcInfo::isSysForbidden

8.380.2.4 uint8_t nas_detailSvcInfo::srvCapability

8.380.2.5 uint8_t nas_detailSvcInfo::srvStatus

8.381 nas_DeviceConfigDetail Struct Reference

Data Fields

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

8.381.1 Detailed Description

Parameters

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

HWVersion	• Hardware version – 0x00 - BSHWREV0 – 0x01 - BSHWREV1 – 0x02 - BSHWREV2 – 0x03 - BSHWREV3 – 0x04 - BSHWREVMAX – 0xFF - BSHWREVUNKNOWN

8.381.2 Field Documentation

8.381.2.1 uint8_t nas_DeviceConfigDetail::Chipset

8.381.2.2 uint8_t nas_DeviceConfigDetail::HWVersion

8.381.2.3 uint8_t nas_DeviceConfigDetail::QLIC

8.381.2.4 uint8_t nas_DeviceConfigDetail::Technology

8.382 nas_dirNum Struct Reference

Data Fields

- uint8_t dirNumLen
- uint8_t dirNum [255]

8.382.1 Detailed Description

This structure contains the parameters for Directory Number Information

Parameters

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

8.382.2 Field Documentation

8.382.2.1 uint8_t nas_dirNum::dirNum[255]

8.382.2.2 uint8_t nas_dirNum::dirNumLen

8.383 nas_DRCParams Struct Reference

Data Fields

- uint8_t DRCValue
- uint8_t DRCCover

8.383.1 Detailed Description

Parameters

DRCValue	• Current Data Rate Channel
DRCCover	• Current Data Rate Channel cover

8.383.2 Field Documentation

8.383.2.1 uint8_t nas_DRCParams::DRCCover

8.383.2.2 uint8_t nas_DRCParams::DRCValue

8.384 nas_ecioListElement Struct Reference

Data Fields

- int16_t ecio
- uint8_t radioIf

8.384.1 Detailed Description

This structure contains the ECIO Information

Parameters

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

8.384.2 Field Documentation

8.384.2.1 int16_t nas_ecioListElement::ecio

8.384.2.2 uint8_t nas_ecioListElement::radioIf

8.385 nas_ECIOThresh Struct Reference

Data Fields

- uint8_t ECIOThresListLen
- int16_t * pECIOThresList

8.385.1 Detailed Description

Parameters

ECIOThresList-Len	• Length of the ECIO threshold list parameter to follow
pECIOThresList	• 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 – Each ECIO threshold value is a signed 2 byte value – Each ECIO threshold value increments in negative 0.5 dB, e.g., an ECIO threshold value of 2 means -1dB – Maximum number of threshold values is 16 – At least one value must be specified – Threshold values specified above are used for all RATs

8.385.2 Field Documentation

8.385.2.1 uint8_t nas_ECIOThresh::ECIOThresListLen

8.385.2.2 int16_t* nas_ECIOThresh::pECIOThresList

8.386 nas_errorRateListElement Struct Reference

Data Fields

- uint16_t errorRate
- uint8_t radioIf

8.386.1 Detailed Description

This structure contains the Error Rate Information

Parameters

errorRate	• Error rate value corresponds to the RAT that is currently registered. – For CDMA, the error rate reported is Frame Error Rate: * Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% * A value of 0xFFFF indicates that the error rate is unknown or unavailable – For HDR, the error rate reported is Packet Error Rate: * Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% * A value of 0xFFFF indicates that the error rate is unknown or unavailable – For GSM, the error rate reported is Bit Error Rate: * Valid values are 0, 100, 200, 300, 400, 500, 600, and 700 The reported value divided by 100 gives the error rate as an RxQual value, e.g.,a value of 300 represents an RxQual value of 3. * A value of 25500 indicates No Data – For WCDMA, the error rate reported is Block Error Rate (BLER): * Valid values are 1 to 10000 * The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3%. * A value of 0 indicates No Data
radioIf	• Radio interface technology of the signal being measured – 0x00 – RADIO_IF_NO_SVC – None (no service) – 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X – 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000 HRPD (1xEV-DO) – 0x03 – RADIO_IF_AMPS – AMPS – 0x04 – RADIO_IF_GSM – GSM – 0x05 – RADIO_IF_UMTS – UMTS

8.386.2 Field Documentation

8.386.2.1 uint16_t nas_errorRateListElement::errorRate

8.386.2.2 uint8_t nas_errorRateListElement::radioIf

8.387 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]

8.387.1 Detailed Description

This structure contains information about the GERAN Network.

Parameters

cellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
plmn[NAS_PLMN_LENGTH]	• MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when nmrCellID is not present.
lac	• Location area code. • This field is ignored when nmrCellID is not present. – 0xFFFF - Not Available
arfcn	• Absolute RF channel number. – 0xFFFF - Not Available
bsic	• Base station identity code. – 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. – 0xFFFFFFFF - Not Available
rxLev	• Serving Cell Rx measurement. • Values range between 0 and 63. • Mapped to a measured signal level: – Rxlev 0 is a signal strength less than -110 dBm – Rxlev 1 is -110 dBm to -109 dBm – Rxlev 2 is -109 dBm to -108 dBm – ... – Rxlev 62 is -49 dBm to -48 dBm – Rxlev 63 is greater than -48 dBm – 0xFFFF - Not Available
nmrInst	• Provides the number of set of instances which follow. • If 0(zero), then no information follows it.
insNmrCellInfo[NAS_MAX_DESCRIPTION_LENGTH]	• See nas_nmrCellInfo for more information.

8.387.2 Field Documentation

8.387.2.1 uint16_t nas_GERANInfo::arfcn

8.387.2.2 uint8_t nas_GERANInfo::bsic

8.387.2.3 uint32_t nas_GERANInfo::cellID

8.387.2.4 nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]

8.387.2.5 uint16_t nas_GERANInfo::lac

8.387.2.6 uint8_t nas_GERANInfo::nmrInst

8.387.2.7 uint8_t nas_GERANInfo::plmn[3]

8.387.2.8 uint16_t nas_GERANInfo::rxLev

8.387.2.9 uint32_t nas_GERANInfo::timingAdvance

8.388 nas_geranInstInfo Struct Reference

Data Fields

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

8.388.1 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. • 0xFF indicates information is not present.
geranBsicBcc	• Base station identity code base station color code. • 0xFF indicates information is not present.
geranRssi	• Received signal strength indicator.

8.388.2 Field Documentation

8.388.2.1 uint16_t nas_geranInstInfo::geranArfcn

8.388.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

8.388.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

8.388.2.4 int16_t nas_geranInstInfo::geranRssi

8.389 nas_gsmCellInfo Struct Reference

Data Fields

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

8.389.1 Detailed Description

This structure contains information about the GSM Cell.

Parameters

arfcn	• GSM frequency being reported. • Range: 0 to 1023.
band1900	• Band indicator for the GSM ARFCN • This field is only valid if arfcn is in the overlapping region. • If TRUE and the cell is in the overlapping region, the ARFCN is on the 1900 band. • If FALSE, it is on the 1800 band.
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. • The lower 6 bits can be set to any value.
rssi	• Measured RSSI value in 1/10 dB. • Range: -200.0 dB to 0
srxlev	• Cell selection Rx level (Srxlev) value. • Range: -128 to 128. • This field is only valid when ue_in_idle is TRUE.

8.389.2 Field Documentation

8.389.2.1 uint16_t nas_gsmCellInfo::arfcn

8.389.2.2 uint8_t nas_gsmCellInfo::band1900

8.389.2.3 uint8_t nas_gsmCellInfo::bsicId

8.389.2.4 uint8_t nas_gsmCellInfo::cellIdValid

8.389.2.5 int16_t nas_gsmCellInfo::rssi

8.389.2.6 int16_t nas_gsmCellInfo::srxlev

8.390 nas_GSMRSSIThresh Struct Reference

Data Fields

- uint8_t GSMRSSIThreshListLen
- int16_t * pGSMRSSIThreshList

8.390.1 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) • Maximum of 32 values • Range for RSSI values: -111 to -48 (in dBm)

8.390.2 Field Documentation

8.390.2.1 uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen

8.390.2.2 int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList

8.391 nas_GSMSrvStatusInfo Struct Reference

Data Fields

- uint8_t srvStatus
- uint8_t trueSrvStatus
- uint8_t isPrefDataPath

8.391.1 Detailed Description

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

Parameters

srvStatus	• Service status of the system. – 0x00 - No service – 0x01 - Limited service – 0x02 - Service – 0x03 - Limited regional service – 0x04 - Power save – 0xFF - Not Available
trueSrvStatus	• True service status of the system. • Not applicable to CDMA/HDR. – 0x00 - No service – 0x01 - Limited service – 0x02 - Service – 0x03 - Limited regional service – 0x04 - Power save – 0xFF - Not Available
isPrefDataPath	• Whether the RAT is the preferred data path. – 0x00 - Not preferred – 0x01 - Preferred – 0xFF - Not Available

8.391.2 Field Documentation

8.391.2.1 uint8_t nas_GSMSrvStatusInfo::isPrefDataPath

8.391.2.2 uint8_t nas_GSMSrvStatusInfo::srvStatus

8.391.2.3 uint8_t nas_GSMSrvStatusInfo::trueSrvStatus

8.392 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

8.392.1 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.. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
lac	• Location area code. • Only applies to 3GPP. – 0xFFFF - Not Available
cellIdValid	• Indicates whether the cell ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
cellId	• Cell ID. – 0xFFFFFFFF - Not Available
regRejectInfo-Valid	• Indicates whether the registration reject information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
rejectSrvDomain	• Type of service domain in which the registration is rejected. – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
rejCause	• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. – 0xFF - Not Available

networkIdValid	• Indicates whether the network ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
MCC[NAS_PLMN_LENGTH]	• Mobile Country Code. • MCC digits in ASCII characters
MNC[NAS_PLMN_LENGTH]	• Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.
egprsSuppValid	• Indicates whether the EGPRS support is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
egprsSupp	• EGPRS support indication. • Only applicable for GSM. – 0x00 - Not available – 0x01 - Available – 0xFF - Not Available
dtmSuppValid	• Indicates whether Dual Transfer mode support is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
dtmSupp	• Dual Transfer mode support indication. • Only applicable for GSM. – 0x00 - Not available – 0x01 - Available – 0xFF - Not Available

8.392.2 Field Documentation

8.392.2.1 uint32_t nas_GSMSysInfo::cellId

8.392.2.2 uint8_t nas_GSMSysInfo::cellIdValid

8.392.2.3 uint8_t nas_GSMSysInfo::dtmSupp

8.392.2.4 uint8_t nas_GSMSysInfo::dtmSuppValid

8.392.2.5 uint8_t nas_GSMSysInfo::egprsSupp

8.392.2.6 uint8_t nas_GSMSysInfo::egprsSuppValid

8.392.2.7 uint16_t nas_GSMSysInfo::lac

8.392.2.8 uint8_t nas_GSMSysInfo::lacValid

8.392.2.9 uint8_t nas_GSMSysInfo::MCC[3]

8.392.2.10 uint8_t nas_GSMSysInfo::MNC[3]

8.392.2.11 uint8_t nas_GSMSysInfo::networkIdValid

8.392.2.12 uint8_t nas_GSMSysInfo::regRejectInfoValid

8.392.2.13 uint8_t nas_GSMSysInfo::rejCause

8.392.2.14 uint8_t nas_GSMSysInfo::rejectSrvDomain

8.392.2.15 nas_sysInfoCommon nas_GSMSysInfo::sysInfoGSM

8.393 nas_HDRECIOThresh Struct Reference

Data Fields

- uint8_t HDRECIOThreshListLen
- int16_t * pHDRECIOThreshList

8.393.1 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) • Maximum of 32 values • Range for ECIO values: -31.5 to 0 (in dB).

8.393.2 Field Documentation

8.393.2.1 uint8_t nas_HDRECIOThresh::HDRECIOThreshListLen

8.393.2.2 int16_t* nas_HDRECIOThresh::pHDRECIOThreshList

8.394 nas_HDRIOThresh Struct Reference

Data Fields

- uint8_t HDRIOThreshListLen
- int16_t * pHDRIOThreshList

8.394.1 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) • Maximum of 32 values • Range for IO values: -128 to -13 (in dBm).

8.394.2 Field Documentation

8.394.2.1 uint8_t nas_HDRIOThresh::HDRIOThreshListLen

8.394.2.2 int16_t* nas_HDRIOThresh::pHDRIOThreshList

8.395 nas_HDRRSSIThresh Struct Reference

Data Fields

- uint8_t HDRRSSIThreshListLen
- int16_t * pHDRRSSIThreshList

8.395.1 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) • Maximum of 32 values. • Range for RSSI values: -118 to -13 (in dBm).

8.395.2 Field Documentation

8.395.2.1 uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen

8.395.2.2 int16_t* nas_HDRRSSIThresh::pHDRRSSIThreshList

8.396 nas_HDRSINRThresh Struct Reference

Data Fields

- uint8_t HDRSINRThresListLen
- uint8_t * pHDRSINRThresList

8.396.1 Detailed Description

Parameters

HDRSINRThres-ListLen	• Length of the HDR SINR threshold list parameter to follow
pHDRSINR-ThresList	• 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 – SINR is reported only for HDR – Each SINR threshold value is an unsigned 1 byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.396.2 Field Documentation

8.396.2.1 uint8_t nas_HDRSINRThresh::HDRSINRThresListLen

8.396.2.2 uint8_t* nas_HDRSINRThresh::pHDRSINRThresList

8.397 nas_HDRSINRThreshold Struct Reference

Data Fields

- uint8_t HDRSINRThreshListLen
- uint16_t * pHDRSINRThreshList

8.397.1 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) • maximum of 32 values. • Valid levels are 0 to 8 – 0x00 - SINR_LEVEL_0 is -9 dB – 0x01 - SINR_LEVEL_1 is -6 dB – 0x02 - SINR_LEVEL_2 is -4.5 dB – 0x03 - SINR_LEVEL_3 is -3 dB – 0x04 - SINR_LEVEL_4 is -2 dB – 0x05 - SINR_LEVEL_5 is +1 dB – 0x06 - SINR_LEVEL_6 is +3 dB – 0x07 - SINR_LEVEL_7 is +6 dB – 0x08 - SINR_LEVEL_8 is +9 dB

8.397.2 Field Documentation

8.397.2.1 uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen

8.397.2.2 uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList

8.398 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]

8.398.1 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. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
isSysPrlMatch	• Indicates whether the system is in a PRL. • Only applies to CDMA/HDR. – 0x00 - System is not in a PRL – 0x01 - System is in a PRL – 0xFF - Not Available • If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL. • If the system is in a PRL, roam_status is set to the value based on the standard specification.
hdrPersonality-Valid	• Indicates whether the HDR personality is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
hdrPersonality	• HDR personality information. • Only applicable for HDR. – 0x00 - None – 0x02 - HRPD – 0x03 - eHRPD – 0xFF - Not Available
hdrActiveProt-Valid	• Indicates whether the HDR active protocol revision information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
hdrActiveProt	• HDR active protocol revision information . • Only applicable for HDR. – 0x00 - None – 0x02 - HDR Rel 0 – 0x03 - HDR Rel A – 0x04 - HDR Rel B – 0xFF - Not Available
is856SysIdValid	• Indicates whether the IS-856 system ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

is856SysId[SLQS_SYSTEM_ID_SIZE]	• IS-856 system ID. • Only applicable for HDR.

8.398.2 Field Documentation

8.398.2.1 uint8_t nas_HDRSysInfo::hdrActiveProt

8.398.2.2 uint8_t nas_HDRSysInfo::hdrActiveProtValid

8.398.2.3 uint8_t nas_HDRSysInfo::hdrPersonality

8.398.2.4 uint8_t nas_HDRSysInfo::hdrPersonalityValid

8.398.2.5 uint8_t nas_HDRSysInfo::is856SysId[16]

8.398.2.6 uint8_t nas_HDRSysInfo::is856SysIdValid

8.398.2.7 uint8_t nas_HDRSysInfo::isSysPrlMatch

8.398.2.8 uint8_t nas_HDRSysInfo::isSysPrlMatchValid

8.398.2.9 nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR

8.399 nas_homeSIDNID Struct Reference

Data Fields

- uint8_t numInstances
- nas_sidNid SidNid [255]

8.399.1 Detailed Description

Parameters

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

8.399.2 Field Documentation

8.399.2.1 uint8_t nas_homeSIDNID::numInstances

8.399.2.2 nas_sidNid nas_homeSIDNID::SidNid[255]

8.400 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]

8.400.1 Detailed Description

This structure contains information about the inter-frequency.

Parameters

earfcn	• E-UTRA absolute radio frequency channel number of the serving cell. • Range: 0 to 65535.
threshXLow	• Cell Srxlev low threshold. • Range: 0 to 31. • 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. • Range: 0 to 31. • When the serving cell exceeds thresh_serving_low, the value of an evaluated cell must be greater than this value to be considered for re-selection.
cell_resel_priority	• Cell re-selection priority • Range: 0 to 7. • 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.

8.400.2 Field Documentation

8.400.2.1 uint8_t nas_infoInterFreq::cell_resel_priority

8.400.2.2 nas_cellParams nas_infoInterFreq::cellInterFreqParams[255]

8.400.2.3 uint8_t nas_infoInterFreq::cells_len

8.400.2.4 uint16_t nas_infoInterFreq::earfcn

8.400.2.5 uint8_t nas_infoInterFreq::threshXHigh

8.400.2.6 uint8_t nas_infoInterFreq::threshXLow

8.401 nas_IOThresh Struct Reference

Data Fields

- uint8_t IOThresListLen
- int32_t * pIOThresList

8.401.1 Detailed Description

Parameters

IOThresListLen	• Length of the LTE SNR threshold list parameter to follow
pIOThresList	• Sequence of thresholds delimiting IO event reporting bands • 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 – IO is applicable only for HDR – Each IO threshold value is a signed 4 byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.401.2 Field Documentation

8.401.2.1 uint8_t nas_IOThresh::IOThresListLen

8.401.2.2 int32_t* nas_IOThresh::pIOThresList

8.402 nas_lteEARFCN Struct Reference

Data Fields

- uint8_t status
- uint32_t earfcn0
- uint32_t earfcn1

8.402.1 Detailed Description

Parameters

status	• 0 - Disable • 1 - Enable

earfcn0	• Primary DL EARFCN to which the UE is locked
earfcn1	• Secondary DL EARFCN to which the UE is locked • Note : Make earfcn1 value equal to earfcn0 if only one EARFCN is desired.

8.402.2 Field Documentation

8.402.2.1 uint32_t nas_lteEARFCN::earfcn0

8.402.2.2 uint32_t nas_lteEARFCN::earfcn1

8.402.2.3 uint8_t nas_lteEARFCN::status

8.403 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]

8.403.1 Detailed Description

This structure contains information about the LTE GSM Cell.

Parameters

cellReselPriority	• Priority of this frequency group. • Range: 0 to 7. • This field is only valid when ue_in_idle is TRUE.
threshGsmHigh	• Reselection threshold for high priority layers. • Range: 0 to 31. • This field is only valid when ue_in_idle is TRUE.
threshGsmLow	• Reselection threshold for low priority layers. • Range: 0 to 31. • 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. • Range: 0 to 255. • 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. • 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.

8.403.2 Field Documentation

8.403.2.1 uint8_t nas_lteGsmCellInfo::cellReselPriority

8.403.2.2 uint8_t nas_lteGsmCellInfo::cells_len

8.403.2.3 nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]

8.403.2.4 uint8_t nas_lteGsmCellInfo::nccPermitted

8.403.2.5 uint8_t nas_lteGsmCellInfo::threshGsmHigh

8.403.2.6 uint8_t nas_lteGsmCellInfo::threshGsmLow

8.404 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

8.404.1 Detailed Description

Structure for storing the LTE information for the device.

Parameters

band	• LTE Band – 1 ∼ 40 (Band in decimal) – 0xFF - Invalid

bandwidth	• BandWidth. – 0x00 - 1.4 MHz – 0x01 - 3 MHz – 0x02 - 5 MHz – 0x03 - 10 MHz – 0x04 - 15 MHz – 0x05 - 20 MHz – 0x06 - Invalid – 0xFF - Unknown
RXChan	• RX channel number in decimal – 0xFFFF - Not Available
TXChan	• TX channel number in decimal – 0xFFFF - Not Available
emmState	• EMM State. – 0x00 - Deregistered – 0x01 - Reg Initiated – 0x02 - Registered – 0x03 - TAU Initiated – 0x04 - SR Initiated – 0x05 - Dereg Initiated – 0x06 - Invalid – 0xFF - Unknown

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

8.404.2 Field Documentation

8.404.2.1 uint8_t nas_LTEInfo::band

8.404.2.2 uint8_t nas_LTEInfo::bandwidth

8.404.2.3 uint8_t nas_LTEInfo::emmConnState

8.404.2.4 uint8_t nas_LTEInfo::emmState

8.404.2.5 uint8_t nas_LTEInfo::emmSubState

8.404.2.6 uint16_t nas_LTEInfo::RXChan

8.404.2.7 uint16_t nas_LTEInfo::TXChan

8.405 nas_LTEInfoInterfreq Struct Reference

Data Fields

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

8.405.1 Detailed Description

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

Parameters

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

8.405.2 Field Documentation

8.405.2.1 uint8_t nas_LTEInfoInterfreq::freqsLen

8.405.2.2 nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq[255]

8.405.2.3 uint8_t nas_LTEInfoInterfreq::ueInIdle

8.406 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]

8.406.1 Detailed Description

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

Parameters

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

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

8.406.2 Field Documentation

8.406.2.1 nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]

8.406.2.2 uint8_t nas_LTEInfoIntrafreq::cellReselPriority

8.406.2.3 uint8_t nas_LTEInfoIntrafreq::cellsLen

8.406.2.4 uint16_t nas_LTEInfoIntrafreq::earfcn

8.406.2.5 uint32_t nas_LTEInfoIntrafreq::globalCellId

8.406.2.6 uint8_t nas_LTEInfoIntrafreq::plmn[3]

8.406.2.7 uint16_t nas_LTEInfoIntrafreq::servingCellId

8.406.2.8 uint8_t nas_LTEInfoIntrafreq::sIntraSearch

8.406.2.9 uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch

8.406.2.10 uint16_t nas_LTEInfoIntrafreq::tac

8.406.2.11 uint8_t nas_LTEInfoIntrafreq::threshServingLow

8.406.2.12 uint8_t nas_LTEInfoIntrafreq::ueInIdle

8.407 nas_LTEInfoNeighboringGSM Struct Reference

Data Fields

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

8.407.1 Detailed Description

This structure contains information about the LTE Neighboring GSM Network.

Parameters

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

8.407.2 Field Documentation

8.407.2.1 uint8_t nas_LTEInfoNeighboringGSM::freqsLen

8.407.2.2 nas_lteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.407.2.3 uint8_t nas_LTEInfoNeighboringGSM::ueInIdle

8.408 nas_LTEInfoNeighboringWCDMA Struct Reference

Data Fields

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

8.408.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA Network.

Parameters

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

8.408.2 Field Documentation

8.408.2.1 uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen

8.408.2.2 nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.408.2.3 uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle

8.409 nas_ltePCI Struct Reference

Data Fields

- uint8_t status
- uint32_t earfcn
- uint32_t pci

8.409.1 Detailed Description

Parameters

status	• 0 - Disable • 1 - Enable
earfcn	• UARFCN to which UE is locked
pci	• PCI to which the UE is locked

8.409.2 Field Documentation

8.409.2.1 uint32_t nas_ltePCI::earfcn

8.409.2.2 uint32_t nas_ltePCI::pci

8.409.2.3 uint8_t nas_ltePCI::status

8.410 nas_lteRsrpinformation Struct Reference

Data Fields

- int16_t rsrplevel

8.410.1 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).

8.410.2 Field Documentation

8.410.2.1 int16_t nas_lteRsrpinformation::rsrplevel

8.411 nas_LTERSRPThresh Struct Reference

Data Fields

- uint8_t LTERSRPThreshListLen
- int16_t * pLTERSRPThreshList

8.411.1 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) • Maximum of 32 values • Range for RSRP values: -140 to -44 (in dBm).

8.411.2 Field Documentation

8.411.2.1 uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen

8.411.2.2 int16_t* nas_LTERSRPThresh::pLTERSRPThreshList

8.412 nas_LTERSRQThresh Struct Reference

Data Fields

- uint8_t LTERSRQThreshListLen
- int16_t * pLTERSRQThreshList

8.412.1 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) • Maximum of 32 values. • Range for RSRQ values: -20 to -3 (in dBm)

8.412.2 Field Documentation

8.412.2.1 uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen

8.412.2.2 int16_t* nas_LTERSRQThresh::pLTERSRQThreshList

8.413 nas_LTERSSIThresh Struct Reference

Data Fields

- uint8_t LTERSSIThreshListLen
- int16_t * pLTERSSIThreshList

8.413.1 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) • Maximum of 32 values. • Range for RSSI values: -120 to 0 (in dBm)

8.413.2 Field Documentation

8.413.2.1 uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen

8.413.2.2 int16_t* nas_LTERSSIThresh::pLTERSSIThreshList

8.414 nas_LTESigRptCfg Struct Reference

Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

8.414.1 Detailed Description

Parameters

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

8.414.2 Field Documentation

8.414.2.1 uint8_t nas_LTESigRptCfg::avgPeriod

8.414.2.2 uint8_t nas_LTESigRptCfg::rptRate

8.415 nas_LTESigRptConfig Struct Reference

Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

8.415.1 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 • 0 - Report using the default configuration • 1 - Report every 1 sec • 2 - Report every 2 sec • 3 - Report every 3 sec • 4 - Report every 4 sec • 5 - Report every 5 sec
avgPeriod	• Averaging period to be used for the LTE signal. • Values – 0 - Average using the default configuration – 1 - Average over 1 sec – 2 - Average over 2 sec – 3 - Average over 3 sec – 4 - Average over 4 sec – 5 - Average over 5 sec – 6 - Average over 6 sec – 7 - Average over 7 sec – 8 - Average over 8 sec – 9 - Average over 9 sec – 10 - Average over 10 sec

8.415.2 Field Documentation

8.415.2.1 uint8_t nas_LTESigRptConfig::avgPeriod

8.415.2.2 uint8_t nas_LTESigRptConfig::rptRate

8.416 nas_lteSnrinformation Struct Reference

Data Fields

- int16_t snrlevel

8.416.1 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.

8.416.2 Field Documentation

8.416.2.1 int16_t nas_lteSnrinformation::snrlevel

8.417 nas_LTESNRThresh Struct Reference

Data Fields

- uint8_t LTESNRThresListLen
- int16_t * pLTESNRThresList

8.417.1 Detailed Description

Parameters

LTESNRThres-ListLen	• Length of the LTE SNR threshold list parameter to follow
pLTESNRThres-List	• Sequence of thresholds delimiting SNR event reporting bands • 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 – For LTE, each SNR threshold value is a signed 2 Byte value – Maximum number of threshold values is 16 – At least one value must be specified – 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

8.417.2 Field Documentation

8.417.2.1 uint8_t nas_LTESNRThresh::LTESNRThresListLen

8.417.2.2 int16_t* nas_LTESNRThresh::pLTESNRThresList

8.418 nas_LTESNRThreshold Struct Reference

Data Fields

- uint8_t LTESNRThreshListLen
- int16_t * pLTESNRThreshList

8.418.1 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) • Maximum of 32 values • Range for SNR values: -20 to 30 (in dB).

8.418.2 Field Documentation

8.418.2.1 uint8_t nas_LTESNRThreshold::LTESNRThreshListLen

8.418.2.2 int16_t* nas_LTESNRThreshold::pLTESNRThreshList

8.419 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

8.419.1 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.. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
lac	• Location area code. • Only applies to 3GPP. – 0xFFFF - Not Available

cellIdValid	• Indicates whether the cell ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
cellId	• Cell ID. – 0xFFFFFFFF - Not Available
regRejectInfo-Valid	• Indicates whether the registration reject information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
rejectSrvDomain	• Type of service domain in which the registration is rejected. – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
rejCause	• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. – 0xFF - Not Available
networkIdValid	• Indicates whether the network ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
MCC[NAS_PLM-N_LENGTH]	• Mobile Country Code. • MCC digits in ASCII characters
MNC[NAS_PLM-N_LENGTH]	• Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.

tacValid	• Indicates whether tracking area code is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
tac	• Tracking area code. • Only applicable for LTE. – 0xFFFF - Not Available

8.419.2 Field Documentation

8.419.2.1 uint32_t nas_LTESysInfo::cellId

8.419.2.2 uint8_t nas_LTESysInfo::cellIdValid

8.419.2.3 uint16_t nas_LTESysInfo::lac

8.419.2.4 uint8_t nas_LTESysInfo::lacValid

8.419.2.5 uint8_t nas_LTESysInfo::MCC[3]

8.419.2.6 uint8_t nas_LTESysInfo::MNC[3]

8.419.2.7 uint8_t nas_LTESysInfo::networkIdValid

8.419.2.8 uint8_t nas_LTESysInfo::regRejectInfoValid

8.419.2.9 uint8_t nas_LTESysInfo::rejCause

8.419.2.10 uint8_t nas_LTESysInfo::rejectSrvDomain

8.419.2.11 nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE

8.419.2.12 uint16_t nas_LTESysInfo::tac

8.419.2.13 uint8_t nas_LTESysInfo::tacValid

8.420 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]

8.420.1 Detailed Description

This structure contains information about the LTE WCDMA Cell.

Parameters

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

8.420.2 Field Documentation

8.420.2.1 uint8_t nas_lteWcdmaCellInfo::cellReselPriority

8.420.2.2 uint8_t nas_lteWcdmaCellInfo::cellsLen

8.420.2.3 uint16_t nas_lteWcdmaCellInfo::threshXhigh

8.420.2.4 uint16_t nas_lteWcdmaCellInfo::threshXlow

8.420.2.5 uint16_t nas_lteWcdmaCellInfo::uarfcn

8.420.2.6 nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]

8.421 nas_minBasedIMSI Struct Reference

Data Fields

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

- uint8_t imsiMS2 [3]

8.421.1 Detailed Description

Parameters

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

8.421.2 Field Documentation

8.421.2.1 uint16_t nas_minBasedIMSI::imsiM1112

8.421.2.2 uint8_t nas_minBasedIMSI::imsiMS1[7]

8.421.2.3 uint8_t nas_minBasedIMSI::imsiMS2[3]

8.421.2.4 uint8_t nas_minBasedIMSI::mccM[3]

8.422 nas_MNRInfo Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- uint32_t rat

8.422.1 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. – 0x04 - RADIO_IF_GSM – 0x05 - RADIO_IF_UMTS – 0x08 - RADIO_IF_LTE

8.422.2 Field Documentation

8.422.2.1 uint16_t nas_MNRInfo::mcc

8.422.2.2 uint16_t nas_MNRInfo::mnc

8.422.2.3 uint32_t nas_MNRInfo::rat

8.423 nas_namName Struct Reference

Data Fields

- uint8_t namNameLen
- uint8_t namName [12]

8.423.1 Detailed Description

Parameters

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

8.423.2 Field Documentation

8.423.2.1 uint8_t nas_namName::namName[12]

8.423.2.2 uint8_t nas_namName::namNameLen

8.424 nas_netSelectionPref Struct Reference

Data Fields

- uint8_t netReg
- uint16_t mcc
- uint16_t mnc

8.424.1 Detailed Description

Contain the network selection preference.

Parameters

netReg	• specifies one of the following actions: – 0x00 - Automatic registration * Device registers according to its provisioning; mcc and mnc fields are ignored – 0x01 - Manual Registration * 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

8.424.2 Field Documentation

8.424.2.1 uint16_t nas_netSelectionPref::mcc

8.424.2.2 uint16_t nas_netSelectionPref::mnc

8.424.2.3 uint8_t nas_netSelectionPref::netReg

8.425 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

8.425.1 Detailed Description

Parameters

State	• CDMA current access state – 0x01 - Searching – 0x02 - Idle – 0x03 - Traffic – Others - NA

SO	• CDMA service option – 0xFFFF - Not in a call – 0x0001 - Basic Variable Rate Voice Service(8kbps) – 0x0002 - Mobile Station Loopback(8kbps) – 0x0003 - Enhanced Variable Rate Codec(EVRC) Voice Service(8kbps) – 0x0006 - Short message Services(Rate Set 1) – 0x0009 - Mobile Station Loopback(13kbps) – 0x000E - Short Message Service (Rate Set 2) – 0x0011 - High Rate Voice Service(13kbps) – 0x0020 - Test Data Service Option(TDSO) – 0x0021 - cdma2000 High Speed Packet Data Service, Internet or ISO Protocol Stack – 0x0044 - EVRC-B Voice Service(8 kbps) – 0x0046 - EVRC-WB Voice Service(8 kbps) – 0x0049 - Voice Echo mode supports smart blanking(EVRC-NW) – 0x004B - Enhanced loopback – 0x8000 - Proprietary Service Option (Qualcomm Inc.)
RX_PWR	• RX Pwr(dBm) – 0xABCD00EF - -ABCD.EF dBm – ABCD00EF should be transferred to decimal while displaying – Example: 0x12340056 - -4660.86dBm 0x1234 = 4660, 0x0056 = 86 – 0xFFFFFFFF - NA
RX_EC_IO	• RX EC/IO(dB) – 0xABCD - -AB.CD dB – ABCD should be transferred to decimal while displaying – Example: 0x1234 - -18.52dB 0x12 = 18, 0x34 = 52 – 0xFFFF - NA
TX_PWR	• TX PWR(dBm) – 0xFFFFFFFF - NA – Others - display actual value in decimal – Example: 0x1234 - -4660dBm 0x1234 = 4660
ActSetCnt	• Count of active pilot PN elements • As input specifies number of sets of parameter pActPilotElements for which memory has been assigned • 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 • As input specifies number of sets of parameter pNeighborSetPilotPN for which memory has been assigned • As output specifies the actual number of sets of parameter pNeighborSetPilotPN returned by device
pNeighborSet-PilotPN	• Neighbor pilot PN

8.425.2 Field Documentation

8.425.2.1 uint8_t nas_NetworkStat1x::ActSetCnt

8.425.2.2 uint8_t nas_NetworkStat1x::NeighborSetCnt

8.425.2.3 nas_ActPilotPNElement* nas_NetworkStat1x::pActPilotPNElements

8.425.2.4 uint16_t* nas_NetworkStat1x::pNeighborSetPilotPN

8.425.2.5 uint16_t nas_NetworkStat1x::RX_EC_IO

8.425.2.6 uint32_t nas_NetworkStat1x::RX_PWR

8.425.2.7 uint16_t nas_NetworkStat1x::SO

8.425.2.8 uint8_t nas_NetworkStat1x::State

8.425.2.9 uint32_t nas_NetworkStat1x::TX_PWR

8.426 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

8.426.1 Detailed Description

Parameters

State	• EVDO network access state – 0x00 - Sleep – 0x01 - Searching – 0x02 - Idle – 0x03 - Active – 0xFF - NA
MACIndex	• HDR Mac index – 0xFF - NA – Others - Display the actual value in decimal – Example: 0x12 - 18 0x12 = 18
SectorIDLen	(IN/OUT) • Sector ID length • As input specifies length of parameter pSectorID for which memory has been assigned • As output specifies the actual length of parameter pSectorID returned by device
pSectorID	• Sector ID
RX_PWR	• TX PWR(dBm) – 0xABCD - -ABCD dBm – ABCD should be transferred to decimal while displaying – Example: 0x1234 - -4660dBm 0x1234 = 4660 – 0xFFFF - NA
PER	• HDR Packet Error Rate – 0xFFFF - Unknown – Others - display the actual value in decimal – Example: 0x1234 - -4660dBm 0x1234 = 4660
PilotEnergy	• Pilt Energy (dB) – 0xFFFF - NA – 0xABCD should be transferred to decimal while displaying – Example: 0x1234 - -4660dBm 0x1234 = 4660
SNR	• Signal to Noise ratio (dB)

8.426.2 Field Documentation

8.426.2.1 uint8_t nas_NetworkStatEVDO::MACIndex

8.426.2.2 uint16_t nas_NetworkStatEVDO::PER

8.426.2.3 uint16_t nas_NetworkStatEVDO::PilotEnergy

8.426.2.4 uint16_t* nas_NetworkStatEVDO::pSectorID

8.426.2.5 uint16_t nas_NetworkStatEVDO::RX_PWR

8.426.2.6 uint8_t nas_NetworkStatEVDO::SectorIDLen

8.426.2.7 uint8_t nas_NetworkStatEVDO::SNR

8.426.2.8 uint8_t nas_NetworkStatEVDO::State

8.427 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

8.427.1 Detailed Description

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

Parameters

nmrCellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
nmrPlmn[NAS_PLMN_LENGTH]	• MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when nmrCellID is not present.
nmrLac	• Location area code. • This field is ignored when nmrCellID is not present. – 0xFFFF - Not Available
nmrArfcn	• Absolute RF channel number. – 0xFFFF - Not Available
nmrBsic	• Base station identity code. – 0xFF - Not Available

nmrRxLev	• Cell Rx measurement. • Values range between 0 and 63. • Mapped to a measured signal level: – Rxlev 0 is a signal strength less than -110 dBm – Rxlev 1 is -110 dBm to -109 dBm – Rxlev 2 is -109 dBm to -108 dBm – ... – Rxlev 62 is -49 dBm to -48 dBm – Rxlev 63 is greater than -48 dBm – 0xFFFF - Not Available

8.427.2 Field Documentation

8.427.2.1 uint16_t nas_nmrCellInfo::nmrArfcn

8.427.2.2 uint8_t nas_nmrCellInfo::nmrBsic

8.427.2.3 uint32_t nas_nmrCellInfo::nmrCellID

8.427.2.4 uint16_t nas_nmrCellInfo::nmrLac

8.427.2.5 uint8_t nas_nmrCellInfo::nmrPlmn[3]

8.427.2.6 uint16_t nas_nmrCellInfo::nmrRxLev

8.428 nas_operatorNameString Struct Reference

Data Fields

- uint8_t PLMNName [255]

8.428.1 Detailed Description

Parameters

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

8.428.2 Field Documentation

8.428.2.1 uint8_t nas_operatorNameString::PLMNName[255]

8.429 nas_OperatorPLMNData Struct Reference

Data Fields

- uint8_t mcc [3]
- uint8_t mnc [3]

- uint16_t lac1
- uint16_t lac2
- uint8_t PLMNRecID

8.429.1 Detailed Description

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.

8.429.2 Field Documentation

8.429.2.1 uint16_t nas_OperatorPLMNData::lac1

8.429.2.2 uint16_t nas_OperatorPLMNData::lac2

8.429.2.3 uint8_t nas_OperatorPLMNData::mcc[3]

8.429.2.4 uint8_t nas_OperatorPLMNData::mnc[3]

8.429.2.5 uint8_t nas_OperatorPLMNData::PLMNRecID

8.430 nas_operatorPLMNList Struct Reference

Data Fields

- uint16_t numInstance
- nas_OperatorPLMNData PLMNData [255]

8.430.1 Detailed Description

Parameters

numInstance	• Number of sets of the elements.

PLMNData	• Refer OperatorPLMNLData for details (Optional).

8.430.2 Field Documentation

8.430.2.1 uint16_t nas_operatorPLMNList::numInstance

8.430.2.2 nas_OperatorPLMNData nas_operatorPLMNList::PLMNData[255]

8.431 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

8.431.1 Detailed Description

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

Parameters

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

8.431.2 Field Documentation

8.431.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggPcellInfo::dl_bw_value

8.431.2.2 uint16_t nas_PhyCaAggPcellInfo::freq

8.431.2.3 uint16_t nas_PhyCaAggPcellInfo::iLTEbandValue

8.431.2.4 uint16_t nas_PhyCaAggPcellInfo::pci

8.431.2.5 uint8_t nas_PhyCaAggPcellInfo::TlvPresent

8.432 nas_PhyCaAggScellDlBw Struct Reference

Data Fields

- NAS_LTE_CPHY_CA_BW_NRB_LITE dl_bw_value
- uint8_t TlvPresent

8.432.1 Detailed Description

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

Parameters

dl_bw_value	• Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information.

8.432.2 Field Documentation

8.432.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellDlBw::dl_bw_value

8.432.2.2 uint8_t nas_PhyCaAggScellDlBw::TlvPresent

8.433 nas_PhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

8.433.1 Detailed Description

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

Parameters

scell_idx	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
TlvPresent	• Tlv Present.

8.433.2 Field Documentation

8.433.2.1 uint8_t nas_PhyCaAggScellIndex::scell_idx

8.433.2.2 uint8_t nas_PhyCaAggScellIndex::TlvPresent

8.434 nas_PhyCaAggScellIndType Struct Reference

Data Fields

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

8.434.1 Detailed Description

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

Parameters

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

8.434.2 Field Documentation

8.434.2.1 uint16_t nas_PhyCaAggScellIndType::freq

8.434.2.2 uint16_t nas_PhyCaAggScellIndType::pci

8.434.2.3 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellIndType::scell_state

8.434.2.4 uint8_t nas_PhyCaAggScellIndType::TlvPresent

8.435 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

8.435.1 Detailed Description

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

Parameters

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

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

TlvPresent	• Tlv Present.

8.435.2 Field Documentation

8.435.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellInfo::dl_bw_value

8.435.2.2 uint16_t nas_PhyCaAggScellInfo::freq

8.435.2.3 uint16_t nas_PhyCaAggScellInfo::iLTEbandValue

8.435.2.4 uint16_t nas_PhyCaAggScellInfo::pci

8.435.2.5 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellInfo::scell_state

8.435.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

8.436 nas_PilotSetData Struct Reference

Data Fields

- uint8_t NumPilots
- nas_PilotSetParams * pPilotSetInfo

8.436.1 Detailed Description

Parameters

NumPilots	• Number of Pilot Sets • As input specifies number of sets of parameter pPilotSetInfo for which memory has been assigned • As output specifies the actual number of sets of parameter pPilotSetInfo returned by device
pPilotSetInfo	• Pilot Set Parameters • 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.

8.436.2 Field Documentation

8.436.2.1 uint8_t nas_PilotSetData::NumPilots

8.436.2.2 nas_PilotSetParams* nas_PilotSetData::pPilotSetInfo

8.437 nas_PilotSetParams Struct Reference

Data Fields

- uint32_t PilotType
- uint16_t PilotPN
- uint16_t PilotStrength

8.437.1 Detailed Description

Parameters

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

8.437.2 Field Documentation

8.437.2.1 uint16_t nas_PilotSetParams::PilotPN

8.437.2.2 uint16_t nas_PilotSetParams::PilotStrength

8.437.2.3 uint32_t nas_PilotSetParams::PilotType

8.438 nas_PLMNNetworkName Struct Reference

Data Fields

- uint8_t numInstance
- nas_PLMNNetworkNameData PLMNNetName [255]

8.438.1 Detailed Description

Parameters

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

8.438.2 Field Documentation

8.438.2.1 uint8_t nas_PLMNNetworkName::numInstance

8.438.2.2 nas_PLMNNetworkNameData nas_PLMNNetworkName::PLMNNetName[255]

8.439 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]

8.439.1 Detailed Description

Parameters

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

8.439.2 Field Documentation

8.439.2.1 uint8_t nas_PLMNNetworkNameData::codingScheme

8.439.2.2 uint8_t nas_PLMNNetworkNameData::countryInitials

8.439.2.3 uint8_t nas_PLMNNetworkNameData::longName[255]

8.439.2.4 uint8_t nas_PLMNNetworkNameData::longNameLen

8.439.2.5 uint8_t nas_PLMNNetworkNameData::longNameSpareBits

8.439.2.6 uint8_t nas_PLMNNetworkNameData::shortName[255]

8.439.2.7 uint8_t nas_PLMNNetworkNameData::shortNameLen

8.439.2.8 uint8_t nas_PLMNNetworkNameData::shortNameSpareBits

8.440 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

8.440.1 Detailed Description

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

8.440.2 Field Documentation

8.440.2.1 uint16_t nas_protocolSubtypeElement::AccessMac

8.440.2.2 uint16_t nas_protocolSubtypeElement::AuthProt

8.440.2.3 uint16_t nas_protocolSubtypeElement::ControlMac

8.440.2.4 uint16_t nas_protocolSubtypeElement::EncryptProt

8.440.2.5 uint16_t nas_protocolSubtypeElement::ForwardMac

8.440.2.6 uint16_t nas_protocolSubtypeElement::IdleState

8.440.2.7 uint16_t nas_protocolSubtypeElement::KeyExchange

8.440.2.8 uint16_t nas_protocolSubtypeElement::MultDisc

8.440.2.9 uint16_t nas_protocolSubtypeElement::PhysicalLayer

8.440.2.10 uint16_t nas_protocolSubtypeElement::ReverseMac

8.440.2.11 uint16_t nas_protocolSubtypeElement::SecProt

8.440.2.12 uint16_t nas_protocolSubtypeElement::VirtStream

8.441 nas_qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- uint8_t leapSeconds
- uint8_t localTimeOffset
- uint8_t daylightSavings

8.441.1 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 – 0x00 - OFF (daylight savings not in effect) – 0x01 - ON (daylight savings in effect)

8.441.2 Field Documentation

8.441.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

8.441.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

8.441.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

8.442 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]

8.442.1 Detailed Description

Parameters

MCC	Mobile Country Code
MNC	Mobile Network Code
InUse	current network or not
Roaming	Home/Roam Status of the Network
Forbidden	Network in the forbidden network list or not
Preferred	Network in the Preferred network list or not
Desription	Network Name/Description

8.442.2 Field Documentation

8.442.2.1 char nas_QmiNas3GppNetworkInfo::Desription[255]

8.442.2.2 uint32_t nas_QmiNas3GppNetworkInfo::Forbidden

8.442.2.3 uint32_t nas_QmiNas3GppNetworkInfo::InUse

8.442.2.4 uint16_t nas_QmiNas3GppNetworkInfo::MCC

8.442.2.5 uint16_t nas_QmiNas3GppNetworkInfo::MNC

8.442.2.6 uint32_t nas_QmiNas3GppNetworkInfo::Preferred

8.442.2.7 uint32_t nas_QmiNas3GppNetworkInfo::Roaming

8.443 nas_QmiNas3GppNetworkRAT Struct Reference

Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t RAT

8.443.1 Detailed Description

Contain the 3GPP radio access technology information.

Parameters

MCC	• Mobile Country Code
MNC	• Mobile Network Code

RAT	• Radio Access Technology – 0x04 - GERAN – 0x05 - UMTS – 0x08 - LTE – 0x09 - TD-SCDMA

8.443.2 Field Documentation

8.443.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

8.443.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

8.443.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

8.444 nas_QmisNasPcsDigit Struct Reference

Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t includes_pcs_digit

8.444.1 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 • 0x01 - MNC is a three-digit value • 0x00 - MNC is a two-digit value

8.444.2 Field Documentation

8.444.2.1 uint8_t nas_QmisNasPcsDigit::includes_pcs_digit

8.444.2.2 uint16_t nas_QmisNasPcsDigit::MCC

8.444.2.3 uint16_t nas_QmisNasPcsDigit::MNC

8.445 nas_RejectReasonTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t serviceDomain
- uint32_t rejectCause

8.445.1 Detailed Description

Parameters

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

8.445.2 Field Documentation

8.445.2.1 uint32_t nas_RejectReasonTlv::rejectCause

8.445.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

8.445.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

8.446 nas_RFInfoTlv Struct Reference

Data Fields

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

8.446.1 Detailed Description

Parameters

TlvPresent	indicating the presence of the TLV in the QMI ind
radioInterface	radio interface technology of the signal being measured
activeBandClass	active band class
activeChannel	active channel

8.446.2 Field Documentation

8.446.2.1 uint32_t nas_RFInfoTlv::activeBandClass[255]

8.446.2.2 uint32_t nas_RFInfoTlv::activeChannel[255]

8.446.2.3 uint32_t nas_RFInfoTlv::radioInterface[255]

8.446.2.4 uint8_t nas_RFInfoTlv::radioInterfaceSize

8.446.2.5 uint8_t nas_RFInfoTlv::TlvPresent

8.447 nas_roamIndList Struct Reference

Data Fields

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

8.447.1 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 – defaults to zero
radioInterface	• Radio Interface currently in use • Values: – 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE
roamIndicator	• Roaming Indicator • Values: – 0x00 - Roaming – 0x01 - Home

8.447.2 Field Documentation

8.447.2.1 uint8_t nas_roamIndList::numInstances

8.447.2.2 uint8_t nas_roamIndList::radioInterface[32]

8.447.2.3 uint8_t nas_roamIndList::roamIndicator[32]

8.448 nas_RSRPThresh Struct Reference

Data Fields

- uint8_t RSRPThresListLen
- int16_t * pRSRPThresList

8.448.1 Detailed Description

Parameters

RSRPThresList-Len	• Length of the LTE RSRP threshold list parameter to follow
pRSRPThresList	• Sequence of thresholds delimiting current RSRP event reporting bands • 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 – RSRP values are applicable only for LTE – RSRP values are measured in dBm, with a range of -44 dBm to -140 dBm – Each RSRP threshold value is a signed byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.448.2 Field Documentation

8.448.2.1 int16_t* nas_RSRPThresh::pRSRPThresList

8.448.2.2 uint8_t nas_RSRPThresh::RSRPThresListLen

8.449 nas_rsrqInformation Struct Reference

Data Fields

- int8_t rsrq
- uint8_t radioIf

8.449.1 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 – 0x08 – LTE

8.449.2 Field Documentation

8.449.2.1 uint8_t nas_rsrqInformation::radioIf

8.449.2.2 int8_t nas_rsrqInformation::rsrq

8.450 nas_RSRQThresh Struct Reference

Data Fields

- uint8_t RSRQThresListLen
- int8_t * pRSRQThresList

8.450.1 Detailed Description

Parameters

RSRQThresList-Len	• Length of the LTE RSRQ threshold list parameter to follow
pRSRQThresList	• Sequence of thresholds delimiting current RSRQ event reporting bands • 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 – RSRQ values are applicable only for LTE – RSRQ values are measured in dBm, with a range of -20 dBm to -3 dBm – Each RSRQ threshold value is a signed byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.450.2 Field Documentation

8.450.2.1 int8_t* nas_RSRQThresh::pRSRQThresList

8.450.2.2 uint8_t nas_RSRQThresh::RSRQThresListLen

8.451 nas_RSSIThresh Struct Reference

Data Fields

- uint8_t RSSIThresListLen
- int8_t * pRSSIThresList

8.451.1 Detailed Description

Parameters

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

8.451.2 Field Documentation

8.451.2.1 int8_t* nas_RSSIThresh::pRSSIThresList

8.451.2.2 uint8_t nas_RSSIThresh::RSSIThresListLen

8.452 nas_rxInfo Struct Reference

Data Fields

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

8.452.1 Detailed Description

Parameters

isRadioTuned	• Whether Rx is tuned to a channel: – 0x00 - Not tuned – 0x01 - Tuned – 0xFF - Not Available • If the radio is tuned, instantaneous values are set for the signal information fields below. • 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 dbm; valid for CDMA, HDR, GSM, WCDMA, and LTE.

rscp	• Received signal code power in 1/10 dbm. • 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. • When the phase is unknown, 0xFFFFFFFF is used.

8.452.2 Field Documentation

8.452.2.1 int32_t nas_rxInfo::ecio

8.452.2.2 uint8_t nas_rxInfo::isRadioTuned

8.452.2.3 uint32_t nas_rxInfo::phase

8.452.2.4 int32_t nas_rxInfo::rscp

8.452.2.5 int32_t nas_rxInfo::rsrp

8.452.2.6 int32_t nas_rxInfo::rxPower

8.453 nas_RxSigInfo Struct Reference

Data Fields

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

8.453.1 Detailed Description

This structure contains the parameters for Rx Signal Info.

Parameters

rxChainIndex	• Rx antenna path • Valid Values – 0 - Primary Rx – 1 - Diversity Rx
isRadioTuned	• Rx path is tuned to a channel or Not • Values – 0x00 - Not tuned – 0x01 - Tuned

Note

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

Parameters

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

8.453.2 Field Documentation

8.453.2.1 uint8_t nas_RxSigInfo::isRadioTuned

8.453.2.2 int32_t nas_RxSigInfo::rsrp

8.453.2.3 uint8_t nas_RxSigInfo::rxChainIndex

8.453.2.4 int32_t nas_RxSigInfo::rxPower

8.454 nas_rxSignalStrengthListElement Struct Reference

Data Fields

- int16_t rxSignalStrength
- uint8_t radioIf

8.454.1 Detailed Description

This structure contains the Received Signal Strength Information

Parameters

rxSignalStrength	• Received signal strength in dBm – For CDMA and UMTS, this indicates forward link pilotEc. – For GSM, the received signal strength. – For LTE, this indicates the total received wideband power observed by UE.
radioIf	• Radio interface technology of the signal being radio_if measured – 0x00 – RADIO_IF_NO_SVC – None (no service) – 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X – 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000 HRPD (1xEV-DO) – 0x03 – RADIO_IF_AMPS – AMPS – 0x04 – RADIO_IF_GSM – GSM – 0x05 – RADIO_IF_UMTS – UMTS – 0x08 – RADIO_IF_LTE – LTE

Note

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

8.454.2 Field Documentation

8.454.2.1 uint8_t nas_rxSignalStrengthListElement::radioIf

8.454.2.2 int16_t nas_rxSignalStrengthListElement::rxSignalStrength

8.455 nas_SccRxInfo Struct Reference

Data Fields

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

8.455.1 Detailed Description

This structure contains information about the SccRxInfo parameters.

Parameters

rsrq	• Current reference signal • Receive quality in 1/10 dB resolution
snr	• Reference signal signal-to-noise ratio in dB. • Range -10 to 30
numInstances	• Number of sets of the following – rxChainIndex – isRadioTuned – rxPower – rsrp
sigInfo	• See nas_RxSigInfo for more information
TlvPresent	• Tlv Present.

8.455.2 Field Documentation

8.455.2.1 uint8_t nas_SccRxInfo::numInstances

8.455.2.2 int32_t nas_SccRxInfo::rsrq

8.455.2.3 nas_RxSigInfo nas_SccRxInfo::sigInfo[255]

8.455.2.4 int16_t nas_SccRxInfo::snr

8.455.2.5 uint8_t nas_SccRxInfo::TlvPresent

8.456 nas_serviceProviderName Struct Reference

Data Fields

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

8.456.1 Detailed Description

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.

8.456.2 Field Documentation

8.456.2.1 uint8_t nas_serviceProviderName::displayCondition

8.456.2.2 uint8_t nas_serviceProviderName::spn[255]

8.456.2.3 uint8_t nas_serviceProviderName::spnLength

8.457 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]

8.457.1 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 • Values: – 0 - Not Registered; mobile is not currently searching for a new network to provide service – 1 - Registered with a network – 2 - Not registered, but mobile is currently searching for a new network to provide service – 3 - Registration denied by visible network – 4 - Registration state is unknown
csAttachState	• CS Attach State - Circuit-switched domain attach state of the mobile • Values: – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached
psAttachState	• PS Attach State - Packet-switched domain attach state of the mobile • Values: – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached
selNetwork	• Selected Network - Type of selected radio access network • Values: – 0 - Unknown – 1 - 3GPP2 network – 2 - 3GPP network
numRadio-Interfaces	• In Use Radio Interfaces Number – Number of radio interfaces currently in use – defaults to zero
radioInterface	• Radio Interface currently in use • Values: – 0x00 - RADIO_IF_NO_SVC - None(no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE

8.457.2 Field Documentation

8.457.2.1 uint8_t nas_servSystem::csAttachState

8.457.2.2 uint8_t nas_servSystem::numRadioInterfaces

8.457.2.3 uint8_t nas_servSystem::psAttachState

8.457.2.4 uint8_t nas_servSystem::radioInterface[32]

8.457.2.5 uint8_t nas_servSystem::regState

8.457.2.6 uint8_t nas_servSystem::selNetwork

8.458 nas_sidNid Struct Reference

Data Fields

- uint16_t nid
- uint16_t sid

8.458.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

nid	• Network ID
sid	• System ID

8.458.2 Field Documentation

8.458.2.1 uint16_t nas_sidNid::nid

8.458.2.2 uint16_t nas_sidNid::sid

8.459 nas_SignalStrengthTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- int8_t signalStrength
- uint32_t radioInterface

8.459.1 Detailed Description

Parameters

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

8.459.2 Field Documentation

8.459.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

8.459.2.2 int8_t nas_SignalStrengthTlv::signalStrength

8.459.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

8.460 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

8.460.1 Detailed Description

Parameters

rxSignal-StrengthDelta	RSSI delta(in dBm) at which an event report indication
ecioDelta	ecio delta
ioDelta	io delta
sinrDelta	sinr delta
rsrqDelta	rsrq delta
ecioThreshold-ListLen	
ecioThreshold-List	
sinrThreshold-ListLen	
sinrThreshold-List	
lteSnrDelta	lte snr delta
lteRsrpDelta	lte rsrp delta

8.460.2 Field Documentation

8.460.2.1 uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta

8.460.2.2 int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.460.2.3 uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.460.2.4 uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta

8.460.2.5 uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta

8.460.2.6 uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta

8.460.2.7 uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta

8.460.2.8 uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.460.2.9 uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta

8.460.2.10 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.460.2.11 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.461 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

8.461.1 Detailed Description

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

8.461.2 Field Documentation

8.461.2.1 nas_ecioListElement nas_SLQSSignalStrengthsInformation::ecioInfo

8.461.2.2 nas_errorRateListElement nas_SLQSSignalStrengthsInformation::errorRateInfo

8.461.2.3 uint32_t nas_SLQSSignalStrengthsInformation::io

8.461.2.4 nas_lteRsrpinformation nas_SLQSSignalStrengthsInformation::lteRsrpinfo

8.461.2.5 nas_lteSnrinformation nas_SLQSSignalStrengthsInformation::lteSnrinfo

8.461.2.6 nas_rsrqInformation nas_SLQSSignalStrengthsInformation::rsrqInfo

8.461.2.7 nas_rxSignalStrengthListElement nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.461.2.8 uint8_t nas_SLQSSignalStrengthsInformation::sinr

8.462 nas_SLQSSignalStrengthsTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- nas_SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo

8.462.1 Detailed Description

Parameters

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

8.462.2 Field Documentation

8.462.2.1 nas_SLQSSignalStrengthsInformation nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo

8.462.2.2 uint8_t nas_SLQSSignalStrengthsTlv::TlvPresent

8.463 nas_SrvStatusInfo Struct Reference

Data Fields

- uint8_t srvStatus
- uint8_t isPrefDataPath

8.463.1 Detailed Description

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

Parameters

srvStatus	• Service status of the system. – 0x00 - No service – 0x01 - Limited service – 0x02 - Service – 0x03 - Limited regional service – 0x04 - Power save – 0xFF - Not Available

isPrefDataPath	• Whether the RAT is the preferred data path. – 0x00 - Not preferred – 0x01 - Preferred – 0xFF - Not Available

8.463.2 Field Documentation

8.463.2.1 uint8_t nas_SrvStatusInfo::isPrefDataPath

8.463.2.2 uint8_t nas_SrvStatusInfo::srvStatus

8.464 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

8.464.1 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. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
srvDomain	• Service domain registered on the system. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available

srvCapability-Valid	• Indicates whether the service capability is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
srvCapability	• Current system's service capability. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
roamStatusValid	• Indicates whether the roaming status is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
roamStatus	• Current roaming status. – 0x00 - Off – 0x01 - On – 0x02 - Blinking – 0x03 - Out of the neighborhood – 0x04 - Out of the building – 0x05 - Preferred system – 0x06 - Available system – 0x07 - Alliance partner – 0x08 - Premium partner – 0x09 - Full service – 0x0A - Partial service – 0x0B - Banner is on – 0x0C - Banner is off – 0x0D to 0x3F - Reserved for Standard Enhanced Roaming Indicator Numbers – 0x40 to 0x7F - Reserved for Non-Standard Enhanced Roaming Indicator Numbers – 0x40 to 0xFF - Reserved. – 0xFF - Not Available • Values from 0x02 onward are only applicable for 3GPP2
isSysForbidden-Valid	• Indicates whether the forbidden system is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

isSysForbidden	• Whether the system is forbidden. – 0x00 - Not forbidden – 0x01 - Forbidden – 0xFF - Not Available

8.464.2 Field Documentation

8.464.2.1 uint8_t nas_sysInfoCommon::isSysForbidden

8.464.2.2 uint8_t nas_sysInfoCommon::isSysForbiddenValid

8.464.2.3 uint8_t nas_sysInfoCommon::roamStatus

8.464.2.4 uint8_t nas_sysInfoCommon::roamStatusValid

8.464.2.5 uint8_t nas_sysInfoCommon::srvCapability

8.464.2.6 uint8_t nas_sysInfoCommon::srvCapabilityValid

8.464.2.7 uint8_t nas_sysInfoCommon::srvDomain

8.464.2.8 uint8_t nas_sysInfoCommon::srvDomainValid

8.465 nas_TDSCDMAECIOThresh Struct Reference

Data Fields

- uint8_t TDSCDMAECIOThreshListLen
- float * pTDSCDMAECIOThreshList

8.465.1 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 • Maximum of 32 values.

8.465.2 Field Documentation

8.465.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.465.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.466 nas_TDSCDMARSCPThresh Struct Reference

Data Fields

- uint8_t TDSCDMARSCPThreshListLen
- int16_t * pTDSCDMARSCPThreshList

8.466.1 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) • Maximum of 32 values • Range for RSCP values: -120 to -25 (in dBm).

8.466.2 Field Documentation

8.466.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.466.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.467 nas_TDSCDMARSSIThresh Struct Reference

Data Fields

- uint8_t TDSCDMARSSIThreshListLen
- float * pTDSCDMARSSIThreshList

8.467.1 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 • Maximum of 32 values.

8.467.2 Field Documentation

8.467.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.467.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.468 nas_TDSCDMASINRCONFThresh Struct Reference

Data Fields

- uint8_t TDSCDMASINRCONFThreshListLen
- float * pTDSCDMASINRCONFThreshList

8.468.1 Detailed Description

Parameters

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

8.468.2 Field Documentation

8.468.2.1 float* nas_TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

8.468.2.2 uint8_t nas_TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

8.469 nas_TDSCDMASINRThresh Struct Reference

Data Fields

- uint8_t TDSCDMASINRThreshListLen
- float * pTDSCDMASINRThreshList

8.469.1 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 • Maximum of 32 values

8.469.2 Field Documentation

8.469.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.469.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.470 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

8.470.1 Detailed Description

This structure contains the parameters for Network Time.

Parameters

year	• Year
month	• Month • 1 is January and 12 is December
day	• Day • Range - 1 to 31
hour	• Hour • Range - 0 to 59
minute	• Minute • Range - 0 to 59
second	• Second • Range - 0 to 59
dayOfWeek	• Day of the week • 0 is Monday and 6 is Sunday

timeZone	• Offset from Universal time • The difference between local time and Universal time, in increments of 15 min • Signed Value
dayLtSavingAdj	• Daylight saving adjustment in hours • Possible values - 0, 1, and 2. • This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVDO
radioInterface	• Radio interface from which the information comes • Values – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA -TD-SCDMA
TlvPresent	• Tlv Present.

8.470.2 Field Documentation

8.470.2.1 uint8_t nas_timeInfo::day

8.470.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

8.470.2.3 uint8_t nas_timeInfo::dayOfWeek

8.470.2.4 uint8_t nas_timeInfo::hour

8.470.2.5 uint8_t nas_timeInfo::minute

8.470.2.6 uint8_t nas_timeInfo::month

8.470.2.7 uint8_t nas_timeInfo::radioInterface

8.470.2.8 uint8_t nas_timeInfo::second

8.470.2.9 int8_t nas_timeInfo::timeZone

8.470.2.10 uint8_t nas_timeInfo::TlvPresent

8.470.2.11 uint16_t nas_timeInfo::year

8.471 nas_trueIMSI Struct Reference

Data Fields

- uint8_t mccT [3]
- uint16_t imsiT1112
- uint8_t imsiTS1 [7]
- uint8_t imsiTS2 [3]
- uint8_t imsiTaddrNum

8.471.1 Detailed Description

Parameters

mccT	• ASCII character representation of MCC_T
imsiT1112	• ASCII character representation of IMSI_T_11_12 value – 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 – 0xFF - Not Available

8.471.2 Field Documentation

8.471.2.1 uint16_t nas_trueIMSI::imsiT1112

8.471.2.2 uint8_t nas_trueIMSI::imsiTaddrNum

8.471.2.3 uint8_t nas_trueIMSI::imsiTS1[7]

8.471.2.4 uint8_t nas_trueIMSI::imsiTS2[3]

8.471.2.5 uint8_t nas_trueIMSI::mccT[3]

8.472 nas_txInfo Struct Reference

Data Fields

- uint8_t isInTraffic
- int32_t txPower

8.472.1 Detailed Description

Parameters

isInTraffic	• Whether the device is in traffic. – 0x00 - not in traffic – 0x01 - in traffic • The txPower field is only meaningful when in the device is in traffic.
txPower	• Tx power value in 1/10 dbm.

8.472.2 Field Documentation

8.472.2.1 uint8_t nas_txInfo::isInTraffic

8.472.2.2 int32_t nas_txInfo::txPower

8.473 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]

8.473.1 Detailed Description

This structure contains information about the UMTS Network.

Parameters

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

uarfcn	• UTRA absolute RF channel number. – 0xFFFF - Not Available
psc	• Primary scrambling code. – 0xFFFF - Not Available
rscp	• Received signal code power. – 0xFFFF - Not Available
ecio	• ECIO(Signal-to-Interference-ratio). – 0xFFFF - Not Available
umtsInst	• Provides the number of set of UMTS info instances. • 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. • If 0(zero), then no information follows it.
GeranInstInfo[NAS_MAX_DESCRIPTION_LENGTH]	• See nas_geranInstInfo for more information.

8.473.2 Field Documentation

8.473.2.1 uint16_t nas_UMTSInfo::cellID

8.473.2.2 int16_t nas_UMTSInfo::ecio

8.473.2.3 uint8_t nas_UMTSInfo::geranInst

8.473.2.4 nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]

8.473.2.5 uint16_t nas_UMTSInfo::lac

8.473.2.6 uint8_t nas_UMTSInfo::plmn[3]

8.473.2.7 uint16_t nas_UMTSInfo::psc

8.473.2.8 int16_t nas_UMTSInfo::rscp

8.473.2.9 uint16_t nas_UMTSInfo::uarfcn

8.473.2.10 uint8_t nas_UMTSInfo::umtsInst

8.473.2.11 nas_UMTSinstInfo nas_UMTSInfo::UMTSInstInfo[255]

8.474 nas_UMTSinstInfo Struct Reference

Data Fields

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

8.474.1 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).

8.474.2 Field Documentation

8.474.2.1 int16_t nas_UMTSinstInfo::umtsEcio

8.474.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

8.474.2.3 int16_t nas_UMTSinstInfo::umtsRscp

8.474.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

8.475 nas_umtsLTENbrCell Struct Reference

Data Fields

- uint16_t earfcn
- uint16_t pci
- uint32_t rsrp
- uint32_t rsrq
- int16_t srxlev
- uint8_t cellIsTDD

8.475.1 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. • 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. • 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.

8.475.2 Field Documentation

8.475.2.1 uint8_t nas_umtsLTENbrCell::cellIsTDD

8.475.2.2 uint16_t nas_umtsLTENbrCell::earfcn

8.475.2.3 uint16_t nas_umtsLTENbrCell::pci

8.475.2.4 uint32_t nas_umtsLTENbrCell::rsrp

8.475.2.5 uint32_t nas_umtsLTENbrCell::rsrq

8.475.2.6 int16_t nas_umtsLTENbrCell::srxlev

8.476 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

8.476.1 Detailed Description

This structure contains the parameters for Universal Time Information.

Parameters

year	• Year.
month	• Month. – 1 is January and 12 is December.
day	• Day. – Range 1 to 31.
hour	• Hour. – Range 0 to 59.
minute	• Minute. – Range 0 to 59.
second	• Second. – Range 0 to 59.
dayOfWeek	• Day of the Week. – 0 is Monday and 6 is Sunday.

8.476.2 Field Documentation

8.476.2.1 uint8_t nas_UniversalTime::day

8.476.2.2 uint8_t nas_UniversalTime::dayOfWeek

8.476.2.3 uint8_t nas_UniversalTime::hour

8.476.2.4 uint8_t nas_UniversalTime::minute

8.476.2.5 uint8_t nas_UniversalTime::month

8.476.2.6 uint8_t nas_UniversalTime::second

8.476.2.7 uint16_t nas_UniversalTime::year

8.477 nas_wcdmaCellInfo Struct Reference

Data Fields

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

8.477.1 Detailed Description

This structure contains information about the WCDMA Cell.

Parameters

psc	• Primary scrambling code. • Range: 0 to 511.
cpich_rscp	• Absolute power level (in 1/10 dBm) of the common pilot channel as received by the UE. • Range: -120.0 dBm to -25.0 dBm
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. • Range: -50.0 dB to 0.
srxlev	• Cell selection Rx level (Srxlev) value. • Range: -128 to 128. • This field is only valid when ue_in_idle is TRUE.

8.477.2 Field Documentation

8.477.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

8.477.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

8.477.2.3 uint16_t nas_wcdmaCellInfo::psc

8.477.2.4 int16_t nas_wcdmaCellInfo::srxlev

8.478 nas_WCDMAECIOThresh Struct Reference

Data Fields

- uint8_t WCDMAECIOThreshListLen
- int16_t * pWCDMAECIOThreshList

8.478.1 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) • Maximum of 32 values • Range for ECIO values: -31.5 to 0 (in dB)

8.478.2 Field Documentation

8.478.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

8.478.2.2 uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen

8.479 nas_WCDMAInfoLTENeighborCell Struct Reference

Data Fields

- uint32_t wcdmaRRCState
- uint8_t umtsLTENbrCellLen
- nas_umtsLTENbrCell UMTSLTENbrCell [255]

8.479.1 Detailed Description

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

Parameters

wcdmaRRC-State	• WCDMA RRC states. • Defined in 3GPP TS 25.331 • Values: – 0x00 - NAS_WCDMA_RRC_STATE_DISCONNECTED * WCDMA RRC State is IDLE – 0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH * WCDMA RRC state is CELL_PCH – 0x02 - NAS_WCDMA_RRC_STATE_URA_PCH * WCDMA RRC state is URA_PCH – 0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH * WCDMA RRC state is CELL_FACH – 0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH * WCDMA RRC state is CELL_DCH
umtsLTENbr-CellLen	• Number of sets of UMTS LTE Neighbors.
UMTSLTENbr-Cell	• See nas_umtsLTENbrCell for more information.

8.479.2 Field Documentation

8.479.2.1 nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.479.2.2 uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.479.2.3 uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCState

8.480 nas_WCDMARSSIThresh Struct Reference

Data Fields

- uint8_t WCDMARSSIThreshListLen
- int16_t * pWCDMARSSIThreshList

8.480.1 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) • Maximum of 32 values. • Range for RSSI values: -121 to 0 (in dBm)

8.480.2 Field Documentation

8.480.2.1 int16_t* nas_WCDMARSSIThresh::pWCDMARSSIThreshList

8.480.2.2 uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen

8.481 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

8.481.1 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.. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
lac	• Location area code. • Only applies to 3GPP. – 0xFFFF - Not Available
cellIdValid	• Indicates whether the cell ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
cellId	• Cell ID. – 0xFFFFFFFF - Not Available
regRejectInfo-Valid	• Indicates whether the registration reject information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
rejectSrvDomain	• Type of service domain in which the registration is rejected. – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
rejCause	• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. – 0xFF - Not Available

networkIdValid	• Indicates whether the network ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
MCC[NAS_PLMN_LENGTH]	• Mobile Country Code. • MCC digits in ASCII characters
MNC[NAS_PLMN_LENGTH]	• Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.
hsCallStatusValid	• Indicates whether the high-speed call status is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
hsCallStatus	• Call status on high speed. • Only applicable for WCDMA. – 0x00 - HSDPA and HSUPA are unsupported – 0x01 - HSDPA is supported – 0x02 - HSUPA is supported – 0x03 - HSDPA and HSUPA are supported – 0x04 - HSDPA+ is supported – 0x05 - HSDPA+ and HSUPA are supported – 0x06 - Dual-cell HSDPA+ is supported – 0x07 - Dual-cell HSDPA+ and HSUPA are supported – 0xFF - Not Available
hsIndValid	• Indicates whether high-speed service indication is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

hsInd	• High-speed service indication • Only applicable for WCDMA. – 0x00 - HSDPA and HSUPA are unsupported – 0x01 - HSDPA is supported – 0x02 - HSUPA is supported – 0x03 - HSDPA and HSUPA are supported – 0x04 - HSDPA+ is supported – 0x05 - HSDPA+ and HSUPA are supported – 0x06 - Dual-cell HSDPA+ is supported – 0x07 - Dual-cell HSDPA+ and HSUPA are supported – 0xFF - Not Available
pscValid	• Indicates whether primary scrambling code is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
psc	• Primary scrambling code. – 0xFFFF - Not Available

8.481.2 Field Documentation

8.481.2.1 uint32_t nas_WCDMASysInfo::cellId

8.481.2.2 uint8_t nas_WCDMASysInfo::cellIdValid

8.481.2.3 uint8_t nas_WCDMASysInfo::hsCallStatus

8.481.2.4 uint8_t nas_WCDMASysInfo::hsCallStatusValid

8.481.2.5 uint8_t nas_WCDMASysInfo::hsInd

8.481.2.6 uint8_t nas_WCDMASysInfo::hsIndValid

8.481.2.7 uint16_t nas_WCDMASysInfo::lac

8.481.2.8 uint8_t nas_WCDMASysInfo::lacValid

8.481.2.9 uint8_t nas_WCDMASysInfo::MCC[3]

8.481.2.10 uint8_t nas_WCDMASysInfo::MNC[3]

8.481.2.11 uint8_t nas_WCDMASysInfo::networkIdValid

8.481.2.12 uint16_t nas_WCDMASysInfo::psc

8.481.2.13 uint8_t nas_WCDMASysInfo::pscValid

8.481.2.14 uint8_t nas_WCDMASysInfo::regRejectInfoValid

8.481.2.15 uint8_t nas_WCDMASysInfo::rejCause

8.481.2.16 uint8_t nas_WCDMASysInfo::rejectSrvDomain

8.481.2.17 nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA

8.482 nas_wcdmaUARFCN Struct Reference

Data Fields

- uint8_t status
- uint32_t uarfcn

8.482.1 Detailed Description

Parameters

status	• 0 - Disable • 1 - Enable
uarfcn	• UARFCN to which UE is locked

8.482.2 Field Documentation

8.482.2.1 uint8_t nas_wcdmaUARFCN::status

8.482.2.2 uint32_t nas_wcdmaUARFCN::uarfcn

8.483 NASBandPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t band_pref

8.483.1 Field Documentation

8.483.1.1 uint64_t NASBandPreferenceTlv::band_pref

8.483.1.2 uint8_t NASBandPreferenceTlv::TlvPresent

8.484 nasCellLocationInfoResp Struct Reference

Data Fields

- GERANInfo * pGERANInfo
- UMTSInfo * pUMTSInfo

- CDMAInfo * pCDMAInfo
- LTEInfoIntrafreq * pLTEInfoIntrafreq
- LTEInfoInterfreq * pLTEInfoInterfreq
- LTEInfoNeighboringGSM * pLTEInfoNeighboringGSM
- LTEInfoNeighboringWCDMA * pLTEInfoNeighboringWCDMA
- ULONG * pUMTSCellID
- WCDMAInfoLTENeighborCell * pWCDMAInfoLTENeighborCell

8.484.1 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. • 0xFFFFFFFF indicates cell ID information is not present.
pWCDMAInfoLT-ENeighborCell	• See WCDMAInfoLTENeighborCell for more information.

8.484.2 Field Documentation

8.484.2.1 CDMAInfo* nasCellLocationInfoResp::pCDMAInfo

8.484.2.2 GERANInfo* nasCellLocationInfoResp::pGERANInfo

8.484.2.3 LTEInfoInterfreq* nasCellLocationInfoResp::pLTEInfoInterfreq

8.484.2.4 LTEInfoIntrafreq* nasCellLocationInfoResp::pLTEInfoIntrafreq

8.484.2.5 LTEInfoNeighboringGSM∗ nasCellLocationInfoResp::pLTEInfoNeighboringGSM

8.484.2.6 LTEInfoNeighboringWCDMA∗ nasCellLocationInfoResp::pLTEInfoNeighboringWCDMA

8.484.2.7 ULONG∗ nasCellLocationInfoResp::pUMTSCellID

8.484.2.8 UMTSInfo∗ nasCellLocationInfoResp::pUMTSInfo

8.484.2.9 WCDMAInfoLTENeighborCell∗ nasCellLocationInfoResp::pWCDMAInfoLTENeighborCell

8.485 NASEmergencyModeTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t EmerMode

8.485.1 Field Documentation

8.485.1.1 uint8_t NASEmergencyModeTlv::EmerMode

8.485.1.2 uint8_t NASEmergencyModeTlv::TlvPresent

8.486 nasGet3GPP2SubscriptionInfoReq Struct Reference

Data Fields

- BYTE namID

8.486.1 Detailed Description

This structure contains the Get3GPP2SubscriptionInfo request parameters

Parameters

namID	[Mandatory] • 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.

8.486.2 Field Documentation

8.486.2.1 BYTE nasGet3GPP2SubscriptionInfoReq::namID

8.487 nasGet3GPP2SubscriptionInfoResp Struct Reference

Data Fields

- namName ∗ pNAMNameInfo
- dirNum ∗ pDirNum
- homeSIDNID ∗ pHomeSIDNID
- minBasedIMSI ∗ pMinBasedIMSI

- trueIMSI * pTrueIMSI
- CDMAChannel * pCDMAChannel

8.487.1 Detailed Description

This structure contains the SLQSNasGet3GPP2Subscription response parameters.

Parameters

pNAMNameInfo	[Optional] • See namName for more information
pDirNum	[Optional] • See dirNum for more information
pHomeSIDNID	[Optional] • See homeSIDNID for more information
pMinBasedIMSI	[Optional] • See minBasedIMSI for more information
pTrueIMSI	[Optional] • See trueIMSI for more information
pCDMAChannel	[Optional] • See CDMAChannel for more information

8.487.2 Field Documentation

8.487.2.1 CDMAChannel* nasGet3GPP2SubscriptionInfoResp::pCDMAChannel

8.487.2.2 dirNum* nasGet3GPP2SubscriptionInfoResp::pDirNum

8.487.2.3 homeSIDNID* nasGet3GPP2SubscriptionInfoResp::pHomeSIDNID

8.487.2.4 minBasedIMSI* nasGet3GPP2SubscriptionInfoResp::pMinBasedIMSI

8.487.2.5 namName* nasGet3GPP2SubscriptionInfoResp::pNAMNameInfo

8.487.2.6 trueIMSI* nasGet3GPP2SubscriptionInfoResp::pTrueIMSI

8.488 nasGetHDRColorCodeResp Struct Reference

Data Fields

- BYTE * pColorCode

8.488.1 Detailed Description

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

Parameters

pColorCode	[Optional] • Color code value • Color code corresponding to the sector to which the AT is sending the access probe • See 3GPP2 C.S0024-B V3.0, Section 7.11.6.2.1 for more information. – 0xFF - Not Available

8.488.2 Field Documentation

8.488.2.1 BYTE* nasGetHDRColorCodeResp::pColorCode

8.489 nasGetLTECphyCa Struct Reference

Data Fields

- PhyCaAggScellIndType sPhyCaAggScellIndType
- PhyCaAggScellDlBw sPhyCaAggScellDlBw
- PhyCaAggScellInfo sPhyCaAggScellInfo
- PhyCaAggPcellInfo sPhyCaAggPcellInfo
- PhyCaAggScellIndex sPhyCaAggScellIndex

8.489.1 Field Documentation

8.489.1.1 PhyCaAggPcellInfo nasGetLTECphyCa::sPhyCaAggPcellInfo

8.489.1.2 PhyCaAggScellDlBw nasGetLTECphyCa::sPhyCaAggScellDlBw

8.489.1.3 PhyCaAggScellIndex nasGetLTECphyCa::sPhyCaAggScellIndex

8.489.1.4 PhyCaAggScellIndType nasGetLTECphyCa::sPhyCaAggScellIndType

8.489.1.5 PhyCaAggScellInfo nasGetLTECphyCa::sPhyCaAggScellInfo

8.490 NasGetLTECphyCaInfo Struct Reference

Data Fields

- NASPhyCaAggScellIndType PhyCaAggScellIndType
- NASPhyCaAggScellDlBw PhyCaAggScellDlBw
- NASPhyCaAggScellInfo PhyCaAggScellInfo
- NASPhyCaAggPcellInfo PhyCaAggPcellInfo
- NASPhyCaAggScellIndex PhyCaAggScellIndex

8.490.1 Field Documentation

8.490.1.1 NASPhyCaAggPcellInfo NasGetLTECphyCaInfo::PhyCaAggPcellInfo

8.490.1.2 NASPhyCaAggScellDlBw NasGetLTECphyCaInfo::PhyCaAggScellDlBw

8.490.1.3 NASPhyCaAggScellIndex NasGetLTECphyCaInfo::PhyCaAggScellIndex

8.490.1.4 NASPhyCaAggScellIndType NasGetLTECphyCaInfo::PhyCaAggScellIndType

8.490.1.5 NASPhyCaAggScellInfo NasGetLTECphyCaInfo::PhyCaAggScellInfo

8.491 nasGetLTECphyCaResp Struct Reference

Data Fields

- PhyCaAggScellIndType * pPhyCaAggScellIndType
- PhyCaAggScellDlBw * pPhyCaAggScellDlBw
- PhyCaAggScellInfo * pPhyCaAggScellInfo
- PhyCaAggPcellInfo * pPhyCaAggPcellInfo
- PhyCaAggScellIndex * pPhyCaAggScellIndex

8.491.1 Field Documentation

8.491.1.1 PhyCaAggPcellInfo* nasGetLTECphyCaResp::pPhyCaAggPcellInfo

8.491.1.2 PhyCaAggScellDlBw* nasGetLTECphyCaResp::pPhyCaAggScellDlBw

8.491.1.3 PhyCaAggScellIndex* nasGetLTECphyCaResp::pPhyCaAggScellIndex

8.491.1.4 PhyCaAggScellIndType* nasGetLTECphyCaResp::pPhyCaAggScellIndType

8.491.1.5 PhyCaAggScellInfo* nasGetLTECphyCaResp::pPhyCaAggScellInfo

8.492 nasGetSigInfoResp Struct Reference

Data Fields

- CDMASSInfo * pCDMASSInfo
- HDRSSInfo * pHDRSSInfo
- INT8 * pGSMSSInfo
- CDMASSInfo * pWCDMASSInfo
- LTESSInfo * pLTESSInfo
- INT8 * pTDSCDMASigInfoRscp
- TDSCDMASigInfoExt * pTDSCDMASigInfoExt

8.492.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

Parameters

pCDMASSInfo	[Optional] • See CDMASSInfo for more information
pHDRSSInfo	[Optional] • See HDRSSInfo for more information
pGSMSSInfo	[Optional] • GSM signal strength is the RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal.

pWCDMASSInfo	[Optional] • See CDMASSInfo for more information
pLTESSInfo	[Optional] • See LTESSInfo for more information
pTDSCDMASig-InfoRscp	[Optional] • RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. • Measurement range: -120 dBm to -25 dBm.
pTDSCDMASig-InfoExt	[Optional] • See TDSCDMASigInfoExt for more information.

8.492.2 Field Documentation

8.492.2.1 CDMASSInfo* nasGetSigInfoResp::pCDMASSInfo

8.492.2.2 INT8* nasGetSigInfoResp::pGSMSSInfo

8.492.2.3 HDRSSInfo* nasGetSigInfoResp::pHDRSSInfo

8.492.2.4 LTESSInfo* nasGetSigInfoResp::pLTESSInfo

8.492.2.5 TDSCDMASigInfoExt* nasGetSigInfoResp::pTDSCDMASigInfoExt

8.492.2.6 INT8* nasGetSigInfoResp::pTDSCDMASigInfoRscp

8.492.2.7 CDMASSInfo* nasGetSigInfoResp::pWCDMASSInfo

8.493 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

8.493.1 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. • 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. • 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. – 0x00 - Voice is not supported – 0x01 - Voice is supported
pGSMCipher-DomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pWCDMA-CipherDomain-SysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched

8.493.2 Field Documentation

8.493.2.1 AddCDMASysInfo∗ nasGetSysInfoResp::pAddCDMASysInfo

8.493.2.2 AddSysInfo∗ nasGetSysInfoResp::pAddGSMSysInfo

8.493.2.3 WORD∗ nasGetSysInfoResp::pAddHDRSysInfo

8.493.2.4 WORD∗ nasGetSysInfoResp::pAddLTESysInfo

8.493.2.5 AddSysInfo∗ nasGetSysInfoResp::pAddWCDMASysInfo

8.493.2.6 SrvStatusInfo∗ nasGetSysInfoResp::pCDMASrvStatusInfo

8.493.2.7 CDMASysInfo∗ nasGetSysInfoResp::pCDMASysInfo

8.493.2.8 CallBarringSysInfo∗ nasGetSysInfoResp::pGSMCallBarringSysInfo

8.493.2.9 BYTE∗ nasGetSysInfoResp::pGSMCipherDomainSysInfo

8.493.2.10 GSMSrvStatusInfo∗ nasGetSysInfoResp::pGSMSrvStatusInfo

8.493.2.11 GSMSysInfo∗ nasGetSysInfoResp::pGSMSysInfo

8.493.2.12 SrvStatusInfo∗ nasGetSysInfoResp::pHDRSrvStatusInfo

8.493.2.13 HDRSysInfo∗ nasGetSysInfoResp::pHDRSysInfo

8.493.2.14 GSMSrvStatusInfo∗ nasGetSysInfoResp::pLTESrvStatusInfo

8.493.2.15 LTESysInfo∗ nasGetSysInfoResp::pLTESysInfo

8.493.2.16 BYTE∗ nasGetSysInfoResp::pLTEVoiceSupportSysInfo

8.493.2.17 CallBarringSysInfo∗ nasGetSysInfoResp::pWCDMACallBarringSysInfo

8.493.2.18 BYTE∗ nasGetSysInfoResp::pWCDMACipherDomainSysInfo

8.493.2.19 GSMSrvStatusInfo∗ nasGetSysInfoResp::pWCDMASrvStatusInfo

8.493.2.20 WCDMASysInfo∗ nasGetSysInfoResp::pWCDMASysInfo

8.494 nasGetTxRxInfoReq Struct Reference

Data Fields

- BYTE radio_if

8.494.1 Detailed Description

This structure contains the GetTxRxInfoReq request parameters

Parameters

radio_if	[Mandatory] • Radio interface technology of the signal being measured • Valid Values – 0x01 - NAS_RADIO_IF_CDMA_1X - CDMA – 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - HDR – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE

8.494.2 Field Documentation

8.494.2.1 BYTE nasGetTxRxInfoReq::radio_if

8.495 nasGetTxRxInfoResp Struct Reference

Data Fields

- rxInfo ∗ pRXChain0Info
- rxInfo ∗ pRXChain1Info
- txInfo ∗ pTXInfo

8.495.1 Detailed Description

This structure contains the GetTxRxInfoResp response parameters.

Parameters

pRXChain0Info	[Optional] • See rxInfo for more information.
pRXChain1Info	[Optional] • See rxInfo for more information.
pTXInfo	[Optional] • See txInfo for more information.

8.495.2 Field Documentation

8.495.2.1 rxInfo* nasGetTxRxInfoResp::pRXChain0Info

8.495.2.2 rxInfo* nasGetTxRxInfoResp::pRXChain1Info

8.495.2.3 txInfo* nasGetTxRxInfoResp::pTXInfo

8.496 NASGWAcqOrderPrefTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t GWAcqOrderPref

8.496.1 Field Documentation

8.496.1.1 uint32_t NASGWAcqOrderPrefTlv::GWAcqOrderPref

8.496.1.2 uint8_t NASGWAcqOrderPrefTlv::TlvPresent

8.497 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

8.497.1 Detailed Description

This structure contains the SLQSNasIndicationRegisterExt request parameters.

Parameters

pSystem-SelectionInd	[Optional] • System Selection Preference indication registration. The following callbacks would not be invoked if the indication is disabled. tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem – 0x00 - Disable – 0x01 - Enable
pDDTMInd	[Optional] • DDTM (Data Dedicated Transmission Mode) indication registration. The following callbacks would not be invoked if the indication is disabled. tFNDDTM – 0x00 - Disable – 0x01 - Enable
pServing-SystemInd	[Optional] • Serving System indication registration. The following callbacks would not be invoked if the indication is disabled. tFNBandPreference – 0x00 - Disable – 0x01 - Enable
pDualStandBy-PrefInd	[Optional] • Dual Standby Preference indication registration.The following callbacks would not be invoked if the indication is disabled. tFNDualStandByPref – 0x00 - Disable – 0x01 - Enable
pSubscription-InfoInd	[Optional] • Subscription Information indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSubscriptionInfo – 0x00 - Disable – 0x01 - Enable
pNetworkTime-Ind	[Optional] • Network Time indication registration.The following callbacks would not be invoked if the indication is disabled. tFNNetworkTime – 0x00 - Disable – 0x01 - Enable

pSysInfoInd	[Optional] • System Information indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSysInfo – 0x00 - Disable – 0x01 - Enable
pSignalStrength-Ind	[Optional] • Signal Strength indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSigInfo – 0x00 - Disable – 0x01 - Enable
pErrorRateInd	[Optional] • Error Rate indication registration.The following callbacks would not be invoked if the indication is disabled. tFNErrRate – 0x00 - Disable – 0x01 - Enable
pHDRNewUATI-AssInd	[Optional] • HDR New UATI Assigned indication registration.The following callbacks would not be invoked if the indication is disabled. tFNHDRUATIUpdate – 0x00 - Disable – 0x01 - Enable
pHDRSession-CloseInd	[Optional] • HDR Session Closed indication registration.The following callbacks would not be invoked if the indication is disabled. tFNHDRSessionClose – 0x00 - Disable – 0x01 - Enable
pManaged-RoamingInd	[Optional] • Managed Roaming indication registration.The following callbacks would not be invoked if the indication is disabled. tFNManagedRoaming – 0x00 - Disable – 0x01 - Enable
pNetworkReject-Ind	[Optional] • Network reject information indication registration.The following callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable

pSuppressSysInfoInd	[Optional] • Controls the reporting of QMI_NAS_SYS_INFO_IND when only the reject_cause field has changed. Values: – 0x00 - Disable – 0x01 - Enable
pLTECphyCa	[Optional] • LTE Physical Carrier Aggregation Information indication registration.The following callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable

Note

Atleast one parameter must be provided as request. 'NULL' value confirms that the indication value is not sent.

8.497.2 Field Documentation

8.497.2.1 BYTE* nasIndicationRegisterReq::pDDTMInd

8.497.2.2 BYTE* nasIndicationRegisterReq::pDualStandByPrefInd

8.497.2.3 BYTE* nasIndicationRegisterReq::pErrorRateInd

8.497.2.4 BYTE* nasIndicationRegisterReq::pHDRNewUATIAssInd

8.497.2.5 BYTE* nasIndicationRegisterReq::pHDRSessionCloseInd

8.497.2.6 BYTE* nasIndicationRegisterReq::pLTECphyCa

8.497.2.7 BYTE* nasIndicationRegisterReq::pManagedRoamingInd

8.497.2.8 BYTE* nasIndicationRegisterReq::pNetworkRejectInd

8.497.2.9 BYTE* nasIndicationRegisterReq::pNetworkTimeInd

8.497.2.10 BYTE* nasIndicationRegisterReq::pServingSystemInd

8.497.2.11 BYTE* nasIndicationRegisterReq::pSignalStrengthInd

8.497.2.12 BYTE* nasIndicationRegisterReq::pSubscriptionInfoInd

8.497.2.13 BYTE* nasIndicationRegisterReq::pSuppressSysInfoInd

8.497.2.14 BYTE* nasIndicationRegisterReq::pSysInfoInd

8.497.2.15 BYTE* nasIndicationRegisterReq::pSystemSelectionInd

8.498 nasInitNetworkReg Struct Reference

Data Fields

- ULONG regAction
- MNRInfo * pMNRInfo
- ULONG * pChangeDuration
- BOOL * pMncPcsDigitStatus

8.498.1 Detailed Description

This structure contains Initiate Network Registration request parameters

Parameters

regAction	• Specifies one of the following register actions : – AUTO_REGISTER - Device registers according to its provisioning and optional parameters supplied with the command are ignored. – MANUAL_REGISTER - Device registers to a specified network and the optional Manual Network Register Information parameter pMNRInfo must also be included for the command to process successfully and supported only for 3GPP.
pMNRInfo	[Optional] • Pointer to structure MNRInfo – See MNRInfo for more information
pChange-Duration	[Optional] • Duration of the change. – 0x00 - Power cycle - Remains active until the next device power cycle – 0x01 - Permanent - Remains active through power cycles until changed by the client
pMncPcsDigit-Status	[Optional] • MNC PCS Digit Include Status – True - MNC is a 3-digit value. – False - MNC is a 2-digit value.

8.498.2 Field Documentation

8.498.2.1 ULONG* nasInitNetworkReg::pChangeDuration

8.498.2.2 BOOL* nasInitNetworkReg::pMncPcsDigitStatus

8.498.2.3 MNRInfo* nasInitNetworkReg::pMNRInfo

8.498.2.4 ULONG nasInitNetworkReg::regAction

8.499 NASLTEBandPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t LTEBandPref

8.499.1 Field Documentation

8.499.1.1 uint64_t NASLTEBandPreferenceTlv::LTEBandPref

8.499.1.2 uint8_t NASLTEBandPreferenceTlv::TlvPresent

8.500 NASLteNasReleaseInfoTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t nas_release
- uint8_t nas_major
- uint8_t nas_minor

8.500.1 Field Documentation

8.500.1.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

8.500.1.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

8.500.1.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

8.500.1.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

8.501 NASModePreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t ModePref

8.501.1 Field Documentation

8.501.1.1 uint16_t NASModePreferenceTlv::ModePref

8.501.1.2 uint8_t NASModePreferenceTlv::TlvPresent

8.502 NASNetSelPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t NetSelPref

8.502.1 Field Documentation

8.502.1.1 uint8_t NASNetSelPreferenceTlv::NetSelPref

8.502.1.2 uint8_t NASNetSelPreferenceTlv::TlvPresent

8.503 nasNetworkReject Struct Reference

Data Fields

- BYTE radioIf
- TYPE_OF_SERVICE_DOMAIN serviceDomain
- BYTE rejectCause
- PlmnID * pPlmnId
- CsgId * pCsgId

8.503.1 Detailed Description

Structure for storing the network reject indication parameters.

Parameters

radioIf	• Radio interface from which to get the information. Values: – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA - TDSCDMA
serviceDomain	• Type of service domain in which the registration is rejected. Values: – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only – 0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only – 0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched – 0x04 - SYS_SRV_DOMAIN_CAMPED - Camped

rejectCause	• Reject cause values. – 2 - IMSI unknown in HLR – 3 - Illegal MS – 4 - IMSI unknown in VLR – 5 - IMEI not accepted – 6 - Illegal ME – 7 - EPS services not allowed – 8 - EPS services and non-EPS services not allowed – 9 - UE identity cannot be derived by the network – 10 - Implicitly detached – 11 - PLMN not allowed – 12 - Location Area not allowed – 13 - Roaming not allowed in this location area – 14 - EPS services not allowed in this PLMN – 15 - No Suitable Cells In Location Area – 16 - MSC temporarily not reachable – 17 - Network failure – 18 - CS domain not available – 19 - ESM failure – 20 - MAC failure – 21 - Synch failure – 22 - Congestion – 23 - GSM authentication unacceptable – 24 - Security mode rejected, unspecified – 25 - Not authorized for this CSG – 26 - Non-EPS authentication unacceptable – 32 - Service option not supported – 33 - Requested service option not subscribed – 34 - Service option temporarily out of order – 38 - Call cannot be identified – 48 to 63 - retry upon entry into a new cell – 95 - Semantically incorrect message – 96 - Invalid mandatory information – 97 - Message type non-existent or not implemented – 98 - Message type not compatible with the protocol state – 99 - Information element non-existent or not implemented – 100 - Conditional IE error – 101 - Message not compatible with the protocol state – 111 - Protocol error, unspecified – Note - Any other value received by the mobile station shall be treated as 34, 'Service option temporarily out of order'. * Any other value received by the network shall be treated as 111, 'Protocol error, unspecified'. 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) • See PlmnID for more information

pCsgId	(optional) • See CsgId for more information.

8.503.2 Field Documentation

8.503.2.1 CsgId* nasNetworkReject::pCsgId

8.503.2.2 PlmnID* nasNetworkReject::pPlmnId

8.503.2.3 BYTE nasNetworkReject::radioIf

8.503.2.4 BYTE nasNetworkReject::rejectCause

8.503.2.5 TYPE_OF_SERVICE_DOMAIN nasNetworkReject::serviceDomain

8.504 nasNetworkTime Struct Reference

Data Fields

- UniversalTime universalTime
- BYTE * pTimeZone
- BYTE * pDayItSavAdj
- BYTE * pRadioInterface

8.504.1 Detailed Description

Structure for storing the nasSysInfo indication parameters.

Parameters

universalTime	• See UniversalTime for more information.
pTimeZone	• Time Zone. • 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. • Daylight saving adjustment in hr. – Possible values: 0, 1, and 2.

pRadioInterface	• Radio interface from which the information comes • Values – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA -TD-SCDMA

8.504.2 Field Documentation

8.504.2.1 BYTE* nasNetworkTime::pDayltSavAdj

8.504.2.2 BYTE* nasNetworkTime::pRadioInterface

8.504.2.3 BYTE* nasNetworkTime::pTimeZone

8.504.2.4 UniversalTime nasNetworkTime::universalTime

8.505 nasOperatorNameResp Struct Reference

Data Fields

- serviceProviderName * pSrvcProviderName
- operatorPLMNList * pOperatorPLMNList
- PLMNNetworkName * pPLMNNetworkName
- operatorNameString * pOperatorNameString
- PLMNNetworkNameData * pNITZInformation

8.505.1 Detailed Description

This structure contains Operator Name Data related from multiple sources.

Parameters

pSrvcProvider-Name	• Refer serviceProviderName for details (Optional). • Can provide NULL if this parameter is not required.
pOperatorPLM-NList	• Refer operatorPLMNList for details (Optional). • Can provide NULL if this parameter is not required.
pPLMNNetwork-Name	• Refer PLMNNetworkName for details (Optional). • Can provide NULL if this parameter is not required.
pOperatorName-String	• Refer operatorNameString for details (Optional). • Can provide NULL if this parameter is not required.

pNITZ-Information	• Refer PLMNNetworkNameData for details (Optional). • Can provide NULL if this parameter is not required.

8.505.2 Field Documentation

8.505.2.1 PLMNNetworkNameData* nasOperatorNameResp::pNITZInformation

8.505.2.2 operatorNameString* nasOperatorNameResp::pOperatorNameString

8.505.2.3 operatorPLMNList* nasOperatorNameResp::pOperatorPLMNList

8.505.2.4 PLMNNetworkName* nasOperatorNameResp::pPLMNNetworkName

8.505.2.5 serviceProviderName* nasOperatorNameResp::pSrvcProviderName

8.506 NASOTAMessageTlv Struct Reference

Data Fields

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

8.506.1 Field Documentation

8.506.1.1 uint8_t NASOTAMessageTlv::data_buf[2048]

8.506.1.2 uint16_t NASOTAMessageTlv::data_len

8.506.1.3 uint32_t NASOTAMessageTlv::message_type

8.506.1.4 uint8_t NASOTAMessageTlv::TlvPresent

8.507 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

8.507.1 Detailed Description

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

Parameters

pci	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
freq	• Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
dl_bw_value	• Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB for more information.
scell_state	• Scell state Values. • See NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.507.2 Field Documentation

8.507.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value

8.507.2.2 uint32_t NASPhyCaAggPcellInfo::freq

8.507.2.3 uint32_t NASPhyCaAggPcellInfo::iLTEbandValue

8.507.2.4 uint32_t NASPhyCaAggPcellInfo::pci

8.507.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

8.508 NASPhyCaAggScellDlBw Struct Reference

Data Fields

- LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- uint8_t TlvPresent

8.508.1 Detailed Description

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

Parameters

dl_bw_value	• Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB for more information.

8.508.2 Field Documentation

8.508.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellDlBw::dl_bw_value

8.508.2.2 uint8_t NASPhyCaAggScellDlBw::TlvPresent

8.509 NASPhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

8.509.1 Detailed Description

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

Parameters

scell_idx	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
TlvPresent	• Tlv Present.

8.509.2 Field Documentation

8.509.2.1 uint8_t NASPhyCaAggScellIndex::scell_idx

8.509.2.2 uint8_t NASPhyCaAggScellIndex::TlvPresent

8.510 NASPhyCaAggScellIndType Struct Reference

Data Fields

- uint32_t pci
- uint32_t freq
- LITEQMI_NAS_LTE_CPHY_SCELL_STATE scell_state
- uint8_t TlvPresent

8.510.1 Detailed Description

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

Parameters

pci	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
freq	• Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.

scell_state	• Scell state Values. • See NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.510.2 Field Documentation

8.510.2.1 uint32_t NASPhyCaAggScellIndType::freq

8.510.2.2 uint32_t NASPhyCaAggScellIndType::pci

8.510.2.3 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

8.510.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

8.511 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

8.511.1 Detailed Description

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

Parameters

pci	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
freq	• Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
dl_bw_value	• Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB for more information.
iLTEbandValue	• Band value. • Range for LTE Band class 120 to 160.

scell_state	• Scell state Values. • See NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.511.2 Field Documentation

8.511.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

8.511.2.2 uint32_t NASPhyCaAggScellInfo::freq

8.511.2.3 uint32_t NASPhyCaAggScellInfo::iLTEbandValue

8.511.2.4 uint32_t NASPhyCaAggScellInfo::pci

8.511.2.5 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

8.511.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

8.512 nasPLMNNameReq Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- BYTE * pMncPcsStatus

8.512.1 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 • Used to interpret the length of the corresponding MNC reported in the PLMN TLV(0x01). • Values – TRUE - MNC is a three-digit value. e.g. a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a two-digit value. e.g. a reported value of 90 corresponds to an MNC value of 90

Note

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

8.512.2 Field Documentation

8.512.2.1 WORD nasPLMNNameReq::mcc

8.512.2.2 WORD nasPLMNNameReq::mnc

8.512.2.3 BYTE* nasPLMNNameReq::pMncPcsStatus

8.513 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]

8.513.1 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: – 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0 – 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: – 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0 – 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: – 0x00 - Do not add the letters for the countrys initials to the name – 0x01 - Add the countrys initials and a text string to the name – 0xFF - Not specified
shortNameSB	• PLMN short name spare bits. This value is ignored if shortNameLen is zero. Values: – 0x01 - Bit 8 is spare and set to 0 in octet – 0x02 - Bits 7 and 8 are spare and set to 0 in octet n – 0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n – 0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n – 0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n – 0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n – 0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n – 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: – 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0 – 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: – 0x00 - Do not add the letters for the countrys initials to the name – 0x01 - Add the countrys initials and a text string to the name – 0xFF - Not specified
longNameSB	• PLMN long name spare bits. This value is ignored if longNameLen is zero. Values: – 0x01 - Bit 8 is spare and set to 0 in octet – 0x02 - Bits 7 and 8 are spare and set to 0 in octet n – 0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n – 0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n – 0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n – 0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n – 0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n – 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

8.513.2 Field Documentation

8.513.2.1 BYTE nasPLMNNameResp::longName[255]

8.513.2.2 BYTE nasPLMNNameResp::longNameCI

8.513.2.3 BYTE nasPLMNNameResp::longNameEn

8.513.2.4 BYTE nasPLMNNameResp::longNameLen

8.513.2.5 BYTE nasPLMNNameResp::longNameSB

8.513.2.6 BYTE nasPLMNNameResp::shortName[255]

8.513.2.7 BYTE nasPLMNNameResp::shortNameCI

8.513.2.8 BYTE nasPLMNNameResp::shortNameEn

8.513.2.9 BYTE nasPLMNNameResp::shortNameLen

8.513.2.10 BYTE nasPLMNNameResp::shortNameSB

8.513.2.11 BYTE nasPLMNNameResp::spn[255]

8.513.2.12 BYTE nasPLMNNameResp::spnEncoding

8.513.2.13 BYTE nasPLMNNameResp::spnLength

8.514 NASPRLPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t PRLPref

8.514.1 Field Documentation

8.514.1.1 uint16_t NASPRLPreferenceTlv::PRLPref

8.514.1.2 uint8_t NASPRLPreferenceTlv::TlvPresent

8.515 NASQmiCbkNasSwiOTAMessageInd Struct Reference

Data Fields

- NASOTAMessageTlv otaMsgTlv
- NASLteNasReleaseInfoTlv nasRelInfoTlv
- NASTimeInfoTlv timeTlv

8.515.1 Field Documentation

8.515.1.1 NASLteNasReleaseInfoTlv NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv

8.515.1.2 NASOTAMessageTlv NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv

8.515.1.3 NASTimeInfoTlv NASQmiCbkNasSwiOTAMessageInd::timeTlv

8.516 NASQmiCbkNasSystemSelPrefInd Struct Reference

Data Fields

- NASEmergencyModeTlv EMTlv
- NASModePreferenceTlv MPTlv
- NASBandPreferenceTlv BPTlv
- NASPRLPreferenceTlv PRLPTlv
- NASRoamPreferenceTlv RPTlv
- NASLTEBandPreferenceTlv LBPTlv
- NASNetSelPreferenceTlv NSPTlv
- NASServDomainPrefTlv SDPTlv
- NASGWAcqOrderPrefTlv GWAOPTlv

8.516.1 Field Documentation

8.516.1.1 NASBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::BPTlv

8.516.1.2 NASEmergencyModeTlv NASQmiCbkNasSystemSelPrefInd::EMTlv

8.516.1.3 NASGWAcqOrderPrefTlv NASQmiCbkNasSystemSelPrefInd::GWAOPTlv

8.516.1.4 NASLTEBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::LBPTlv

8.516.1.5 NASModePreferenceTlv NASQmiCbkNasSystemSelPrefInd::MPTlv

8.516.1.6 NASNetSelPreferenceTlv NASQmiCbkNasSystemSelPrefInd::NSPTlv

8.516.1.7 NASPRLPreferenceTlv NASQmiCbkNasSystemSelPrefInd::PRLPTlv

8.516.1.8 NASRoamPreferenceTlv NASQmiCbkNasSystemSelPrefInd::RPTlv

8.516.1.9 NASServDomainPrefTlv NASQmiCbkNasSystemSelPrefInd::SDPTlv

8.517 NASRoamPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t RoamPref

8.517.1 Field Documentation

8.517.1.1 uint16_t NASRoamPreferenceTlv::RoamPref

8.517.1.2 uint8_t NASRoamPreferenceTlv::TlvPresent

8.518 NASServDomainPrefTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t SrvDomainPref

8.518.1 Field Documentation

8.518.1.1 uint32_t NASServDomainPrefTlv::SrvDomainPref

8.518.1.2 uint8_t NASServDomainPrefTlv::TlvPresent

8.519 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

8.519.1 Detailed Description

This structure will hold the serving system parameters information

Parameters

registrationState	- Registration state of the mobile • 0 - QMI_NAS_NOT_REGISTERED Not registered;mobile is not currently searching for a new network to provide service • 1 - QMI_NAS_REGISTERED Registered with a network • 2 - QMI_NAS_NOT_REGISTERED_SEARCHING Not registered, but mobile is currently searching for a new network to provide service • 3 - QMI_NAS_REGISTRATION_DENIED Registration denied by the visible network • 4 - QMI_NAS_REGISTRATION_UNKNOWN Registration state is unknown
csAttachState	- Circuit Switch domain attach state of the mobile • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached

psAttachState	- Packet domain attach state of the mobile • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached
selectedNetwork	- Type of selected radio access network • 0x00 - Unknown • 0x01 - 3GPP2 network • 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) • 0x00 - None (no service) • 0x01 - cdma2000 1X • 0x02 - cdma2000 HRPD (1xEV-DO) • 0x03 - AMPS • 0x04 - GSM • 0x05 - UMTS • 0x08 - LTE
hdrPersonality	- HDR personality information (valid only for EVDO) • 0x00 - Unknown • 0x01 - HRPD • 0x02 - eHRPD

Note: None

8.519.2 Field Documentation

8.519.2.1 uint8_t NASServingSystemInfo::csAttachState

8.519.2.2 uint8_t NASServingSystemInfo::hdrPersonality

8.519.2.3 uint8_t NASServingSystemInfo::psAttachState

8.519.2.4 uint8_t NASServingSystemInfo::radioInterfaceList[255]

8.519.2.5 uint8_t NASServingSystemInfo::radioInterfaceNo

8.519.2.6 uint8_t NASServingSystemInfo::registrationState

8.519.2.7 uint8_t NASServingSystemInfo::selectedNetwork

8.520 nasSigInfo Struct Reference

Data Fields

- CDMASSInfo * pCDMASigInfo
- HDRSSInfo * pHDRSigInfo
- INT8 * pGSMSigInfo

- CDMASSInfo * pWCDMASigInfo
- LTESSInfo * pLTESigInfo
- INT8 * pRscp
- TDSCDMASigInfoExt * pTDSCDMASigInfoExt

8.520.1 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.

8.520.2 Field Documentation

8.520.2.1 CDMASSInfo* nasSigInfo::pCDMASigInfo

8.520.2.2 INT8* nasSigInfo::pGSMSigInfo

8.520.2.3 HDRSSInfo* nasSigInfo::pHDRSigInfo

8.520.2.4 LTESSInfo* nasSigInfo::pLTESigInfo

8.520.2.5 INT8* nasSigInfo::pRscp

8.520.2.6 TDSCDMASigInfoExt* nasSigInfo::pTDSCDMASigInfoExt

8.520.2.7 CDMASSInfo* nasSigInfo::pWCDMASigInfo

8.521 nasSwiGetChannelLockResp Struct Reference

Data Fields

- wcdmaUARFCN * pWcdmaUARFCN
- lteEARFCN * pLteEARFCN
- ltePCI * pLtePCI

8.521.1 Detailed Description

This structure contains the SLQSNASSwiGetChannelLock response parameters.

Parameters

pWcdmaUARFCN	[Optional] • See wcdmaUARFCN for more information
pLteEARFCN	[Optional] • See lteEARFCN for more information
pLtePCI	[Optional] • See ltePCI for more information

8.521.2 Field Documentation

8.521.2.1 lteEARFCN* nasSwiGetChannelLockResp::pLteEARFCN

8.521.2.2 ltePCI* nasSwiGetChannelLockResp::pLtePCI

8.521.2.3 wcdmaUARFCN* nasSwiGetChannelLockResp::pWcdmaUARFCN

8.522 NasSwiIndReg Struct Reference

Data Fields

- BYTE lteEsmUl
- BYTE lteEsmDl
- BYTE lteEmmUl
- BYTE lteEmmDl
- BYTE gsmUmtsUl
- BYTE gsmUmtsDl
- BYTE * pRankIndicatorInd
- BYTE * pTimer

8.522.1 Detailed Description

This structure contains the OTA message indication.

Parameters

lteEsmUl	• 0 - do not report • 1 - report LTE ESM uplink messages

lteEsmDl	• 0 - do not report • 1 - report LTE ESM downlink messages
lteEmmUl	• 0 - do not report • 1 - report LTE EMM uplink messages
lteEmmDl	• 0 - do not report • 1 - report GSM/UMTS uplink messages
gsmUmtsUl	• 0 - do not report • 1 - report GSM/UMTS uplink messages
gsmUmtsDl	• 0 - do not report • 1 - report GSM/UMTS downlink messages
pRankIndicator-Ind	• 0 - do not report • 1 - report Rank Indicator messages
pTimer	• 0 - do not report • 1 - report Timer Indicator messages

8.522.2 Field Documentation

8.522.2.1 BYTE NasSwiIndReg::gsmUmtsDl

8.522.2.2 BYTE NasSwiIndReg::gsmUmtsUl

8.522.2.3 BYTE NasSwiIndReg::lteEmmDl

8.522.2.4 BYTE NasSwiIndReg::lteEmmUl

8.522.2.5 BYTE NasSwiIndReg::lteEsmDl

8.522.2.6 BYTE NasSwiIndReg::lteEsmUl

8.522.2.7 BYTE* NasSwiIndReg::pRankIndicatorInd

8.522.2.8 BYTE* NasSwiIndReg::pTimer

8.523 nasSwiSetChannelLockReq Struct Reference

Data Fields

- wcdmaUARFCN * pWcdmaUARFCN

- lteEARFCN * pLteEARFCN
- ltePCI * pLtePCI

8.523.1 Detailed Description

This structure contains the SLQSNASSwiSetChannelLock response parameters.

Parameters

pWcdmaUARFCN	[Optional] • See wcdmaUARCFN for more information
pLteEARFCN	[Optional] • See lteEARCFN for more information
pLtePCI	[Optional] • See ltePCI for more information

8.523.2 Field Documentation

8.523.2.1 lteEARFCN* nasSwiSetChannelLockReq::pLteEARFCN

8.523.2.2 ltePCI* nasSwiSetChannelLockReq::pLtePCI

8.523.2.3 wcdmaUARFCN* nasSwiSetChannelLockReq::pWcdmaUARFCN

8.524 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

8.524.1 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. • 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. • 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. – 0x00 - Voice is not supported – 0x01 - Voice is supported
pGSMCipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pWCDMACipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pSysInfoNoChange	• System Info No Change. • Flag used to notify clients that a request to select a network ended with no change in the PLMN. – 0x01 - No change in system information

8.524.2 Field Documentation

8.524.2.1 AddCDMASysInfo* nasSysInfo::pAddCDMASysInfo

8.524.2.2 AddSysInfo* nasSysInfo::pAddGSMSysInfo

8.524.2.3 WORD* nasSysInfo::pAddHDRSysInfo

8.524.2.4 WORD* nasSysInfo::pAddLTESysInfo

8.524.2.5 AddSysInfo* nasSysInfo::pAddWCDMASysInfo

8.524.2.6 SrvStatusInfo* nasSysInfo::pCDMASrvStatusInfo

8.524.2.7 CDMASysInfo* nasSysInfo::pCDMASysInfo

8.524.2.8 CallBarringSysInfo* nasSysInfo::pGSMCallBarringSysInfo

8.524.2.9 BYTE* nasSysInfo::pGSMCipherDomainSysInfo

8.524.2.10 GSMSrvStatusInfo* nasSysInfo::pGSMSrvStatusInfo

8.524.2.11 GSMSysInfo* nasSysInfo::pGSMSysInfo

8.524.2.12 SrvStatusInfo* nasSysInfo::pHDRSrvStatusInfo

8.524.2.13 HDRSysInfo* nasSysInfo::pHDRSysInfo

8.524.2.14 GSMSrvStatusInfo* nasSysInfo::pLTESrvStatusInfo

8.524.2.15 LTESysInfo* nasSysInfo::pLTESysInfo

8.524.2.16 BYTE* nasSysInfo::pLTEVoiceSupportSysInfo

8.524.2.17 BYTE* nasSysInfo::pSysInfoNoChange

8.524.2.18 CallBarringSysInfo* nasSysInfo::pWCDMACallBarringSysInfo

8.524.2.19 BYTE* nasSysInfo::pWCDMACipherDomainSysInfo

8.524.2.20 GSMSrvStatusInfo* nasSysInfo::pWCDMASrvStatusInfo

8.524.2.21 WCDMASysInfo* nasSysInfo::pWCDMASysInfo

8.525 NASTimeInfoTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t time

8.525.1 Field Documentation

8.525.1.1 uint64_t NASTimeInfoTlv::time

8.525.1.2 uint8_t NASTimeInfoTlv::TlvPresent

8.526 nasTimers Struct Reference

Data Fields

- char t3396_apn [101]
- BYTE t3396_plmn_id [3]
- ULONG t3396_val

8.526.1 Detailed Description

This structure contains Network Timer

Parameters

t3396_apn	• APN name

t3396_plmn_id	• PLMN ID
t3396_val	• T3396 timer value

8.526.2 Field Documentation

8.526.2.1 char nasTimers::t3396_apn[101]

8.526.2.2 BYTE nasTimers::t3396_plmn_id[3]

8.526.2.3 ULONG nasTimers::t3396_val

8.527 netSelectionPref Struct Reference

Data Fields

- BYTE netReg
- WORD mcc
- WORD mnc

8.527.1 Detailed Description

Contain the network selection preference.

Parameters

netReg	• specifies one of the following actions: – 0x00 - Automatic registration * Device registers according to its provisioning; mcc and mnc fields are ignored – 0x01 - Manual Registration * 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

8.527.2 Field Documentation

8.527.2.1 WORD netSelectionPref::mcc

8.527.2.2 WORD netSelectionPref::mnc

8.527.2.3 BYTE netSelectionPref::netReg

8.528 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

8.528.1 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

8.528.2 Field Documentation

8.528.2.1 ULONGLONG NetStats::rx_bytes

8.528.2.2 ULONG NetStats::rx_errors

8.528.2.3 ULONG NetStats::rx_overflows

8.528.2.4 ULONG NetStats::rx_packets

8.528.2.5 ULONGLONG NetStats::tx_bytes

8.528.2.6 ULONG NetStats::tx_errors

8.528.2.7 ULONG NetStats::tx_overflows

8.528.2.8 ULONG NetStats::tx_packets

8.529 NetworkDebugResp Struct Reference

Data Fields

- BYTE * pObjectVer
- NetworkStat1x * pNetworkStat1x
- NetworkStatEVDO * pNetworkStatEVDO
- DeviceConfigDetail * pDeviceConfigDetail
- DataStatusDetail * pDataStatusDetail

8.529.1 Detailed Description

This structure contains information about the SLQSSwiNetworkDebug response parameters.

Parameters

pObjectVer	• Object's version number for the host to handle – 0xFF - NA – 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.

8.529.2 Field Documentation

8.529.2.1 DataStatusDetail* NetworkDebugResp::pDataStatusDetail

8.529.2.2 DeviceConfigDetail* NetworkDebugResp::pDeviceConfigDetail

8.529.2.3 NetworkStat1x* NetworkDebugResp::pNetworkStat1x

8.529.2.4 NetworkStatEVDO* NetworkDebugResp::pNetworkStatEVDO

8.529.2.5 BYTE* NetworkDebugResp::pObjectVer

8.530 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

8.530.1 Detailed Description

This structure contains 1x network status details

Parameters

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

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

8.530.2 Field Documentation

8.530.2.1 BYTE NetworkStat1x::ActSetCnt

8.530.2.2 BYTE NetworkStat1x::NeighborSetCnt

8.530.2.3 ActPilotPNElement* NetworkStat1x::pActPilotPNElements

8.530.2.4 WORD* NetworkStat1x::pNeighborSetPilotPN

8.530.2.5 WORD NetworkStat1x::RX_EC_IO

8.530.2.6 ULONG NetworkStat1x::RX_PWR

8.530.2.7 WORD NetworkStat1x::SO

8.530.2.8 BYTE NetworkStat1x::State

8.530.2.9 ULONG NetworkStat1x::TX_PWR

8.531 NetworkStatEVDO Struct Reference

Data Fields

- BYTE State
- BYTE MACIndex
- BYTE SectorIDLen
- WORD * pSectorID
- WORD RX_PWR
- WORD PER
- WORD PilotEnergy
- BYTE SNR

8.531.1 Detailed Description

This structure contains EVDO network status details

Parameters

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

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

8.531.2 Field Documentation

8.531.2.1 BYTE NetworkStatEVDO::MACIndex

8.531.2.2 WORD NetworkStatEVDO::PER

8.531.2.3 WORD NetworkStatEVDO::PilotEnergy

8.531.2.4 WORD* NetworkStatEVDO::pSectorID

8.531.2.5 WORD NetworkStatEVDO::RX_PWR

8.531.2.6 BYTE NetworkStatEVDO::SectorIDLen

8.531.2.7 BYTE NetworkStatEVDO::SNR

8.531.2.8 BYTE NetworkStatEVDO::State

8.532 newMTMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSMTMessageInfo MTMessageInfo

8.532.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
MTMessageInfo	• MT Message • See sMSMTMessageInfo for more information

8.532.2 Field Documentation

8.532.2.1 sMSMTMessageInfo newMTMessageTlv::MTMessageInfo

8.532.2.2 uint8_t newMTMessageTlv::TlvPresent

8.533 newPwdData Struct Reference

Data Fields

- BYTE newPwd [4]
- BYTE newPwdAgain [4]

8.533.1 Detailed Description

This structure contains New Password Data.

Parameters

newPwd[PASSWORD_LENGTH]	• New password. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.
newPwdAgain[PASSWORD_LENGTH]	• New password again. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.

8.533.2 Field Documentation

8.533.2.1 BYTE newPwdData::newPwd[4]

8.533.2.2 BYTE newPwdData::newPwdAgain[4]

8.534 nmrCellInfo Struct Reference

Data Fields

- ULONG nmrCellID
- BYTE nmrPlmn [3]
- WORD nmrLac

- WORD nmrArfcn
- BYTE nmrBsic
- WORD nmrRxLev

8.534.1 Detailed Description

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

Parameters

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

8.534.2 Field Documentation

8.534.2.1 WORD nmrCellInfo::nmrArfcn

8.534.2.2 BYTE nmrCellInfo::nmrBsic

8.534.2.3 ULONG nmrCellInfo::nmrCellID

8.534.2.4 WORD nmrCellInfo::nmrLac

8.534.2.5 BYTE nmrCellInfo::nmrPlmn[3]

8.534.2.6 WORD nmrCellInfo::nmrRxLev

8.535 NSSAudioCtrl Struct Reference

Data Fields

- BYTE upLink
- BYTE downLink

8.535.1 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].

8.535.2 Field Documentation

8.535.2.1 BYTE NSSAudioCtrl::downLink

8.535.2.2 BYTE NSSAudioCtrl::upLink

8.536 NWProfile Struct Reference

Data Fields

- WORD tech
- BYTE * pProfSz
- WORD * pProfValues

8.536.1 Detailed Description

This structure contains Network supported QoS profile

Parameters

tech	Technology on which the network supported QoS profiles are being returned: • CDMA – 0x8001
exponent	

8.536.2 Field Documentation

8.536.2.1 BYTE* NWProfile::pProfSz

8.536.2.2 WORD* NWProfile::pProfValues

8.536.2.3 WORD NWProfile::tech

8.537 omaDmConfigTlv Struct Reference

Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- USHORT alertmsglength
- BYTE alertmsg [256]

8.537.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

Parameters

state	• 0x01 - OMA-DM Read Request • 0x02 - OMA-DM Change Request • 0x03 - OMA-DM Config Complete
user_input_req	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject
user_input_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)

8.537.2 Field Documentation

8.537.2.1 BYTE omaDmConfigTlv::alertmsg[256]

8.537.2.2 USHORT omaDmConfigTlv::alertmsglength

8.537.2.3 BYTE omaDmConfigTlv::state

8.537.2.4 BYTE omaDmConfigTlv::userInputReq

8.537.2.5 USHORT omaDmConfigTlv::userInputTimeout

8.538 omaDmConfigTlvExt Struct Reference

Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- USHORT alertmsglength
- BYTE alertmsg [256]

8.538.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

Parameters

state	• 1 - reserved • 2 - reserved • 3 - reserved • 4 - CI DC Success • 5 - CI DC Failure • 6 - User/device initiated PRL update success. • 7 - User/device initiated PRL update failure. • 8 - HFA DC session start • 9 - HFA DC sucess. • 10 - HFA is cancelled. • 11 - HFA retry. UI Screen 13[1] with 0 percent progress bar should be shown. • 12 - HFA fail after 5 retries. UI Screen 2[1] should be displayed. • 13 - HFA retry down counter. Used to update the process bar of UI Screen 13[1]. • 14 - HFA PRL session start, UI screen 4[1] should be displayed. • 15 - HFA PRL update success. • 16 - Device is launching a NI session. UI Screen 1[1] should be displayed. • 17 - An empty session. UI Screen 2[1] should be displayed. • 18 - No network coverage. • 19 - HFA is not enabled. • 20 - CI DC Start, UI Screen 1[1] should be displayed. • 21 - CI PRL start, UI screen 4[1] should be displayed. • 22 - HFA PRL updates fail. • 23 - Device reboot. • 24 - CI DC is cancelled. • 25 - User/device initiated PRL update is cancelled. • 26 - NI session is cancelled. • 27 - Current NI session is not enabled. • 28 - NI DC success. • 29 - NI DC Fail. • 30 - NI PRL success • 31 - NI PRL fail. • 32 - Reserved • 33 - NI fumo fail • 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) • This string is printed by host

8.538.2 Field Documentation

8.538.2.1 BYTE omaDmConfigTlvExt::alertmsg[256]

8.538.2.2 USHORT omaDmConfigTlvExt::alertmsglength

8.538.2.3 BYTE omaDmConfigTlvExt::state

8.538.2.4 BYTE omaDmConfigTlvExt::userInputReq

8.538.2.5 USHORT omaDmConfigTlvExt::userInputTimeout

8.539 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

8.539.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

Parameters

state	• 0x01 - No Firmware available • 0x02 - Query Firmware Download • 0x03 - Firmware Downloading • 0x04 - Firmware downloaded • 0x05 - Query Firmware Update • 0x06 - Firmware updating • 0x07 - Firmware updated
user_input_req	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject
user_input_timeout	• Timeout for user input in minutes. A value of 0 means no time-out
fw_dload_size	• The size (in bytes) of the firmware update package
fw_dload_complete	• The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment.
update_complete_status	• See table below.
severity	• 0x01 - Mandatory • 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 • 0x01 - Network initiated

8.539.2 Field Documentation

8.539.2.1 BYTE omaDmFotaTlv::description[256]

8.539.2.2 USHORT omaDmFotaTlv::descriptionlength

8.539.2.3 ULONG omaDmFotaTlv::fwdloadsize

8.539.2.4 ULONG omaDmFotaTlv::fwloadComplete

8.539.2.5 USHORT omaDmFotaTlv::namelength

8.539.2.6 BYTE omaDmFotaTlv::package_name[256]

8.539.2.7 BYTE omaDmFotaTlv::sessionType

8.539.2.8 BYTE omaDmFotaTlv::severity

8.539.2.9 BYTE omaDmFotaTlv::state

8.539.2.10 USHORT omaDmFotaTlv::updateCompleteStatus

8.539.2.11 BYTE omaDmFotaTlv::userInputReq

8.539.2.12 USHORT omaDmFotaTlv::userInputTimeout

8.539.2.13 BYTE omaDmFotaTlv::version[256]

8.539.2.14 USHORT omaDmFotaTlv::versionlength

8.540 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]

8.540.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

Parameters

state	• 0x01 - No Firmware available • 0x02 - reserved • 0x03 - Update progress bar, UI screen 7[1] should be displayed • 0x04 - reserved • 0x05 - FUMO image download success, UI screen 8[1] should be displayed • 0x06 - reserved • 0x07 - FUMO image installation result, used to return error code. • 0x08 - FUMO session start • 0x09 - UI Screen 7[1] with 0 percent bar should be displayed • 0x0A - FUMO image installation is cancelled by user. • 0x0B - FUMO session fail • 0x0C - Device is sending a report to OMA Server. • 0x0D - Report to server success. • 0x0E - Report to server fails. • 0x0F - reserved • 0x10 - FUMO session is cancelled before image download success. • 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 – 200 image install success • 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)

8.540.2 Field Documentation

8.540.2.1 BYTE omaDmFotaTlvExt::description[256]

8.540.2.2 USHORT omaDmFotaTlvExt::descriptionlength

8.540.2.3 USHORT omaDmFotaTlvExt::fumoResultCode

8.540.2.4 USHORT omaDmFotaTlvExt::namelength

8.540.2.5 BYTE omaDmFotaTlvExt::package_name[256]

8.540.2.6 ULONG omaDmFotaTlvExt::packageSize

8.540.2.7 ULONG omaDmFotaTlvExt::receivedBytes

8.540.2.8 BYTE omaDmFotaTlvExt::reserved

8.540.2.9 BYTE omaDmFotaTlvExt::state

8.540.2.10 USHORT omaDmFotaTlvExt::userInputTimeout

8.540.2.11 BYTE omaDmFotaTlvExt::version[256]

8.540.2.12 USHORT omaDmFotaTlvExt::versionlength

8.541 omaDmNotificationsTlv Struct Reference

Data Fields

- BYTE notification
- USHORT sessionStatus

8.541.1 Field Documentation

8.541.1.1 BYTE omaDmNotificationsTlv::notification

8.541.1.2 USHORT omaDmNotificationsTlv::sessionStatus

8.542 operatorNameString Struct Reference

Data Fields

- BYTE PLMNName [255]

8.542.1 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.

8.542.2 Field Documentation

8.542.2.1 BYTE operatorNameString::PLMNName[255]

8.543 OperatorPLMNData Struct Reference

Data Fields

- BYTE mcc [3]
- BYTE mnc [3]
- WORD lac1
- WORD lac2
- BYTE PLMNRecID

8.543.1 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.

8.543.2 Field Documentation

8.543.2.1 WORD OperatorPLMNData::lac1

8.543.2.2 WORD OperatorPLMNData::lac2

8.543.2.3 BYTE OperatorPLMNData::mcc[3]

8.543.2.4 BYTE OperatorPLMNData::mnc[3]

8.543.2.5 BYTE OperatorPLMNData::PLMNRecID

8.544 operatorPLMNList Struct Reference

Data Fields

- WORD numInstance
- OperatorPLMNData PLMNData [255]

8.544.1 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).

8.544.2 Field Documentation

8.544.2.1 WORD operatorPLMNList::numInstance

8.544.2.2 OperatorPLMNData operatorPLMNList::PLMNData[255]

8.545 pack_dms_ActivateAutomatic_t Struct Reference

Data Fields

- uint8_t actCode [81]

8.545.1 Detailed Description

This structure contains UIM activate automation information

Parameters

spc[IN]	• Service programming code in ASCII format (digits 0 to 9 only)

8.545.2 Field Documentation

8.545.2.1 uint8_t pack_dms_ActivateAutomatic_t::actCode[81]

8.546 pack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

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

8.546.1 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
Tlvresult	• Pack Result

8.546.2 Field Documentation

8.546.2.1 uint8_t pack_dms_GetCustFeaturesV2_t::cust_id[64+1]

8.546.2.2 uint8_t pack_dms_GetCustFeaturesV2_t::list_type

8.546.2.3 uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult

8.547 pack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- uint8_t spc [6]

8.547.1 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)

8.547.2 Field Documentation

8.547.2.1 uint8_t pack_dms_ResetToFactoryDefaults_t::spc[6]

8.548 pack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- uint8_t activationState

8.548.1 Detailed Description

This structure is used to store Service Activation Status parameter .

Parameters

activationState	• Service activation state. • Values – 0 - Do not report – 1 - Report activation state changes

8.548.2 Field Documentation

8.548.2.1 uint8_t pack_dms_SetActivationStatusCallback_t::activationState

8.549 pack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t crashAction

8.549.1 Detailed Description

Modem action in case of a crash

Parameters

crashAction	• 0 - USB Memory Download. Modem will reset after a crash and will stay in USB download mode with only DM port enumerated. • 1 - Reset. Modem will reset and come back in ONLINE mode. Minimal crash data will be available and can be extracted with at!gcdump? AT command • 2 - No Action

8.549.2 Field Documentation

8.549.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

8.550 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

8.550.1 Field Documentation

8.550.1.1 uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled

8.550.1.2 uint8_t pack_dms_SetCustFeature_t::DisableIMSI

8.550.1.3 uint32_t pack_dms_SetCustFeature_t::GpsEnable

8.550.1.4 uint8_t pack_dms_SetCustFeature_t::GPSLPM

8.550.1.5 uint8_t pack_dms_SetCustFeature_t::GPSSel

8.550.1.6 uint16_t pack_dms_SetCustFeature_t::IPFamSupport

8.550.1.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

8.550.1.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

8.550.1.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

8.551 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

8.551.1 Detailed Description

This structure contains customization settings set to modem pack

Parameters

cust_id	• Customization ID (Maximum 64 bytes)
value_length	• length of cust_value field

cust_value	• Customization Setting Value (Maximum 8 bytes)
Tlvresult	• Pack Result

8.551.2 Field Documentation

8.551.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

8.551.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

8.551.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

8.551.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

8.552 pack_dms_SetEventReport_t Struct Reference

Data Fields

- uint8_t mode

8.552.1 Field Documentation

8.552.1.1 uint8_t pack_dms_SetEventReport_t::mode

8.553 pack_dms_SetPower_t Struct Reference

Data Fields

- uint32_t mode
- uint16_t Tlvresult

8.553.1 Field Documentation

8.553.1.1 uint32_t pack_dms_SetPower_t::mode

8.553.1.2 uint16_t pack_dms_SetPower_t::Tlvresult

8.554 pack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint8_t USBComp
- uint16_t Tlvresult

8.554.1 Field Documentation

8.554.1.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

8.554.1.2 uint8_t pack_dms_SetUSBComp_t::USBComp

8.555 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint8_t resetInfoInd

8.555.1 Detailed Description

Parameters

resetInfoInd[IN]	• Values – 0 - Disable – 1 - Enable

8.555.2 Field Documentation

8.555.2.1 uint8_t pack_dms_SLQSDmsSwiIndicationRegister_t::resetInfoInd

8.556 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference

Data Fields

- uint8_t clear

8.556.1 Detailed Description

This structure contains crashInfoParams

Parameters

clear	• 0 - do not clear crash data after response • 1 - clear crash data after response

8.556.2 Field Documentation

8.556.2.1 uint8_t pack_dms_SLQSSwiGetCrashInfo_t::clear

8.557 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint8_t ∗ pDestSMSNum
- uint8_t ∗ pDestSMSContent

8.557.1 Detailed Description

Parameters

pDestSMSNum[IN]	• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars. • Optional parameter.
pDestSMSContent[IN]	• SMS COntent as a string of 8 bit ASCII text characters Max 160 chars. • Optional parameter.

8.557.2 Field Documentation

8.557.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

8.557.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

8.558 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

Data Fields

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

8.558.1 Detailed Description

This structure contains SWI set host device info unpack information

Parameters

manString[IN]	• optional parameter, host device manufacture
modelString[IN]	• optional parameter, host device model
swVerString[IN]	• optional parameter, host device software version
plasmaIDString[IN]	• optional parameter, host device plasma ID

8.558.2 Field Documentation

8.558.2.1 char pack_dms_SLQSSwiSetHostDevInfo_t::manString[255]

8.558.2.2 char pack_dms_SLQSSwiSetHostDevInfo_t::modelString[255]

8.558.2.3 char pack_dms_SLQSSwiSetHostDevInfo_t::plasmaIDString[255]

8.558.2.4 char pack_dms_SLQSSwiSetHostDevInfo_t::swVerString[255]

8.559 pack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

- char nameString [255]
- char versionString [255]

8.559.1 Detailed Description

This structure contains SWI set host OS info pack information

Parameters

nameString[IN]	• optional parameter, host device manufacture
VersionString[IN]	• optional parameter, host device model

8.559.2 Field Documentation

8.559.2.1 char pack_dms_SLQSSwiSetOSInfo_t::nameString[255]

8.559.2.2 char pack_dms_SLQSSwiSetOSInfo_t::versionString[255]

8.560 pack_dms_UIMChangePIN_t Struct Reference

Data Fields

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

8.560.1 Detailed Description

This structure contains UIM Unblock PIN Information

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
oldValue[IN]	• Old PIN value of PIN to change

newValue[IN]	• New PIN value of PIN to change

8.560.2 Field Documentation

8.560.2.1 uint8_t pack_dms_UIMChangePIN_t::id

8.560.2.2 uint8_t pack_dms_UIMChangePIN_t::newValue[255]

8.560.2.3 uint8_t pack_dms_UIMChangePIN_t::oldValue[255]

8.561 pack_dms_UIMGetControlKeyStatus_t Struct Reference

Data Fields

- uint8_t facility

8.561.1 Detailed Description

This structure contains UIM get control key status information

Parameters

facility[IN]	• MT or network facility – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)

8.561.2 Field Documentation

8.561.2.1 uint8_t pack_dms_UIMGetControlKeyStatus_t::facility

8.562 pack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.562.1 Detailed Description

This structure contains UIM Get ICCID pack

Parameters

Tlvresult	• Pack result.

8.562.2 Field Documentation

8.562.2.1 uint16_t pack_dms_UIMGetICCID_t::TlvResult

8.563 pack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

- uint8_t facility
- uint8_t facilityState
- uint8_t facilityCk [8]

8.563.1 Detailed Description

This structure contains UIM Set control key protection information

Parameters

facility[IN]	• MT or network facility – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)
facilityState[IN]	• UIM facility state
facilityCk[IN]	• Facility depersonalization control key, string in ASCII text (maximum 8 bytes)

8.563.2 Field Documentation

8.563.2.1 uint8_t pack_dms_UIMSetControlKeyProtection_t::facility

8.563.2.2 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityCk[8]

8.563.2.3 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityState

8.564 pack_dms_UIMSetPINProtection_t Struct Reference

Data Fields

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

8.564.1 Detailed Description

Set activation status pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
in	*reqArg*	request parameter This structure contains PIN Protection Information
	id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
	bEnable[IN]	• Enable/disable PIN protection, 0 = Disable
	pValue[IN]	• PIN value of the PIN to be enabled/disabled

8.564.2 Field Documentation

8.564.2.1 uint8_t pack_dms_UIMSetPINProtection_t::bEnable

8.564.2.2 uint8_t pack_dms_UIMSetPINProtection_t::id

8.564.2.3 uint8_t pack_dms_UIMSetPINProtection_t::value[255]

8.565 pack_dms_UIMUnblockControlKey_t Struct Reference

Data Fields

- uint8_t facility
- uint8_t facilityCk [8]

8.565.1 Detailed Description

This structure contains UIM unblock Control Key information

Parameters

facility[IN]	• MT or network facility – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)
facilityCk[IN]	• Facility depersonalization control key, string in ASCII text (maximum 8 bytes)

8.565.2 Field Documentation

8.565.2.1 uint8_t pack_dms_UIMUnblockControlKey_t::facility

8.565.2.2 uint8_t pack_dms_UIMUnblockControlKey_t::facilityCk[8]

8.566 pack_dms_UIMUnblockPIN_t Struct Reference

Data Fields

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

8.566.1 Detailed Description

This structure contains UIM Unblock PIN Information

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
pukValue[IN]	• PUK value of PIN to be unblocked
newPin[IN]	• New PIN value for the PIN to be unblocked

8.566.2 Field Documentation

8.566.2.1 uint8_t pack_dms_UIMUnblockPIN_t::id

8.566.2.2 uint8_t pack_dms_UIMUnblockPIN_t::newPin[255]

8.566.2.3 uint8_t pack_dms_UIMUnblockPIN_t::pukValue[255]

8.567 pack_dms_UIMVerifyPIN_t Struct Reference

Data Fields

- uint8_t id
- uint8_t value [255]

8.567.1 Detailed Description

This structure contains PIN Value Information

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
pValue[IN]	• PIN value of the PIN to be enabled/disabled

8.567.2 Field Documentation

8.567.2.1 uint8_t pack_dms_UIMVerifyPIN_t::id

8.567.2.2 uint8_t pack_dms_UIMVerifyPIN_t::value[255]

8.568 pack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.568.1 Detailed Description

This structure contains the Get Image Preference information pack

Parameters

Tlvresult	• Pack result

8.568.2 Field Documentation

8.568.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

8.569 pack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.569.1 Detailed Description

This structure contains the Get Stored Images pack

Parameters

Tlvresult	• Pack result

8.569.2 Field Documentation

8.569.2.1 uint16_t pack_fms_GetStoredImages_t::Tlvresult

8.570 pack_fms_SetImagesPreference_t Struct Reference

Data Fields

- uint32_t imageListSize
- FMSPrefImageList * pImageList
- uint32_t bForceDownload
- uint8_t modemindex
- uint16_t Tlvresult

8.570.1 Detailed Description

This structure contains the Set Images Preference pack

Parameters

imageListSize	• Image List Size
pImageList	• Image List • See FMSPrefImageList
bForceDownload	• 0 - Not Force Donwload. • 1 - Focrce Download.
modemindex	• Modem Index.
Tlvresult	• Unpack result

8.570.2 Field Documentation

8.570.2.1 uint32_t pack_fms_SetImagesPreference_t::bForceDownload

8.570.2.2 uint32_t pack_fms_SetImagesPreference_t::imageListSize

8.570.2.3 uint8_t pack_fms_SetImagesPreference_t::modemindex

8.570.2.4 FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList

8.570.2.5 uint16_t pack_fms_SetImagesPreference_t::Tlvresult

8.571 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

8.571.1 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.

8.571.2 Field Documentation

8.571.2.1 loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo

8.571.2.2 loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb

8.571.2.3 loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo

8.571.2.4 loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData

8.571.2.5 loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo

8.571.2.6 uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult

8.572 pack_loc_EventRegister_t Struct Reference

Data Fields

- uint64_t eventRegister
- uint16_t Tlvresult

8.572.1 Detailed Description

This structure contains the Parameter for RegisterEvents

Parameters

eventRegister	• Specifies the events that the control point is interested in receiving. -Values – 0x00000001 - to receive position report event indications – 0x00000002 - to receive satellite report event indications.These reports are sent at a 1 Hz rate. – 0x00000004 - to receive NMEA reports for position and satellites in view.The report is at a 1 Hz rate. – 0x00000008 - to receive NI Notify/Verify request event indications – 0x00000010 - to receive time injection request event indications. – 0x00000020 - to receive predicted orbits request event indications. – 0x00000040 - to receive position injection request event indications. – 0x00000080 - to receive engine state report event indications. – 0x00000100 - to receive fix session status report event indications. – 0x00000200 - to receive Wi-Fi position request event indications. – 0x00000400 - to receive notifications from the location engine indicating its readiness to accept data from the sensors(accelerometer, gyroscope, etc.). – 0x00000800 - to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock. – 0x00001000 - to receive Stationary Position Indicator (SPI) streaming report indications. – 0x00002000 - to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server. – 0x00004000 - to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited. – 0x00008000 - to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, e.g., if GPS is turned off or if the network is unavailable. – 0x00010000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence. – 0x00020000 - to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports. – 0x00040000 - to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data. – 0x00080000 - to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session. – 0x00100000 - to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing. – 0x00200000 - to receive Wi-Fi Access Point (AP) data inject request event indications. – 0x00400000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification. – 0x00800000 - to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.). – 0x01000000 - to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). – 0x02000000 - to receive satellite position reports as polynomials.Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_-CONSTELL_REPORT_CONFIG.

Note

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

Parameters

TlvResult	• Pack result.

8.572.2 Field Documentation

8.572.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

8.572.2.2 uint16_t pack_loc_EventRegister_t::TlvResult

8.573 pack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint32_t extPowerState
- uint16_t TlvResult

8.573.1 Detailed Description

This structure contains the Parameter External Power Source State pack.

Parameters

extPowerState	• Specifies the Power state; injected by the control point. • Values – 0 - Device is not connected to an external power source – 1 - Device is connected to an external power source – 2 - Unknown external power state
TlvResult	• Pack result.

8.573.2 Field Documentation

8.573.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

8.573.2.2 uint16_t pack_loc_SetExtPowerState_t::TlvResult

8.574 pack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint32_t mode
- uint16_t TlvResult

8.574.1 Detailed Description

This structure contains Set Operation Mode pack

Parameters

mode	• 0 - Default Mode.
TlvResult	• Pack result.

8.574.2 Field Documentation

8.574.2.1 uint32_t pack_loc_SetOperationMode_t::mode

8.574.2.2 uint16_t pack_loc_SetOperationMode_t::TlvResult

8.575 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint32_t xid
- uint16_t TlvResult

8.575.1 Detailed Description

This structure contains Set Operation Mode pack

Parameters

xid	• Identifies the transaction. • The transaction ID is returned in the Get Best Available Position indication.
TlvResult	• Pack result.

8.575.2 Field Documentation

8.575.2.1 uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::TlvResult

8.575.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

8.576 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

8.576.1 Detailed Description

This structure contains LOC Inject Position parameters Please check has_<Param_Name> field for presence of optional parameters

Parameters

latitude	• Optional parameter • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude • Note - This field must be specified together with pLongitude and pHorUncCircular.
longitude	• Optional parameter • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude • Note - This field must be specified together with pLatitude and pHorUncCircular.

horUncCircular	• Optional parameter • Horizontal position uncertainty. • Units - Meters • Note - This field must be specified together with pLatitude and pLongitude.
horConfidence	• Optional parameter • Horizontal confidence. • Units - Percent • Values – Valid Values - 1 to 99 – Invalid Values - 0, 101 to 255 – If 100 is received, reinterpret to 99 • 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

- 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

altitudeWrt-Ellipsoid	• Optional parameter • Altitude With Respect to Ellipsoid. • Units - Meters • Values – Positive - height – Negative = depth
altitudeWrt-MeanSeaLevel	• Optional parameter • Altitude With Respect to Sea Level. • Units - Meters
vertUnc	• Optional parameter • Vertical uncertainty. • Units - Meters • Note - This is mandatory if either pAltitudeWrtEllipsoid or pAltitudeWrtMeanSeaLevel is specified.

vertConfidence	• Optional parameter • Vertical confidence. • Units - Percentage • Values – Valid Values - 0 to 99 – Invalid Values - 0, 100-256 – If 100 is received, reinterpret to 99 • Note - This field must be specified together with the vertical uncertainty. If not specified, the default value will be 50.
vertReliability	• 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

altitudeSrcInfo	• Optional parameter

- Pointer to struct altitudeSrcInfo. See altitudeSrcInfo for more information

Parameters

timestampUtc	• Optional parameter • UTC timestamp • Units - Milliseconds since Jan. 1, 1970
timestampAge	• Optional parameter • Position age, which is an estimate of how long ago this fix was made. • Units - Milliseconds

positionSrc	• Optional parameter • Source from which this position was obtained • Valid values – 0 - Position source is GNSS – 1 - Position source is Cell ID – 2 - Position source is Enhanced Cell ID – 3 - Position source is Wi-Fi – 4 - Position source is Terrestrial – 5 - Position source is GNSS Terrestrial Hybrid – 6 - Other sources • 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. – If both altitude and altitude source are specified, the engine assumes that only latitude and longitude were obtained using the specified position source.
rawHorUnc-Circular	• Optional parameter • Horizontal position uncertainty (circular) without any optimization. • Units - Meters
rawHor-Confidence	• Optional parameter • Horizontal confidence associated with raw horizontal uncertainty • Units: Percent • Values – Valid values - 1 to 99 – Invalid values - 0, 101 to 255 – If 100 is received, reinterpret to 99 • Note - This field must be specified together with raw horizontal uncertainty. If not specified when rawHorUncCircular is set, the default value is 50.

8.576.2 Field Documentation

8.576.2.1 altSrcInfo_t pack_loc_SLQSLOCInjectPosition_t::altitudeSrcInfo

8.576.2.2 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtEllipsoid

8.576.2.3 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtMeanSeaLevel

8.576.2.4 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeSrcInfo

8.576.2.5 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtEllipsoid

8.576.2.6 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtMeanSeaLevel

8.576.2.7 int pack_loc_SLQSLOCInjectPosition_t::has_horConfidence

8.576.2.8 int pack_loc_SLQSLOCInjectPosition_t::has_horReliability

8.576.2.9 int pack_loc_SLQSLOCInjectPosition_t::has_horUncCircular

8.576.2.10 int pack_loc_SLQSLOCInjectPosition_t::has_latitude

8.576.2.11 int pack_loc_SLQSLOCInjectPosition_t::has_longitude

8.576.2.12 int pack_loc_SLQSLOCInjectPosition_t::has_positionSrc

8.576.2.13 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorConfidence

8.576.2.14 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorUncCircular

8.576.2.15 int pack_loc_SLQSLOCInjectPosition_t::has_timestampAge

8.576.2.16 int pack_loc_SLQSLOCInjectPosition_t::has_timestampUtc

8.576.2.17 int pack_loc_SLQSLOCInjectPosition_t::has_vertConfidence

8.576.2.18 int pack_loc_SLQSLOCInjectPosition_t::has_vertRelicability

8.576.2.19 int pack_loc_SLQSLOCInjectPosition_t::has_vertUnc

8.576.2.20 uint8_t pack_loc_SLQSLOCInjectPosition_t::horConfidence

8.576.2.21 uint32_t pack_loc_SLQSLOCInjectPosition_t::horReliability

8.576.2.22 float pack_loc_SLQSLOCInjectPosition_t::horUncCircular

8.576.2.23 double pack_loc_SLQSLOCInjectPosition_t::latitude

8.576.2.24 double pack_loc_SLQSLOCInjectPosition_t::longitude

8.576.2.25 uint32_t pack_loc_SLQSLOCInjectPosition_t::positionSrc

8.576.2.26 uint8_t pack_loc_SLQSLOCInjectPosition_t::rawHorConfidence

8.576.2.27 float pack_loc_SLQSLOCInjectPosition_t::rawHorUncCircular

8.576.2.28 uint32_t pack_loc_SLQSLOCInjectPosition_t::timestampAge

8.576.2.29 uint64_t pack_loc_SLQSLOCInjectPosition_t::timestampUtc

8.576.2.30 uint8_t pack_loc_SLQSLOCInjectPosition_t::vertConfidence

8.576.2.31 uint32_t pack_loc_SLQSLOCInjectPosition_t::vertReliability

8.576.2.32 float pack_loc_SLQSLOCInjectPosition_t::vertUnc

8.577 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

8.577.1 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) • 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	• Optional parameter • Pointer to struct sensorData. See sensorData for more information
gyroData	• Optional parameter • Pointer to struct sensorData. See sensorData for more information
acceleroTimeSrc	• 3-Axis Accelerometer Data Time Source (Optional parameter) • The location service uses this field to identify the time reference used in the accelerometer data time stamps. • If not specified, the location service assumes that the time source for the accelerometer data is unknown. • Valid values – 0 - Sensor time source is unspecified – 1 - Time source is common between the sensors and the location engine
gyroTimeSrc	• 3-Axis Gyroscope Data Time Source (Optional) • The location service uses this field to identify the time reference used in the gyroscope data time stamps. • If not specified, the location service assumes that the time source for the gyroscope data is unknown. • Valid values – 0 - Sensor time source is unspecified – 1 - Time source is common between the sensors and the location engine
accelTemp	• Optional parameter • Pointer to struct tempratureData. See tempratureData for more information

gyroTemp	• Optional parameter • Pointer to struct temprature Data. See temprature Data for more information

8.577.2 Field Documentation

8.577.2.1 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData

8.577.2.2 uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc

8.577.2.3 tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp

8.577.2.4 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::gyroData

8.577.2.5 tempData_t pack_loc_SLQSLOCInjectSensorData_t::gyroTemp

8.577.2.6 uint32_t pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc

8.577.2.7 int pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc

8.577.2.8 int pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp

8.577.2.9 int pack_loc_SLQSLOCInjectSensorData_t::has_accleroData

8.577.2.10 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroData

8.577.2.11 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp

8.577.2.12 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc

8.577.2.13 int pack_loc_SLQSLOCInjectSensorData_t::has_opaqueId

8.577.2.14 uint32_t pack_loc_SLQSLOCInjectSensorData_t::opaqueId

8.578 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference

Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec

8.578.1 Detailed Description

Parameters

timeMsec[IN]	• The UTC time since Jan. 1, 1970
timeUncMsec[IN]	• The time Uncertainty

8.578.2 Field Documentation

8.578.2.1 uint64_t pack_loc_SLQSLOCInjectUTCTime_t::timeMsec

8.578.2.2 uint32_t pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec

8.579 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference

Data Fields

- uint32_t state
- int has_confidence
- uint8_t confidence

8.579.1 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 • Valid values: – 0 - Device is mounted on the cradle – 1 - Device is not mounted on the cradle – 2 - Unknown cradle mount state
confidence	• Cradle Mount Confidence (Optional) • Confidence in the Cradle Mount state expressed as a percentage. • Range - 0 to 100

8.579.2 Field Documentation

8.579.2.1 uint8_t pack_loc_SLQSLOCSetCradleMountConfig_t::confidence

8.579.2.2 int pack_loc_SLQSLOCSetCradleMountConfig_t::has_confidence

8.579.2.3 uint32_t pack_loc_SLQSLOCSetCradleMountConfig_t::state

8.580 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

8.580.1 Detailed Description

This structure contains the LOC Start pack

Parameters

SessionId	• ID of the session as identified by the control point. • Range: 0 to 255
pRecurrence-Type	• Specifies the type of session in which the control point is interested. • Defaults to SINGLE. -Values – 1 - Request periodic position fixes – 2 - Request a single position fix
pHorizontal-AccuracyLvl	• Specifies the horizontal accuracy level required by the control point. • Defaults to LOW • Values – 1 - Low accuracy – 2 - Medium accuracy – 3 - High accuracy
pIntermediate-ReportState	• Specifies if the control point is interested in receiving intermediate reports. • ON by default. • Values – 1 - Intermediate reports are turned on – 2 - Intermediate reports are turned off
appVersionValid	• Specifies whether the application version string contains a valid value • 0x00 (FALSE) – Application version string is invalid • 0x01 (TRUE) – Application version string is valid
LocApplication-Info	• LOC Application Parameters • See loc_LocApplicationInfo for more information
pConfigAltitude-Assumed	• Configuration for Altitude Assumed Info in GNSS SV Info Event • Defaults to ENABLED. • Values – 1 - Enable Altitude Assumed information in GNSS SV Info Event – 2 - Disable Altitude Assumed information in GNSS SV Info Event

8.580.2 Field Documentation

8.580.2.1 loc_LocApplicationInfo* pack_loc_Start_t::pApplicationInfo

8.580.2.2 uint32_t* pack_loc_Start_t::pConfigAltitudeAssumed

8.580.2.3 uint32_t* pack_loc_Start_t::pHorizontalAccuracyLvl

8.580.2.4 uint32_t* pack_loc_Start_t::pIntermediateReportState

8.580.2.5 uint32_t* pack_loc_Start_t::pMinIntervalTime

8.580.2.6 uint32_t* pack_loc_Start_t::pRecurrenceType

8.580.2.7 uint8_t pack_loc_Start_t::SessionId

8.580.2.8 uint16_t pack_loc_Start_t::Tlvresult

8.581 pack_loc_Stop_t Struct Reference

Data Fields

- uint8_t SessionId
- uint16_t Tlvresult

8.581.1 Detailed Description

This structure contains Stop LOC pack

Parameters

sessionId	• ID of the session as identified by the control point. • Range: 0 to 255
Tlvresult	• Unpack result.

8.581.2 Field Documentation

8.581.2.1 uint8_t pack_loc_Stop_t::SessionId

8.581.2.2 uint16_t pack_loc_Stop_t::Tlvresult

8.582 pack_nas_InitiateDomainAttach_t Struct Reference

Data Fields

- uint32_t action

8.582.1 Detailed Description

Pack structure for Initiates a domain attach/detach of the device.

Parameters

action	• Domain action to attempt 1 - Attach 2 - Detach

8.582.2 Field Documentation

8.582.2.1 uint32_t pack_nas_InitiateDomainAttach_t::action

8.583 pack_nas_SetACCOLC_t Struct Reference

Data Fields

- int8_t spc [6]
- uint8_t accolc

8.583.1 Detailed Description

Parameters

spc	servcie programming code
accolc	accolc

8.583.2 Field Documentation

8.583.2.1 uint8_t pack_nas_SetACCOLC_t::accolc

8.583.2.2 int8_t pack_nas_SetACCOLC_t::spc[6]

8.584 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

8.584.1 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 – 0 - Disabled – 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 – 0 - Disabled – 1 - Enabled Note: Enabled can only be specified if pForceRev0 is set to Disabled
pProtocol	• Protocol mask for custom SCP config – 0x00000001 - Subtype 2 Physical Layer – 0x00000002 - Enhanced CCMAC – 0x00000004 - Enhanced ACMAC – 0x00000008 - Enhanced FTCMAC – 0x00000010 - Subtype 3 RTCMAC – 0x00000020 - Subsystem 1 RTCMAC – 0x00000040 - Enhanced Idle – 0x00000080 - Generic Multimode Capable Disc Port – 0xFFFFFFFF - Unknown
pBroadcast	• Broadcast mask for custom SCP config – 0x00000001 - Generic broadcast enabled – 0xFFFFFFFF - Unknown
pApplication	• Application mask for custom SCP config – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application – 0xFFFFFFFF - Unknown
pRoaming	• Roaming preference – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming – 0xFFFFFFFF - Unknown

8.584.2 Field Documentation

8.584.2.1 uint32_t* pack_nas_SetCDMANetworkParameters_t::pApplication

8.584.2.2 uint32_t* pack_nas_SetCDMANetworkParameters_t::pBroadcast

8.584.2.3 uint8_t* pack_nas_SetCDMANetworkParameters_t::pCustomSCP

8.584.2.4 uint8_t* pack_nas_SetCDMANetworkParameters_t::pForceRev0

8.584.2.5 uint32_t* pack_nas_SetCDMANetworkParameters_t::pProtocol

8.584.2.6 uint32_t* pack_nas_SetCDMANetworkParameters_t::pRoaming

8.584.2.7 char* pack_nas_SetCDMANetworkParameters_t::pSPC

8.585 pack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint32_t TechnologyPref
- uint32_t Duration
- uint16_t Tlvresult

8.585.1 Detailed Description

Parameters

TechnologyPref[-IN]	• Bitmask representing the radio technology preference set. • No bits set indicates to the device to automatically determine the technology to use • Values: – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved
Duration[IN]	• Duration of active preference – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call
Tlvresult	• pack result

8.585.2 Field Documentation

8.585.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

8.585.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

8.585.2.3 uint16_t pack_nas_SetNetworkPreference_t::Tlvresult

8.586 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

8.586.1 Detailed Description

Structure for ConfigSigInfo pack.

Parameters

pRSSIThresh	• RSSI threshold List • See nas_RSSIThresh for more details
pECIOThresh	• ECIO Threshold List • See nas_ECIOThresh for more details
pHDRSINR-Thresh	• HDR SINR Threshold List • See nas_HDRSINRThresh for more details
pLTESNR-Thresh	• LTE SNR Threshold List • See nas_LTESNRThresh for more details
pIOThresh	• IO Threshold List • See nas_IOThresh for more details
pRSRQThresh	• RSRQ Threshold List • See nas_RSRQThresh for more details
pRSRPThresh	• RSRP Threshold List • See nas_RSRPThresh for more details
pLTESigRptCfg	• LTE signal report config • See nas_LTESigRptCfg for more details

pTDSCDMASINRCONFThresh	• TD-SCDMA SINR Threshold List • See nas_TDSCDMASINRCONFThresh for more details

8.586.2 Field Documentation

8.586.2.1 nas_ECIOThresh∗ pack_nas_SLQSConfigSigInfo_t::pECIOThresh

8.586.2.2 nas_HDRSINRThresh∗ pack_nas_SLQSConfigSigInfo_t::pHDRSINRThresh

8.586.2.3 nas_IOThresh∗ pack_nas_SLQSConfigSigInfo_t::pIOThresh

8.586.2.4 nas_LTESigRptCfg∗ pack_nas_SLQSConfigSigInfo_t::pLTESigRptCfg

8.586.2.5 nas_LTESNRThresh∗ pack_nas_SLQSConfigSigInfo_t::pLTESNRThresh

8.586.2.6 nas_RSRPThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRPThresh

8.586.2.7 nas_RSRQThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRQThresh

8.586.2.8 nas_RSSIThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSSIThresh

8.586.2.9 nas_TDSCDMASINRCONFThresh∗ pack_nas_SLQSConfigSigInfo_t::pTDSCDMASINRCONFThresh

8.587 pack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- uint8_t ∗ pMncPcsStatus

8.587.1 Detailed Description

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 • Used to interpret the length of the corresponding MNC reported in the PLMN TLV(0x01). • Values – TRUE - MNC is a three-digit value. e.g. a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a two-digit value. e.g. a reported value of 90 corresponds to an MNC value of 90

Note

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

8.587.2 Field Documentation

8.587.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

8.587.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

8.587.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

8.588 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference

Data Fields

- uint32_t regAction
- nas_MNRInfo * pMNRInfo
- uint32_t * pChangeDuration
- uint8_t * pMncPcsDigitStatus

8.588.1 Detailed Description

This structure contains Initiate Network Registration request parameters

Parameters

regAction	• Specifies one of the following register actions : – AUTO_REGISTER - Device registers according to its provisioning and optional parameters supplied with the command are ignored. – MANUAL_REGISTER - Device registers to a specified network and the optional Manual Network Register Information parameter pMNRInfo must also be included for the command to process successfully and supported only for 3GPP.
pMNRInfo	[Optional] • Pointer to structure MNRInfo – See nas_MNRInfo for more information
pChange-Duration	[Optional] • Duration of the change. – 0x00 - Power cycle - Remains active until the next device power cycle – 0x01 - Permanent - Remains active through power cycles until changed by the client
pMncPcsDigit-Status	[Optional] • MNC PCS Digit Include Status – True - MNC is a 3-digit value. – False - MNC is a 2-digit value.

8.588.2 Field Documentation

8.588.2.1 uint32_t* pack_nas_SLQSInitiateNetworkRegistration_t::pChangeDuration

8.588.2.2 uint8_t* pack_nas_SLQSInitiateNetworkRegistration_t::pMncPcsDigitStatus

8.588.2.3 nas_MNRInfo* pack_nas_SLQSInitiateNetworkRegistration_t::pMNRInfo

8.588.2.4 uint32_t pack_nas_SLQSInitiateNetworkRegistration_t::regAction

8.589 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

8.589.1 Detailed Description

Parameters

pCDMARSSI-Thresh	• CDMA RSSI threshold List
pCDMARSSI-Delta	• RSSI delta (in units of 0.1 dBm). • 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). • 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) • 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) • 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) • 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) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pGSMRSSI-Thresh	• GSM RSSI Threshold List • See GSMRSSIThresh for more details
pGSMRSSIDelta	• RSSI delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.

pWCDMARSSI-Thresh	• WCDMA RSSI Threshold List • See WCDMARSSIThresh for more details
pWCDMARSSI-Delta	• RSSI delta (in units of 0.1 dBm). • 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) • 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) • 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) • 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) • 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). • 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) • 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.

8.589.2 Field Documentation

8.589.2.1 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta

8.589.2.2 nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh

8.589.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta

8.589.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh

8.589.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta

8.589.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh

8.589.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta

8.589.2.8 nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh

8.589.2.9 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta

8.589.2.10 nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh

8.589.2.11 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIDelta

8.589.2.12 nas_HDRRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIThresh

8.589.2.13 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRDelta

8.589.2.14 nas_HDRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRThresh

8.589.2.15 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta

8.589.2.16 nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh

8.589.2.17 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta

8.589.2.18 nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh

8.589.2.19 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta

8.589.2.20 nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh

8.589.2.21 nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig

8.589.2.22 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta

8.589.2.23 nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh

8.589.2.24 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta

8.589.2.25 nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh

8.589.2.26 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta

8.589.2.27 nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh

8.589.2.28 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta

8.589.2.29 nas_TDSCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh

8.589.2.30 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta

8.589.2.31 nas_TDSCDMASINRThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh

8.589.2.32 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIODelta

8.589.2.33 nas_WCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh

8.589.2.34 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta

8.589.2.35 nas_WCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh

8.590 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference

Data Fields

- uint8_t namID

8.590.1 Detailed Description

Structure for 3GPP2 Subscription pack.

Parameters

namID	[Mandatory] • 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.

8.590.2 Field Documentation

8.590.2.1 uint8_t pack_nas_SLQSNasGet3GPP2Subscription_t::namID

8.591 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- uint8_t radio_if

8.591.1 Detailed Description

Structure for TxRx Info pack

Parameters

radio_if	[Mandatory] • Radio interface technology of the signal being measured • Valid Values – 0x01 - NAS_RADIO_IF_CDMA_1X - CDMA – 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - HDR – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE

8.591.2 Field Documentation

8.591.2.1 uint8_t pack_nas_SLQSNasGetTxRxInfo_t::radio_if

8.592 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

8.592.1 Detailed Description

Parameters

pSystemSelectionInd	[Optional] • System Selection Preference indication registration. The following callbacks would not be invoked if the indication is disabled. tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem – 0x00 - Disable – 0x01 - Enable
pDDTMInd	[Optional] • DDTM (Data Dedicated Transmission Mode) indication registration. The following callbacks would not be invoked if the indication is disabled. tFNDDTM – 0x00 - Disable – 0x01 - Enable
pServingSystemInd	[Optional] • Serving System indication registration. The following callbacks would not be invoked if the indication is disabled. tFNBandPreference – 0x00 - Disable – 0x01 - Enable
pDualStandByPrefInd	[Optional] • Dual Standby Preference indication registration.The following callbacks would not be invoked if the indication is disabled. tFNDualStandByPref – 0x00 - Disable – 0x01 - Enable
pSubscriptionInfoInd	[Optional] • Subscription Information indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSubscriptionInfo – 0x00 - Disable – 0x01 - Enable
pNetworkTimeInd	[Optional] • Network Time indication registration.The following callbacks would not be invoked if the indication is disabled. tFNNetworkTime – 0x00 - Disable – 0x01 - Enable
pSysInfoInd	[Optional] • System Information indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSysInfo – 0x00 - Disable – 0x01 - Enable

pSignalStrength-Ind	[Optional] • Signal Strength indication registration.The following callbacks would not be invoked if the indication is disabled. tFNSigInfo – 0x00 - Disable – 0x01 - Enable
pErrorRateInd	[Optional] • Error Rate indication registration.The following callbacks would not be invoked if the indication is disabled. tFNErrRate – 0x00 - Disable – 0x01 - Enable
pHDRNewUATI-AssInd	[Optional] • HDR New UATI Assigned indication registration.The following callbacks would not be invoked if the indication is disabled. tFNHDRUATIUpdate – 0x00 - Disable – 0x01 - Enable
pHDRSession-CloseInd	[Optional] • HDR Session Closed indication registration.The following callbacks would not be invoked if the indication is disabled. tFNHDRSessionClose – 0x00 - Disable – 0x01 - Enable
pManaged-RoamingInd	[Optional] • Managed Roaming indication registration.The following callbacks would not be invoked if the indication is disabled. tFNManagedRoaming – 0x00 - Disable
pNetworkReject-Ind	[Optional] • Network reject information indication registration.The following callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pSuppressSys-InfoInd	[Optional] • Controls the reporting of QMI_NAS_SYS_INFO_IND when only the reject_cause field has changed. Values: – 0x00 - Disable – 0x01 - Enable- 0x01 - Enable
pLTECphyCa	[Optional] • LTE Physical Carrier Aggregation Information.The following callbacks would not be invoked if the indication is disabled. tFNManagedRoaming – 0x00 - Disable (default value) – 0x01 - Enable

8.592.2 Field Documentation

8.592.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.592.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.592.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.592.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

8.592.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

8.592.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

8.592.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

8.592.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkRejectInd

8.592.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

8.592.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

8.592.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

8.592.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

8.592.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSuppressSysInfoInd

8.592.2.14 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

8.592.2.15 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

8.593 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

8.593.1 Detailed Description

This structure contains the OTA message indication.

Parameters

lteEsmUl	• 0 - do not report • 1 - report LTE ESM uplink messages

lteEsmDl	• 0 - do not report • 1 - report LTE ESM downlink messages
lteEmmUl	• 0 - do not report • 1 - report LTE EMM uplink messages
lteEmmDl	• 0 - do not report • 1 - report GSM/UMTS uplink messages
gsmUmtsUl	• 0 - do not report • 1 - report GSM/UMTS uplink messages
gsmUmtsDl	• 0 - do not report • 1 - report GSM/UMTS downlink messages
pRankIndicator-Ind	• 0 - do not report • 1 - report Rank Indicator messages
pTimer	• 0 - do not report • 1 - report Timer Indicator messages

8.593.2 Field Documentation

8.593.2.1 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsDl

8.593.2.2 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsUl

8.593.2.3 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmDl

8.593.2.4 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmUl

8.593.2.5 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmDl

8.593.2.6 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmUl

8.593.2.7 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pRankIndicatorInd

8.593.2.8 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pTimer

8.594 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference

Data Fields

- nas_wcdmaUARFCN * pWcdmaUARFCN

- nas_lteEARFCN * pLteEARFCN
- nas_ltePCI * pLtePCI

8.594.1 Detailed Description

Structure for ChannelLock pack.

Parameters

pWcdmaUARFCN	[Optional] • See nas_wcdmaUARCFN for more information
pLteEARFCN	[Optional] • See nas_lteEARCFN for more information
pLtePCI	[Optional] • See nas_ltePCI for more information

8.594.2 Field Documentation

8.594.2.1 nas_lteEARFCN* pack_nas_SLQSNASSwiSetChannelLock_t::pLteEARFCN

8.594.2.2 nas_ltePCI* pack_nas_SLQSNASSwiSetChannelLock_t::pLtePCI

8.594.2.3 nas_wcdmaUARFCN* pack_nas_SLQSNASSwiSetChannelLock_t::pWcdmaUARFCN

8.595 pack_nas_SLQSSetBandPreference_t Struct Reference

Data Fields

- uint64_t bandPref

8.595.1 Detailed Description

Provides information about the band preference.

Parameters

bandpreference	• Bit mask representing the band preference to be set. • Bit position meanings: – 0 - BC0_A - Band Class 0, A-System – 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band – 2 - BC1 - Band Class 1, all blocks – 3 - BC2 - Band Class 2 place holder – 4 - BC3 - Band Class 3, A-System – 5 - BC4 - Band Class 4, all blocks – 6 - BC5 - Band Class 5, all blocks – 7 - GSM_DCS_1800 - GSM DCS band – 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band – 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band – 10 - BC6 - Band Class 6 – 11 - BC7 - Band Class 7 – 12 - BC8 - Band Class 8 – 13 - BC9 - Band Class 9 – 14 - BC10 - Band Class 10 – 15 - BC11 - Band Class 11 – 16 - GSM_450 - GSM 450 band – 17 - GSM_480 - GSM 480 band – 18 - GSM_750 - GSM 750 band – 19 - GSM_850 - GSM 850 band – 20 - GSM_RGSM_900 - GSM Railways GSM Band – 21 - GSM_PCS_1900 - GSM PCS band – 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band – 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band – 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band – 25 - WCDMA_IV_1700 - WCDMA US 1700 band – 26 - WCDMA_V_850 - WCDMA US 850 band – 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band – 28 - BC12 - Band Class 12 – 29 - BC14 - Band Class 14 – 30 - RESERVED_2 - Reserved 2 – 31 - BC15 - Band Class 15 – 32 - 47 - Reserved – 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band – 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band – 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band – 51 to 55 - Reserved – 56 - BBC16 - Band Class 16 – 57 - BC17 - Band Class 17 – 58 - BC18 - Band Class 18 – 59 - BC19 - Band Class 19 – 60 to 64 - Reserved

8.595.2 Field Documentation

8.595.2.1 uint64_t pack_nas_SLQSSetBandPreference_t::bandPref

8.596 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

Data Fields

- uint8_t bEnable
- nas_SLQSSignalStrengthsIndReq * pSigIndReq

8.596.1 Detailed Description

Parameters

bEnable	0/1 to disable/enable RSSI signal stregth indication
pSigIndReq	parameters to control signal strength indication

8.596.2 Field Documentation

8.596.2.1 uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable

8.596.2.2 nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq

8.597 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

8.597.1 Detailed Description

Contain the system selection preferences.

Parameters

pEmerMode	• Optional parameter specifying the emergency Mode • Values: – 0 - OFF (normal) – 1 - ON (Emergency)
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE

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

pPRLPref	• Optional parameter indicating the CDMA PRL Preference • Values: – 0x0001 - Acquire available system only on the A side – 0x0002 - Acquire available system only on the B side – 0x3FFF - Acquire any available systems
pRoamPref	• Optional parameter indicating the roaming Preference • Values: – 0x01 - Acquire only systems for which the roaming indicator is off – 0x02 - Acquire a system as long as its roaming indicator is not off – 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only – 0xFF - Acquire systems, regardless of their roaming indicator
pLTEBandPref	• Optional parameter • Bit mask representing the LTE band preference • Bit Values – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – All other bits are reserved

pNetSelPref	- netSelectionPref • Optional parameter for specifying Network Selection Preference • Modem selects networks based on this parameter(if present). • see netSelectionPref for more information
pChgDuration	• Optional parameter specifying the duration of the change • Values: – 0x00 - Power cycle - Remains active until the next device power cycle – 0x01 - Permanent - Remains active through power cycles until changed by client – Device will use "0x01 - permanent" as default if this parameter is omitted
pMNCIncPCS-DigStat	• Optional parameter indicating if MNC includes PCS digit • Values: – TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
pSrvDomainPref	• Optional parameter indicating Service domain preference • Values: – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach
pGWAcqOrder-Pref	• Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM

8.597.2 Field Documentation

8.597.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

8.597.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

8.597.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

8.597.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

8.597.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

8.597.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

8.597.2.7 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref

8.597.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat

8.597.2.9 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref

8.597.2.10 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref

8.597.2.11 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref

8.597.2.12 unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT

8.597.2.13 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref

8.597.2.14 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref

8.597.2.15 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction

8.597.2.16 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdscdmaBandPref

8.598 pack_nas_SLQSSwiPSDetach_t Struct Reference

Data Fields

- uint8_t * pDetachAction

8.598.1 Detailed Description

Structure for PS connection detach pack.

Parameters

pDetachAction[IN]	• Values – 2- Initiates an immediate packet domain detach.

8.598.2 Field Documentation

8.598.2.1 uint8_t* pack_nas_SLQSSwiPSDetach_t::pDetachAction

8.599 pack_pds_SetPDSDefaults_t Struct Reference

Data Fields

- uint32_t operation
- uint8_t timeout
- uint32_t interval
- uint32_t accuracy

8.599.1 Detailed Description

Structure containing session configuration parameters.

Parameters

operation	• Current session operating mode – 0 - Standalone – 1 - MS based – 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)

8.599.2 Field Documentation

8.599.2.1 uint32_t pack_pds_SetPDSDefaults_t::accuracy

8.599.2.2 uint32_t pack_pds_SetPDSDefaults_t::interval

8.599.2.3 uint32_t pack_pds_SetPDSDefaults_t::operation

8.599.2.4 uint8_t pack_pds_SetPDSDefaults_t::timeout

8.600 pack_pds_SetPDSState_t Struct Reference

Data Fields

- uint32_t enable

8.600.1 Detailed Description

Structure contain PDS state parameters.

Parameters

enable	• Desired PDS state – Zero - disable – Non-Zero - enable

8.600.2 Field Documentation

8.600.2.1 uint32_t pack_pds_SetPDSState_t::enable

8.601 pack_pds_SetPortAutomaticTracking_t Struct Reference

Data Fields

- uint32_t bAuto

8.601.1 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 – 0x00 - Disabled – 0x01 - Enabled

8.601.2 Field Documentation

8.601.2.1 uint32_t pack_pds_SetPortAutomaticTracking_t::bAuto

8.602 pack_qmi_t Struct Reference

Data Fields

- uint16_t xid
- int timeout
- uint16_t msgid
- uint8_t svc

8.602.1 Detailed Description

qmi request context

Parameters

in	*xid*	transaction id
out	*timeout*	timeout recommended in seconds
out	*msgid*	message id
out	*svc*	qmi service

8.602.2 Field Documentation

8.602.2.1 uint16_t pack_qmi_t::msgid

8.602.2.2 uint8_t pack_qmi_t::svc

8.602.2.3 int pack_qmi_t::timeout

8.602.2.4 uint16_t pack_qmi_t::xid

8.603 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

Data Fields

- uint32_t apnId

8.603.1 Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

Parameters

apnId[IN]	• APN id

8.603.2 Field Documentation

8.603.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.604 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

Data Fields

- uint32_t apnId

8.604.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

Parameters

apnId[IN]	• APN id

8.604.2 Field Documentation

8.604.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

8.605 pack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- uint8_t enable

8.605.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

Parameters

enable[IN]	• 1 - Enable QoS event reporting • 0 - Disable QoS event reporting

8.605.2 Field Documentation

8.605.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

8.606 pack_rms_SetSMSWake_t Struct Reference

Data Fields

- uint32_t enabled
- uint32_t wake_mask

8.606.1 Detailed Description

Parameters

enabled	• SMS wake functionality enabled – 0 - Disabled – 1 - Enabled
wake_mask	• SMS wake mask to search for incoming messages (only relevant when enabled)

8.606.2 Field Documentation

8.606.2.1 uint32_t pack_rms_SetSMSWake_t::enabled

8.606.2.2 uint32_t pack_rms_SetSMSWake_t::wake_mask

8.607 pack_sar_SLQSSetRfSarState_t Struct Reference

Data Fields

- uint32_t RfSarState

8.607.1 Detailed Description

Parameters

RfSarState	• SAR RF State – 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

8.607.2 Field Documentation

8.607.2.1 uint32_t pack_sar_SLQSSetRfSarState_t::RfSarState

8.608 pack_sms_SaveSMS_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage

8.608.1 Detailed Description

Sutructure containing parameters to Save an SMS message to device memory

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
messageFormat	• Message format – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 6 - GSM/WCDMA PP
messageSize	• The length of the message contents in bytes
pMessage	• The message contents

8.608.2 Field Documentation

8.608.2.1 uint32_t pack_sms_SaveSMS_t::messageFormat

8.608.2.2 uint32_t pack_sms_SaveSMS_t::messageSize

8.608.2.3 uint8_t* pack_sms_SaveSMS_t::pMessage

8.608.2.4 uint32_t pack_sms_SaveSMS_t::storageType

8.609 pack_sms_SendSMS_t Struct Reference

Data Fields

- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage
- uint8_t * pLinktimer

8.609.1 Detailed Description

Parameters

messageFormat	• Message format – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 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

8.609.2 Field Documentation

8.609.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

8.609.2.2 uint32_t pack_sms_SendSMS_t::messageSize

8.609.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

8.609.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

8.610 pack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- enum eqmiCbkSetStatus status

8.610.1 Detailed Description

Parameters

status	callback parameter

8.610.2 Field Documentation

8.610.2.1 enum eqmiCbkSetStatus pack_sms_SetNewSMSCallback_t::status

8.611 pack_sms_SetSMSCAddress_t Struct Reference

Data Fields

- uint8_t * pSMSCAddress
- uint8_t * pSMSCType

8.611.1 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).

8.611.2 Field Documentation

8.611.2.1 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCAddress

8.611.2.2 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCType

8.612 pack_sms_SLQSDeleteSMS_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t * pMessageIndex
- uint32_t * pMessageTag
- uint8_t * pMessageMode

8.612.1 Detailed Description

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
pMessageIndex	• (Optional) message index
pMessageTag	• (Optional) message tag – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent
pMessageMode	• (Optional) message mode • this must be included if the device is capable of supporting more than one protocol • e.g. CDMA and GW – 0x00 - CDMA, LTE (if network type is CDMA) – 0x01 - GW, LTE (if network type is UMTS)

8.612.2 Field Documentation

8.612.2.1 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageIndex

8.612.2.2 uint8_t* pack_sms_SLQSDeleteSMS_t::pMessageMode

8.612.2.3 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageTag

8.612.2.4 uint32_t pack_sms_SLQSDeleteSMS_t::storageType

8.613 pack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint8_t * pMessageMode

8.613.1 Detailed Description

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV

messageIndex	• Message index
pMessageMode	• 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)

8.613.2 Field Documentation

8.613.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

8.613.2.2 uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode

8.613.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

8.614 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

Data Fields

- uint8_t mode

8.614.1 Detailed Description

Parameters

mode	• Mode – 0x00 - CDMA, LTE (if network type is CDMA) – 0x01 - GW, LTE (if network type is UMTS)

8.614.2 Field Documentation

8.614.2.1 uint8_t pack_sms_SLQSGetSmsBroadcastConfig_t::mode

8.615 pack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t * pRequestedTag
- uint8_t * pMessageMode

8.615.1 Detailed Description

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
requestedTag	• (Optional) Message tag – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent
messageMode	• 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)

8.615.2 Field Documentation

8.615.2.1 uint8_t* pack_sms_SLQSGetSMSList_t::pMessageMode

8.615.2.2 uint32_t* pack_sms_SLQSGetSMSList_t::pRequestedTag

8.615.2.3 uint32_t pack_sms_SLQSGetSMSList_t::storageType

8.616 pack_sms_SLQSModifySMSStatus_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint32_t messageTag
- uint8_t * pMessageMode

8.616.1 Detailed Description

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
messageIndex	• Message index
messageTag	• Message tag – 0 - Read – 1 - Not read

pMessageMode	• 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)

8.616.2 Field Documentation

8.616.2.1 uint32_t pack_sms_SLQSModifySMSStatus_t::messageIndex

8.616.2.2 uint32_t pack_sms_SLQSModifySMSStatus_t::messageTag

8.616.2.3 uint8_t* pack_sms_SLQSModifySMSStatus_t::pMessageMode

8.616.2.4 uint32_t pack_sms_SLQSModifySMSStatus_t::storageType

8.617 pack_sms_SLQSSendAsyncSMS_t Struct Reference

Data Fields

- sms_sendAsyncsmsParams * pSendSmsParams

8.617.1 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

8.617.2 Field Documentation

8.617.2.1 sms_sendAsyncsmsParams* pack_sms_SLQSSendAsyncSMS_t::pSendSmsParams

8.618 pack_sms_SLQSSetIndicationRegister_t Struct Reference

Data Fields

- sms_setIndicationReg * pSetIndicationRegReq

8.618.1 Detailed Description

Structure contain Parameter that sets the registration state of different WMS indications.

Parameters

pSetIndication-Reg	• Pointer to structure of indicationRegReqParams – See sms_setIndicationReg for more information

8.618.2 Field Documentation

8.618.2.1 sms_setIndicationReg* pack_sms_SLQSSetIndicationRegister_t::pSetIndicationRegReq

8.619 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference

Data Fields

- uint8_t mode
- uint8_t broadcastActivate

8.619.1 Detailed Description

Structure contain parameters that Enables or disables the reception of broadcast SMS messages.

Parameters

Mode	• Mode • 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)
broadcast-Activate	• 0x00 - Disable broadcast • 0x01 - Activate broadcast

8.619.2 Field Documentation

8.619.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::broadcastActivate

8.619.2.2 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::mode

8.620 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference

Data Fields

- uint8_t mode
- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

8.620.1 Detailed Description

Structure containing parameters that provides the information about the SMS BroadcastConfiguration

Parameters

mode	• Mode – 0x00 - CDMA, LTE (if network type is CDMA) – 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).

8.620.2 Field Documentation

8.620.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastConfig_t::mode

8.620.2.2 sms_qaQmi3GPPBroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pBroadcastConfig

8.620.2.3 sms_qaQmi3GPP2BroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pCDMABroadcastConfig

8.621 pack_sms_SLQSSetSmsStorage_t Struct Reference

Data Fields

- uint8_t smsStorage

8.621.1 Detailed Description

Structure contain Parameter to set the SMS Storage on the device

Parameters

smsStorage	• SMS Storage – 0x01 - device's permanent memory – 0x02 - UICC

8.621.2 Field Documentation

8.621.2.1 uint8_t pack_sms_SLQSSetSmsStorage_t::smsStorage

8.622 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

Data Fields

- sms_maxStorageSizeReq * pMaxStorageSizeReq

8.622.1 Detailed Description

Parameters

pMaxStorage-SizeReq	• Request parameters for SmsSLQSGetMaxStorageSize – See sms_maxStorageSizeReq for more information

8.622.2 Field Documentation

8.622.2.1 sms_maxStorageSizeReq∗ pack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeReq

8.623 pack_sms_SLQSSmsSetRoutes_t Struct Reference

Data Fields

- sms_setRoutesReq ∗ pSetRoutesReq

8.623.1 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 – See sms_setRoutesReq for more information

8.623.2 Field Documentation

8.623.2.1 sms_setRoutesReq∗ pack_sms_SLQSSmsSetRoutes_t::pSetRoutesReq

8.624 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

8.624.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

Parameters

function	• Setting to indicate when modem should start an automatic GNSS fix – 0 - disabled – 1 - At bootup – 2 - When NMEA port is opened

set_function	• 0 - do not set to modem • l - set to modem
fix_type	• Type of GNSS fix: – 1 - Default Engine mode – 2 - MS-Based – 3 - MS-Assisted – 4 - Standalone
set_fix_type	• 0 - do not set to modem • l - set to modem
max_time	• Maximum time allowed for the receiver to get a fix in seconds • Valid range: 1-255
set_max_time	• 0 - do not set to modem • l - set to modem
max_dist	• Maximum uncertainty of a fix measured by distance in meters • Valid range: 1 - 4294967280
set_max_dist	• 0 - do not set to modem • l - set to modem
fix_rate	• Time between fixes in seconds • Valid range: 1–65535
set_fix_rate	• 0 - do not set to modem • l - set to modem

8.624.2 Field Documentation

8.624.2.1 uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate

8.624.2.2 uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type

8.624.2.3 uint8_t pack_swiloc_SwiLocSetAutoStart_t::function

8.624.2.4 uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist

8.624.2.5 uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time

8.624.2.6 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate

8.624.2.7 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type

8.624.2.8 int pack_swiloc_SwiLocSetAutoStart_t::set_function

8.624.2.9 int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist

8.624.2.10 int pack_swiloc_SwiLocSetAutoStart_t::set_max_time

8.625 pack_swioma_SLQSOMADMCancelSession_t Struct Reference

Data Fields

- uint32_t sessionType

8.625.1 Detailed Description

Structure that contains the session type for OMA cancel session command

Parameters

sessionType[IN]	• Session type – 0x01 - FOTA, to check availability of FW Update – 0xFF - Cancel any active OMADM session

8.625.2 Field Documentation

8.625.2.1 uint32_t pack_swioma_SLQSOMADMCancelSession_t::sessionType

8.626 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint32_t SessionType

8.626.1 Detailed Description

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

Parameters

SessionType[IN]	• Session type – 0x01 - FOTA – 0xFF - Any active OMADM session. If no active sessions are available, then previous OMADM session info is returned

8.626.2 Field Documentation

8.626.2.1 uint32_t pack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

8.627 pack_swioma_SLQSOMADMSendSelection_t Struct Reference

Data Fields

- uint32_t selection
- uint32_t * pDeferTime
- uint32_t * pRejectReason

8.627.1 Detailed Description

Structure containing the OMA DM selection

Parameters

selection[IN]	• OMA-DM NIA Selection – 0x01 - Accept – 0x02 - Reject – 0x03 - Defer
pDeferTime[IN]	• Defer time in minutes. A value of 0 will cause the prompt to be resent immediately. • This TLV is mandatory if selection is set to 0x03.
pRejectReason[IN]	• Reject Reason • This TLV is processed if selection is set to 0x02. If it is not present, the reject reason 0 is used as default.

8.627.2 Field Documentation

8.627.2.1 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pDeferTime

8.627.2.2 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pRejectReason

8.627.2.3 uint32_t pack_swioma_SLQSOMADMSendSelection_t::selection

8.628 pack_swioma_SLQSOMADMSetSettings_t Struct Reference

Data Fields

- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t * pAutosdm
- uint8_t * pFwAutoCheck

8.628.1 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 – 0x00 - Firmware autodownload FALSE – 0x01 - Firmware autodownload TRUE
FOTAUpdate	• 1 byte parameter indicating FOTA Automatic update – 0x00 - Firmware autoupdate FALSE – 0x01 - Firmware autoupdate TRUE
pAutosdm[IN]	• Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject
pFwAutoCheck[-IN]	• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 0x01 - Enabled

8.628.2 Field Documentation

8.628.2.1 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAdownload

8.628.2.2 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAUpdate

8.628.2.3 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pAutosdm

8.628.2.4 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pFwAutoCheck

8.629 pack_swioma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t sessionType

8.629.1 Detailed Description

Structure that contains the session type for OMA start session command

Parameters

sessionType[IN]	• Session type – 0x01 - FOTA, to check availability of FW Update – 0x02 - DM, to check availability of DM Update – 0x03 - PRL, to check availability of PRL Update

8.629.2 Field Documentation

8.629.2.1 uint32_t pack_swioma_SLQSOMADMStartSession_t::sessionType

8.630 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

8.630.1 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. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
pIndicationToken(optional)	• Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function

Note

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

8.630.2 Field Documentation

8.630.2.1 uim_changeUIMPIN pack_uim_ChangePin_t::changePIN

8.630.2.2 uim_encryptedPIN1 pack_uim_ChangePin_t::EncryptedPIN1

8.630.2.3 uint32_t* pack_uim_ChangePin_t::pIndicationToken

8.630.2.4 uint8_t* pack_uim_ChangePin_t::pKeyReferenceID

8.630.2.5 uim_sessionInformation pack_uim_ChangePin_t::sessionInfo

8.630.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

8.631 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

8.631.1 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. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function

pEncrypt-Data(optional)	• Encrypt Data. • Indicates whether the data read from the card is to be encrypted.

Note

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

8.631.2 Field Documentation

8.631.2.1 uim_fileInfo pack_uim_ReadTransparent_t::fileIndex

8.631.2.2 uint8_t* pack_uim_ReadTransparent_t::pEncryptData

8.631.2.3 uint32_t* pack_uim_ReadTransparent_t::pIndicationToken

8.631.2.4 uim_readTransparentInfo pack_uim_ReadTransparent_t::readTransparent

8.631.2.5 uim_sessionInformation pack_uim_ReadTransparent_t::sessionInfo

8.631.2.6 uint16_t pack_uim_ReadTransparent_t::Tlvresult

8.632 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

8.632.1 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. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.

pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function

Note

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

8.632.2 Field Documentation

8.632.2.1 uim_encryptedPIN1 pack_uim_SetPinProtection_t::EncryptedPIN1

8.632.2.2 uint32_t* pack_uim_SetPinProtection_t::pIndicationToken

8.632.2.3 uim_setPINProtection pack_uim_SetPinProtection_t::pinProtection

8.632.2.4 uint8_t* pack_uim_SetPinProtection_t::pKeyReferenceID

8.632.2.5 uim_sessionInformation pack_uim_SetPinProtection_t::sessionInfo

8.632.2.6 uint16_t pack_uim_SetPinProtection_t::Tlvresult

8.633 pack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_authenticationData authData
- uint32_t * pIndicationToken

8.633.1 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. • 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.

8.633.2 Field Documentation

8.633.2.1 uim_authenticationData pack_uim_SLQSUIMAuthenticate_t::authData

8.633.2.2 uint32_t* pack_uim_SLQSUIMAuthenticate_t::pIndicationToken

8.633.2.3 uim_sessionInformation pack_uim_SLQSUIMAuthenticate_t::sessionInfo

8.634 pack_uim_SLQSUIMDepersonalization_t Struct Reference

Data Fields

- uim_depersonalizationInformation depersonilisationInfo

8.634.1 Detailed Description

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

Parameters

depersonilisation-Info	• See uim_depersonalizationInformation for more information.

8.634.2 Field Documentation

8.634.2.1 uim_depersonalizationInformation pack_uim_SLQSUIMDepersonalization_t::depersonilisationInfo

8.635 pack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t eventMask

8.635.1 Detailed Description

Parameters

eventMask	- bit 1 - card status • bit 4 - physical slot status

8.635.2 Field Documentation

8.635.2.1 uint32_t pack_uim_SLQSUIMEventRegister_t::eventMask

8.636 pack_uim_SLQSUIMGetConfiguration_t Struct Reference

Data Fields

- uint32_t * pConfigurationMask

8.636.1 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 – Bit 0 - Automatic selection – Bit 1 - Personalization status – Bit 2 - Halt subscription – All other bits are reserved for future use

Note

- if the TLV is missing, the service returns all configuration items in the response.

8.636.2 Field Documentation

8.636.2.1 uint32_t* pack_uim_SLQSUIMGetConfiguration_t::pConfigurationMask

8.637 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_fileInfo fileIndex
- uint32_t * pIndicationToken

8.637.1 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. • 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.

8.637.2 Field Documentation

8.637.2.1 uim_fileInfo pack_uim_SLQSUIMGetFileAttributes_t::fileIndex

8.637.2.2 uint32_t* pack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

8.637.2.3 uim_sessionInformation pack_uim_SLQSUIMGetFileAttributes_t::sessionInfo

8.638 pack_uim_SLQSUIMPowerDown_t Struct Reference

Data Fields

- uint8_t slot

8.638.1 Detailed Description

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

Parameters

slot	• Indicates the slot to be used. – 1 - Slot 1 – 2 - Slot 2

8.638.2 Field Documentation

8.638.2.1 uint8_t pack_uim_SLQSUIMPowerDown_t::slot

8.639 pack_uim_SLQSUIMPowerUp_t Struct Reference

Data Fields

- uint8_t slot
- uint8_t * pIgnoreHotSwapSwitch

8.639.1 Detailed Description

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

Parameters

slot	• Indicates the slot to be used. – 1 - Slot 1 – 2 - Slot 2

pIgnoreHot-Swap-Switch(optional)	• Hot-swap switch status. – 0 - Checks the hot-swap switch status – 1 - Ignores the hot-swap switch status

8.639.2 Field Documentation

8.639.2.1 uint8_t* pack_uim_SLQSUIMPowerUp_t::pIgnoreHotSwapSwitch

8.639.2.2 uint8_t pack_uim_SLQSUIMPowerUp_t::slot

8.640 pack_uim_SLQSUIMRefreshComplete_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uint8_t refreshComplete

8.640.1 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: – 0 - Refresh was not completed successfully – 1 - Refresh was completed successfully

8.640.2 Field Documentation

8.640.2.1 uint8_t pack_uim_SLQSUIMRefreshComplete_t::refreshComplete

8.640.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshComplete_t::sessionInfo

8.641 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo

8.641.1 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.

8.641.2 Field Documentation

8.641.2.1 uim_sessionInformation pack_uim_SLQSUIMRefreshGetLastEvent_t::sessionInfo

8.642 pack_uim_SLQSUIMRefreshOK_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uint8_t OKtoRefresh

8.642.1 Detailed Description

This structure contains Parameters of the Session Information

Parameters

sessionInfo	• Session Information • See uim_sessionInformation for more information
OKtoRefresh	• Indicates whether a refresh is OK. Valid values: – 0 - Not OK to refresh – 1 - OK to refresh

8.642.2 Field Documentation

8.642.2.1 uint8_t pack_uim_SLQSUIMRefreshOK_t::OKtoRefresh

8.642.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshOK_t::sessionInfo

8.643 pack_uim_SLQSUIMRefreshRegister_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_registerRefresh regRefresh

8.643.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Register.

Parameters

sessionInfo	• Session Information params • See uim_sessionInformation for more information
regRefresh	• Register Refresh parameters • See uim_registerRefresh for more information

8.643.2 Field Documentation

8.643.2.1 uim_registerRefresh pack_uim_SLQSUIMRefreshRegister_t::regRefresh

8.643.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshRegister_t::sessionInfo

8.644 pack_uim_SLQSUIMSwitchSlot_t Struct Reference

Data Fields

- uint8_t bLogicalSlot
- uint32_t ulPhysicalSlot

8.644.1 Detailed Description

This structure contains information of the request parameters associated with a Switch Slot.

Parameters

bLogicalSlot	• Indicates the slot to be used. – 1 - Slot 1 – 2 - Slot 2 – 3 - Slot 3 – 4 - Slot 4 – 5 - Slot 5
ulPhysicalSlot	• 1 - Slot 1 • 2 - Slot 2 • 3 - Slot 3 • 4 - Slot 4 • 5 - Slot 5

8.644.2 Field Documentation

8.644.2.1 uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot

8.644.2.2 uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot

8.645 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

8.645.1 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. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
pIndicationToken(optional)	• Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function

8.645.2 Field Documentation

8.645.2.1 uim_encryptedPIN1 pack_uim_UnblockPin_t::EncryptedPIN1

8.645.2.2 uint32_t* pack_uim_UnblockPin_t::pIndicationToken

8.645.2.3 uim_unblockUIMPIN pack_uim_UnblockPin_t::pinProtection

8.645.2.4 uint8_t* pack_uim_UnblockPin_t::pKeyReferenceID

8.645.2.5 uim_sessionInformation pack_uim_UnblockPin_t::sessionInfo

8.645.2.6 uint16_t pack_uim_UnblockPin_t::TlvresultIt

8.646 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 TlvresultIt

8.646.1 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.
pKeyReferenceID(optional)	• Indicates the PIN key reference ID. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
pIndicationToken(optional)	• Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function

Note

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

8.646.2 Field Documentation

8.646.2.1 uim_encryptedPIN1* pack_uim_VerifyPin_t::pEncryptedPIN1

8.646.2.2 uint32_t* pack_uim_VerifyPin_t::pIndicationToken

8.646.2.3 uint8_t* pack_uim_VerifyPin_t::pKeyReferenceID

8.646.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

8.646.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

8.646.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

8.647 pack_voice_AnswerUSSD_t Struct Reference

Data Fields

- uint8_t * pInfo

8.647.1 Detailed Description

Parameters

pInfo	• USS information • See voice_USSInfo for more details

8.647.2 Field Documentation

8.647.2.1 uint8_t* pack_voice_AnswerUSSD_t::pInfo

8.648 pack_voice_OriginateUSSD_t Struct Reference

Data Fields

- uint8_t * pInfo

8.648.1 Detailed Description

Parameters

pInfo	• USS information • See voice_USSInfo for more details

8.648.2 Field Documentation

8.648.2.1 uint8_t* pack_voice_OriginateUSSD_t::pInfo

8.649 pack_voice_SLQSOriginateUSSD_t Struct Reference

Data Fields

- uint8_t ussDCS
- uint8_t ussLen

- uint8_t ussData [182]

8.649.1 Detailed Description

This structure contains USS Information

Parameters

ussDCS	• 1 - ASCII coding scheme • 2 - 8-BIT coding scheme • 3 - UCS2
ussLen	• Range 1 to 182
ussData	• Data encoded as per the DCS

8.649.2 Field Documentation

8.649.2.1 uint8_t pack_voice_SLQSOriginateUSSD_t::ussData[182]

8.649.2.2 uint8_t pack_voice_SLQSOriginateUSSD_t::ussDCS

8.649.2.3 uint8_t pack_voice_SLQSOriginateUSSD_t::ussLen

8.650 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference

Data Fields

- uint8_t lineValue

8.650.1 Detailed Description

This structure contains ALS Select Line Information Parameters.

Parameters

lineValue	• ALS Line Value. – 0x00 - ALS_LINE1 - Line 1 (default) – 0x01 - ALS_LINE2 - Line 2

8.650.2 Field Documentation

8.650.2.1 uint8_t pack_voice_SLQSVoiceALSSelectLine_t::lineValue

8.651 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference

Data Fields

- uint8_t switchOption

8.651.1 Detailed Description

This structure contains ALS Set Line Switching Information Parameters.

Parameters

switchOption	• Switch Option. – 0x00 - VOICE_LINE_SWITCHING_NOT_ALLOWED - Line switching is not allowed – 0x01 - VOICE_LINE_SWITCHING_ALLOWED - Line switching is allowed

8.651.2 Field Documentation

8.651.2.1 uint8_t pack_voice_SLQSVoiceALSSetLineSwitching_t::switchOption

8.652 pack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.652.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

pCallId	• Unique call identifier for the call that must be answered.

8.652.2 Field Documentation

8.652.2.1 uint8_t* pack_voice_SLQSVoiceAnswerCall_t::pCallId

8.653 pack_voice_SLQSVoiceBindSubscription_t Struct Reference

Data Fields

- uint8_t subsType

8.653.1 Detailed Description

This structure contains Bind Subscription Information Parameters.

Parameters

subsType	• Subscription Type. – 0x00 - VOICE_SUBS_TYPE_PRIMARY - Primary – 0x01 - VOICE_SUBS_TYPE_SECONDARY - Secondary

8.653.2 Field Documentation

8.653.2.1 uint8_t pack_voice_SLQSVoiceBindSubscription_t::subsType

8.654 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference

Data Fields

- voice_burstDTMFInfo BurstDTMFInfo
- voice_DTMFLengths * pBurstDTMFLengths

8.654.1 Detailed Description

This structure contains parameters of burst Dual-Tone Multifrequency (DTMF)

Parameters

BurstDTMFInfo	• Burst DTMF Information – See voice_burstDTMFInfo for more information
pBurstDTMF-Lengths	[optional] • DTMF Lengths – See voice_DTMFLengths for more information

8.654.2 Field Documentation

8.654.2.1 voice_burstDTMFInfo pack_voice_SLQSVoiceBurstDTMF_t::BurstDTMFInfo

8.654.2.2 voice_DTMFLengths* pack_voice_SLQSVoiceBurstDTMF_t::pBurstDTMFLengths

8.655 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

8.655.1 Detailed Description

This structure contains Voice Call Request Parameters

Parameters

callNumber[81]	• Number to be dialed in ASCII string, NULL terminated. • 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: – 0x00 - CALL_TYPE_VOICE - Voice (automatic selection) – 0x01 - CALL_TYPE_VOICE_FORCED - Avoid modem call classification – 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP∗ – 0x09 - CALL_TYPE_EMERGENCY - Emergency
pCLIRType(optional)	• CLIR type values are: – 0x01 - CLIR_SUPPRESSION - Suppression – 0x02 - CLIR_INVOCATION - Invocation
pUUSInFo(optional)	• Pointer to structure of UUSInfo – See voice_UUSInfo for more information
pCUGInfo(optional)	• Pointer to structure of CUGInfo – 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. – Bit 0 - VOICE_EMER_CAT_POLICE_BIT - Police – Bit 1 - VOICE_EMER_CAT_AMBULANCE_BIT - Ambulance – Bit 2 - VOICE_EMER_CAT_FIRE_BRIGADE_BIT- Fire brigade – Bit 3 - VOICE_EMER_CAT_MARINE_GUARD_BIT - Marine guard – Bit 4 - VOICE_EMER_CAT_MOUNTAIN_RESCUE_BIT - Mountain rescue – Bit 5 - VOICE_EMER_CAT_MANUAL_ECALL_BIT - Manual emergency call – Bit 6 - VOICE_EMER_CAT_AUTO_ECALL_BIT - Automatic emergency call – Bit 7 - VOICE_EMER_CAT_SPARE_BIT - Spare bit
pCallPartySubAdd(optional)	• Pointer to structure of calledPartySubAdd – See voice_calledPartySubAdd for more information

pSvc-Type(optional)	• Service Type. – 0x01 - VOICE_DIAL_CALL_SRV_TYPE_AUTOMATIC - Automatic – 0x02 - VOICE_DIAL_CALL_SRV_TYPE_GSM - GSM – 0x03 - VOICE_DIAL_CALL_SRV_TYPE_WCDMA - WCDMA – 0x04 - VOICE_DIAL_CALL_SRV_TYPE_CDMA_AUTOMATIC - CDMA automatic – 0x05 - VOICE_DIAL_CALL_SRV_TYPE_GSM_WCDMA - GSM or WCDMA – 0x06 - VOICE_DIAL_CALL_SRV_TYPE_LTE -LTE

8.655.2 Field Documentation

8.655.2.1 uint8_t pack_voice_SLQSVoiceDialCall_t::callNumber[81]

8.655.2.2 voice_calledPartySubAdd* pack_voice_SLQSVoiceDialCall_t::pCallPartySubAdd

8.655.2.3 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCallType

8.655.2.4 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCLIRType

8.655.2.5 voice_CUGInfo* pack_voice_SLQSVoiceDialCall_t::pCUGInfo

8.655.2.6 uint8_t* pack_voice_SLQSVoiceDialCall_t::pEmergencyCategory

8.655.2.7 uint8_t* pack_voice_SLQSVoiceDialCall_t::pSvcType

8.655.2.8 voice_UUSInfo* pack_voice_SLQSVoiceDialCall_t::pUUSInFo

8.656 pack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.656.1 Detailed Description

Parameters

pCallId	• Unique call identifier for the call that must be ended

8.656.2 Field Documentation

8.656.2.1 uint8_t* pack_voice_SLQSVoiceEndCall_t::pCallId

8.657 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference

Data Fields

- uint8_t reason

- uint8_t * pSvcClass

8.657.1 Detailed Description

This structure contains Voice Get Call Barring Pack Parameters

Parameters

reason	• Call Barring Reason • Values: – 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
pSvcClass	• Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available

8.657.2 Field Documentation

8.657.2.1 uint8_t* pack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

8.657.2.2 uint8_t pack_voice_SLQSVoiceGetCallBarring_t::reason

8.658 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference

Data Fields

- uint8_t Reason
- uint8_t * pSvcClass

8.658.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Pack Parameters

Parameters

Reason	• Call Forwarding Reason • Values: – 0x01 - QMI_VOICE_REASON_FWDREASON_UNCONDITIONAL - Unconditional call forwarding – 0x02 - QMI_VOICE_REASON_FWDREASON_MOBILEBUSY - Forward when the mobile is busy – 0x03 - QMI_VOICE_REASON_FWDREASON_NOREPLY - Forward when there is no reply – 0x04 - QMI_VOICE_REASON_FWDREASON_UNREACHABLE - Forward when the call is unreachable – 0x05 - QMI_VOICE_REASON_FWDREASON_ALLFORWARDING - All forwarding – 0x06 - QMI_VOICE_REASON_FWDREASON_ALLCONDITIONAL - All conditional forwarding
pSvcClass(optional)	• Service Class is a combination (sum) of information class constants • See qaGobiApiTableSupServiceInfoClasses.h for service classes.

8.658.2 Field Documentation

8.658.2.1 uint8_t* pack_voice_SLQSVoiceGetCallForwardingStatus_t::pSvcClass

8.658.2.2 uint8_t pack_voice_SLQSVoiceGetCallForwardingStatus_t::Reason

8.659 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference

Data Fields

- uint8_t callID

8.659.1 Detailed Description

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

Parameters

callID	• Call identifier for the call queried for information.

8.659.2 Field Documentation

8.659.2.1 uint8_t pack_voice_SLQSVoiceGetCallInfo_t::callID

8.660 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

Data Fields

- uint8_t * pSvcClass

8.660.1 Detailed Description

This structure contains Voice Get Call Waiting Response Parameters

Parameters

pSvcClass	• Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available

8.660.2 Field Documentation

8.660.2.1 uint8_t* pack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

8.661 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

8.661.1 Detailed Description

This structure contains Voice Get Configuration Request Parameters

Parameters

pAuto-Answer(optional)	• Indicator to retrieve the Auto Answer Information. – 0x01 - Include auto answer information
pAir-Timer(optional)	• Indicator to retrieve the Air Timer Information. – 0x01 - Include air calls timer count information • Currently Not Supported.
pRoam-Timer(optional)	• Indicator to retrieve the Roam Timer Information. – 0x01 - Include roam calls timer information • Currently Not Supported.

pTTY-Mode(optional)	• Indicator to retrieve the TTY Mode Information. – 0x01 - Include TTY configuration status information
pPrefVoiceS-O(optional)	• Indicator to retrieve the Preferred Voice SO Information. – 0x01 - Include preferred voice configuration status information • Currently Not Supported.
pAMR-Status(optional)	• Indicator to retrieve the AMR Status Information. – 0x01 - Include AMR status information
pPrefVoice-Privacy(optional)	• Indicator to retrieve the Preferred Voice Privacy Information. – 0x01 - Include preferred voice privacy status information
pNamI-D(optional)	• Index of the Number Assignment Module Index (CDMA subscription) to be configured • Range: 0 to 3. • Some modems support only 1 or 2 NAMs. • The NAM Index is valid only when the request contains at least one of Air Timer, Roam Timer, and Preferred Voice SO. • 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. – 0x01 - Include voice domain preference information

Note

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

8.661.2 Field Documentation

8.661.2.1 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAirTimer

8.661.2.2 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAMRStatus

8.661.2.3 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAutoAnswer

8.661.2.4 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pNamID

8.661.2.5 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoicePrivacy

8.661.2.6 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoiceSO

8.661.2.7 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pRoamTimer

8.661.2.8 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pTTYMode

8.661.2.9 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pVoiceDomainPref

8.662 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference

Data Fields

- uint8_t * pRegDTMFEvents
- uint8_t * pRegVoicePrivacyEvents
- uint8_t * pSuppsNotifEvents

8.662.1 Detailed Description

This structure contains parameters of Indication Register Information

Parameters

pRegDTMF-Events(optional)	• Registration Indication For DTMF Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_DTMF_IND indication. – 0x00 - Disable – 0x01 - Enable
pRegVoice-Privacy-Events(optional)	• Registration Indication For Voice Privacy Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_PRIVACY_IND indication. – 0x00 - Disable – 0x01 - Enable
pSuppsNotif-Events(optional)	• Registration Indication For Supplementary Service Notification Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_SUPS_NOTIFICATION_IND indication. – 0x00 - Disable – 0x01 - Enable

Note

One of the optional parameter is mandatory to be present in the request.

8.662.2 Field Documentation

8.662.2.1 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegDTMFEvents

8.662.2.2 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegVoicePrivacyEvents

8.662.2.3 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pSuppsNotifEvents

8.663 pack_voice_SLQSVoiceManageCalls_t Struct Reference

Data Fields

- uint8_t SUPSType

- uint8_t * pCallID

8.663.1 Detailed Description

This structure contains Manage Calls Information.

Parameters

SUPSType	• Supplementary service type during the call. – 0x01 - SUPS_TYPE_RELEASE_HELD_OR_WAITING * Release is held or waiting – 0x02 - SUPS_TYPE_RELEASE_ACTIVE_ACCEPT_HELD_OR_WAITING * Release is active and accepting held or waiting – 0x03 - SUPS_TYPE_HOLD_ACTIVE_ACCEPT_WAITING_OR_HELD * Hold is active and accepting waiting or held – 0x04 - SUPS_TYPE_HOLD_ALL_EXCEPT_SPECIFIED_CALL * Hold all calls except a specified one – 0x05 - SUPS_TYPE_MAKE_CONFERENCE_CALL * Make a conference call – 0x06 - SUPS_TYPE_EXPLICIT_CALL_TRANSFER * Explicit call transfer – 0x07 - SUPS_TYPE_CCBS_ACTIVATION * Activate completion of calls to busy subscriber – 0x08 - SUPS_TYPE_END_ALL_CALLS * End all calls – 0x09 - SUPS_TYPE_RELEASE_SPECIFIED_CALL * Release a specified call
pCallID[Optional]	• Applicable only for SUPSType 0x04, 0x07, and 0x09

8.663.2 Field Documentation

8.663.2.1 uint8_t* pack_voice_SLQSVoiceManageCalls_t::pCallID

8.663.2.2 uint8_t pack_voice_SLQSVoiceManageCalls_t::SUPSType

8.664 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference

Data Fields

- struct voice_USSInfo USSInformation

8.664.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

Parameters

USSInformation	• See voice_USSInfo for more information.

8.664.2 Field Documentation

8.664.2.1 struct voice_USSInfo pack_voice_SLQSVoiceOrigUSSDNoWait_t::USSInformation

8.665 pack_voice_SLQSVoiceSendFlash_t Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t * pFlashPayLd
- uint8_t * pFlashType

8.665.1 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. • Variable Length, NULL terminated.
pFlash-Type(optional)	• Flash type. – 0 - Simple Flash (default) – 1 - Activate answer hold – 2 - Deactivate answer hold

8.665.2 Field Documentation

8.665.2.1 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pCallID

8.665.2.2 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashPayLd

8.665.2.3 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashType

8.666 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

Data Fields

- uint8_t Reason
- uint8_t oldPasswd [4]
- uint8_t newPasswd [4]

- uint8_t newPasswdAgain [4]

8.666.1 Detailed Description

This structure contains Voice Set Call Barring Password Pack Parameters

Parameters

Reason	• Call Barring Reason • Values: – 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
oldPasswd[BARRING_PASSWORD_LENGTH]	• Old password. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.
newPasswd[BARRING_PASSWORD_LENGTH]	• New password. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.
newPasswdAgain[BARRING_PASSWORD_LENGTH]	• New password Again. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.

8.666.2 Field Documentation

8.666.2.1 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswd[4]

8.666.2.2 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswdAgain[4]

8.666.2.3 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::oldPasswd[4]

8.666.2.4 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::Reason

8.667 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

8.667.1 Detailed Description

This structure contains information about the Set Configuration Request Parameters.

Parameters

pAutoAnswer	• Value specified is written to NV_AUTO_ANSWER_I. (optional) • Values: – 0x00 - Disable – 0x01 - Enable
pAirTimerConfig	• Value specified is written to NV_AIR_CNT_I. (optional) • See voice_airTimer for more information
pRoamTimer-Config	• Value specified is written to NV_ROAM_CNT_I. (optional) • See vocie_roamTimer for more information
pTTYMode	• Value specified is written to NV_TTY_I. (optional) • Values: – 0x00 - TTY_MODE_FULL - Full – 0x01 - TTY_MODE_VCO - Voice carry over – 0x02 - TTY_MODE_HCO - Hearing carry over – 0x03 - TTY_MODE_OFF - Off
pPrefVoiceSO	• Value specified is written to NV_PREF_VOICE_SO_I. (optional) • See voice_prefVoiceSO for more information
pPrefVoice-Domain	• Preferred Voice-Domain. (optional) • Values: – 0x00 - VOICE_DOMAIN_PREF_CS_ONLY - Circuit-switched (CS) only – 0x01 - VOICE_DOMAIN_PREF_PS_ONLY - Packet-switched (PS) only – 0x02 - VOICE_DOMAIN_PREF_CS_PREF - CS is preferred, PS is secondary – 0x03 - VOICE_DOMAIN_PREF_PS_PREF - PS is preferred, CS is secondary

Note

One of the optional parameters must be present in the request.

8.667.2 Field Documentation

8.667.2.1 voice_airTimer* pack_voice_SLQSVoiceSetConfig_t::pAirTimerConfig

8.667.2.2 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pAutoAnswer

8.667.2.3 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceDomain

8.667.2.4 voice_prefVoiceSO* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSO

8.667.2.5 voice_roamTimer* pack_voice_SLQSVoiceSetConfig_t::pRoamTimerConfig

8.667.2.6 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pTTYMode

8.668 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference

Data Fields

- uint8_t privacyPref

8.668.1 Detailed Description

This structure contains the preferred voice privacy values.

Parameters

privacyPref	• Voice Privacy Preference – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy

8.668.2 Field Documentation

8.668.2.1 uint8_t pack_voice_SLQSVoiceSetPreferredPrivacy_t::privacyPref

8.669 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

8.669.1 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) – 0x01 - VOICE_SERVICE_ACTIVATE – 0x02 - VOICE_SERVICE_DEACTIVATE – 0x03 - VOICE_SERVICE_REGISTER – 0x04 - VOICE_SERVICE_ERASE
reason	• supplementary service reason values (mandatory) – 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 calls are barred – 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT Outgoing internal calls are barred – 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME Outgoing calls external to home are barred – 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING All incoming calls are barred – 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING Roaming incoming calls are barred – 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
pServiceClass	• Service class is a combination (sum) of information class constants (optional) – 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) – 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) – See voice_callFwdTypeAndPlan for more information

8.669.2 Field Documentation

8.669.2.1 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallBarringPasswd

8.669.2.2 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallForwardingNumber

8.669.2.3 voice_callFwdTypeAndPlan* pack_voice_SLQSVoiceSetSUPSService_t::pCallFwdTypeAndPlan

8.669.2.4 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pServiceClass

8.669.2.5 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pTimerVal

8.669.2.6 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::reason

8.669.2.7 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::voiceSvc

8.670 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t DTMFdigit

8.670.1 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. Start continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • If the call ID value received is 0, no value has been returned by the device

DTMFdigit	• DTMF digit in ASCII.

8.670.2 Field Documentation

8.670.2.1 uint8_t pack_voice_SLQSVoiceStartContDTMF_t::DTMFdigit

8.670.2.2 uint8_t* pack_voice_SLQSVoiceStartContDTMF_t::pCallID

8.671 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference

Data Fields

- uint8_t callID

8.671.1 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. Stop continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • If the call ID value received is 0, no value has been returned by the device

8.671.2 Field Documentation

8.671.2.1 uint8_t pack_voice_SLQSVoiceStopContDTMF_t::callID

8.672 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference

Data Fields

- uint8_t * pEnableNotification

8.672.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

Parameters

pEnable-Notification	• Enable Notification or not

8.672.2 Field Documentation

8.672.2.1 uint8_t* pack_wds_DHCPv4ClientLeaseChange_t::pEnableNotification

8.673 pack_wds_GetDefaultProfile_t Struct Reference

Data Fields

- uint32_t profiletype

8.673.1 Detailed Description

Parameters

profiletype	profile type

8.673.2 Field Documentation

8.673.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

8.674 pack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t type
- uint8_t family

8.674.1 Detailed Description

Parameters

type	profile type • 0 - 3GPP • 1 - 3GPP2
type	profile family • 0 - Embedded • 1 - Tethered

8.674.2 Field Documentation

8.674.2.1 uint8_t pack_wds_GetDefaultProfileNum_t::family

8.674.2.2 uint8_t pack_wds_GetDefaultProfileNum_t::type

8.675 pack_wds_GetDormancyState_t Struct Reference

8.676 pack_wds_GetLastMobileIPError_t Struct Reference

8.677 pack_wds_GetMobileIP_t Struct Reference

8.678 pack_wds_GetMobileIPProfile_t Struct Reference

Data Fields

- uint8_t index

8.678.1 Detailed Description

Parameters

index	mobile ip profile identifier

8.678.2 Field Documentation

8.678.2.1 uint8_t pack_wds_GetMobileIPProfile_t::index

8.679 pack_wds_GetPacketStatistics_t Struct Reference

Data Fields

- uint32_t * pStatMask

8.679.1 Detailed Description

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

8.679.2 Field Documentation

8.679.2.1 uint32_t* pack_wds_GetPacketStatistics_t::pStatMask

8.680 pack_wds_GetPacketStatus_t Struct Reference

Data Fields

- uint32_t statmask

8.680.1 Detailed Description

Parameters

statmask	packet statistics mask

8.680.2 Field Documentation

8.680.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

8.681 pack_wds_GetSessionDuration_t Struct Reference

8.682 pack_wds_RMSetTransferStatistics_t Struct Reference

Data Fields

- rmTrasnferStaticsReq RmTrasnferStaticsReq

8.682.1 Detailed Description

Parameters

rmTrasnfer-StaticsReq	RM Transfer Statistics Indicator

8.682.2 Field Documentation

8.682.2.1 rmTrasnferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq

8.683 pack_wds_SetAutoconnect_t Struct Reference

Data Fields

- uint8_t acsetting
- uint8_t acroamsetting

8.683.1 Detailed Description

auto connect data session parameters.

Parameters

setting	• NDIS autoconnect setting – 0 - Disabled – 1 - Enabled

8.683.2 Field Documentation

8.683.2.1 uint8_t pack_wds_SetAutoconnect_t::acroamsetting

8.683.2.2 uint8_t pack_wds_SetAutoconnect_t::acsetting

8.684 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

8.684.1 Detailed Description

Parameters

profileType	profile type
pdpType	Packet Data Protocol (PDP) type
ipAddress	ip address
primaryDNS	primary dns
secondaryDNS	secondry dns
authentication	authentication type
name	name of the profile
apnname	apn name
username	username of the profile
password	password of profile

8.684.2 Field Documentation

8.684.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

8.684.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

8.684.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

8.684.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

8.684.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

8.684.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

8.684.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

8.684.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

8.684.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

8.684.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

8.685 pack_wds_SetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t type

- uint8_t family
- uint8_t index

8.685.1 Field Documentation

8.685.1.1 uint8_t pack_wds_SetDefaultProfileNum_t::family

8.685.1.2 uint8_t pack_wds_SetDefaultProfileNum_t::index

8.685.1.3 uint8_t pack_wds_SetDefaultProfileNum_t::type

8.686 pack_wds_SetMobileIP_t Struct Reference

Data Fields

- uint32_t mode

8.686.1 Detailed Description

Parameters

mode	• Mobile IP setting – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 2 - Mobile IP only

8.686.2 Field Documentation

8.686.2.1 uint32_t pack_wds_SetMobileIP_t::mode

8.687 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

8.687.1 Detailed Description

Mobile IP parameters information.

Parameters

pSPC	• NULL-terminated string representing six digit service programming code.
pMode	• Mode to be set (optional) – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 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) – Zero - Disabled – NonZero - Enabled
pHA-Authenticator	• MH-HA authenticator calculator (optional) – Zero - Disabled – NonZero - Enabled
pHA2002bis	• MH-HA RFC 2002bis authentication instead of RFC2002 (optional) – Zero - Disabled – NonZero - Enabled

8.687.2 Field Documentation

8.687.2.1 uint8_t* pack_wds_SetMobileIPParameters_t::pHA2002bis

8.687.2.2 uint8_t* pack_wds_SetMobileIPParameters_t::pHAAuthenticator

8.687.2.3 uint32_t* pack_wds_SetMobileIPParameters_t::pMode

8.687.2.4 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegPeriod

8.687.2.5 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegTraffic

8.687.2.6 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryInterval

8.687.2.7 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryLimit

8.687.2.8 char* pack_wds_SetMobileIPParameters_t::pSPC

8.688 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

8.688.1 Detailed Description

Parameters

spc	service programming code string
index	Index of the profile to modify
pEnabled	Enable profile 0-disable nonzero enable
pAddress	Home IPv4 address
pPrimaryHA	Primary home agent IPv4 address
pSecondaryHA	secondary home agent IPv4 address
pRevTunneling	Enable reverse tunneling 0-disable nonzero enable
pNAI	Network access identifier string
pHASPI	Home agent security parameter index
pAAASPI	AAA server security parameter index
pMNHA	MN-HA key string
pMNAAA	MN-AAA key string

8.688.2 Field Documentation

8.688.2.1 uint8_t pack_wds_SetMobileIPProfile_t::index

8.688.2.2 uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI

8.688.2.3 uint32_t* pack_wds_SetMobileIPProfile_t::pAddress

8.688.2.4 uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled

8.688.2.5 uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI

8.688.2.6 int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA

8.688.2.7 int8_t* pack_wds_SetMobileIPProfile_t::pMNHA

8.688.2.8 int8_t* pack_wds_SetMobileIPProfile_t::pNAI

8.688.2.9 uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA

8.688.2.10 uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling

8.688.2.11 uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA

8.688.2.12 int8_t pack_wds_SetMobileIPProfile_t::spc[10]

8.689 pack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

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

8.689.1 Detailed Description

Parameters

ProfileID	• 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
ProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2 – NULL is not allowed
curProfile	• union of 3GPP and 3GPP2 profile

Note

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

8.689.2 Field Documentation

8.689.2.1 wds_profileInfo* pack_wds_SLQSCreateProfile_t::pCurProfile

8.689.2.2 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileId

8.689.2.3 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileType

8.690 pack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileIndex

8.690.1 Detailed Description

Parameters

profileType	profile type
profileIndex	profile index

8.690.2 Field Documentation

8.690.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

8.690.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

8.691 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

8.692 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

8.693 pack_wds_SLQSGetDUNCallInfo_t Struct Reference

Data Fields

- uint32_t Mask
- uint8_t * pReportConnStatus
- transferStatInd * pTransferStatInd
- uint8_t * pReportDormStatus
- uint8_t * pReportDataBearerTech
- uint8_t * pReportChannelRate

8.693.1 Detailed Description

Parameters

Mask	mast bits corresponding to the information requested to 1
pReportConnStatus	Connect Status Indicator
pTransferStatInd	Transfer Statistics Indicator
pReportDormStatus	Dormancy Status Indicator
pReportDataBearerTech	Current Data Bearer Technology Indicator
pReportChannelRate	Channel Rate Indicator

8.693.2 Field Documentation

8.693.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

8.693.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

8.693.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

8.693.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

8.693.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

8.693.2.6 transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

8.694 pack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- uint8_t ProfileId
- uint8_t ProfileType

8.694.1 Detailed Description

Parameters

ProfileID	• 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
ProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2

Note

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

8.694.2 Field Documentation

8.694.2.1 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId

8.694.2.2 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType

8.695 pack_wds_SLQSGetRuntimeSettings_t Struct Reference

Data Fields

- uint32_t * pReqSettings

8.695.1 Detailed Description

Parameters

pReqSettings	Requested Settings (Optional Parameter) • Set bits to 1, corresponding to requested information. All other bits must be set to 0. • If the values are not available, the corresponding TLVs are not returned in the response. • Absence of this mask TLV results in the device returning all of the available information corresponding to bits 0 through 12. • In cases where the information from bit 13 or greater is required, this TLV with all the necessary bits set must be present in the request. • Values – Bit 0 - Profile identifier – Bit 1 - Profile name – Bit 2 - PDP type – Bit 3 - APN name – Bit 4 - DNS address – Bit 5 - UMTS/GPRS granted QoS – Bit 6 - Username – Bit 7 - Authentication Protocol – Bit 8 - IP address – Bit 9 - Gateway info (address and subnet mask) – Bit 10 - PCSCF address using PCO flag – Bit 11 - PCSCF server address list – Bit 12 - PCSCF domain name list – Bit 13 - MTU – Bit 14 - domain name list – Bit 15 - IP family – Bit 16 - IM_CM flag – Bit 17 - Technology name – Bit 18 - Operator reserved PCO

8.695.2 Field Documentation

8.695.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

8.696 pack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint8_t * pProfileId
- uint8_t * pProfileType
- wds_profileInfo curProfile

8.696.1 Detailed Description

Parameters

ProfileID	• 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
ProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2 – NULL is not allowed
curProfile	• union of 3GPP and 3GPP2 profile

Note

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

8.696.2 Field Documentation

8.696.2.1 wds_profileInfo pack_wds_SLQSModifyProfile_t::curProfile

8.696.2.2 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileId

8.696.2.3 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileType

8.697 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

8.697.1 Detailed Description

Parameters

pLTEAttach-Profile	• Optional parameter • LTE Attach Profile – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) • This setting is deprecated on MC/EM74xx

ProfileList	Profile List • an array of 4 profile configurations • Each element points to a single WORD value indicating profile • Optional parameter with possible values – 1 - 16 (MC/EM73xx and before) – 1 - 24 (MC/EM74xx and onwards) • function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
pDefaultPDN-Enabled	• Optional parameter – 0 - disabled – 1 - enabled
p3gppRelease	3GPP release • Optional parameter – 0 - Release_99 – 1 - Release_5 – 2 - Release_6 – 3 - Release_7 – 4 - Release_8 • In 9x30 and onwards – 5 - Release 9 – 6 - Release 10 – 7 - Release 11
pLTEAttach-ProfileList	• pointer to WORD array indicating LTE Attach Profile List – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – Please provide attach profiles in order of decreasing priority in this list.
LTEAttach-ProfileListLen	• Number of element in pLTEAttachProfileList – valid range: 1-24 – This setting is only supported for MC/EM74xx onwards

8.697.2 Field Documentation

8.697.2.1 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

8.697.2.2 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease

8.697.2.3 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled

8.697.2.4 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile

8.697.2.5 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList

8.697.2.6 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pProfileList

8.698 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint8_t IPFamilyPreference

8.698.1 Detailed Description

Parameters

IPFamily-Preference	IP Family preference • PACK_WDS_IPV4 IP Version 4 • PACK_WDS_IPV6 IP Version 6

8.698.2 Field Documentation

8.698.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

8.699 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

8.699.1 Detailed Description

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

8.699.2 Field Documentation

8.699.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

8.699.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

8.699.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearerTechExt

8.699.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

8.699.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

8.699.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

8.699.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

8.699.2.8 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

8.700 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wdsDhcpv4ProfileId * pProfileId

8.700.1 Detailed Description

Parameters

pProfileId	pointer to Profile Id structure

8.700.2 Field Documentation

8.700.2.1 wdsDhcpv4ProfileId* pack_wds_SLQSSGetDHCPv4ClientConfig_t::pProfileId

8.701 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wds_DHCPv4ProfileId * pProfileId
- wds_DHCPv4HWConfig * pHwConfig
- wds_DHCPv4OptionList * pRequestOptionList

8.701.1 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

8.701.2 Field Documentation

8.701.2.1 wds_DHCPv4HWConfig* pack_wds_SLQSSSetDHCPv4ClientConfig_t::pHwConfig

8.701.2.2 wds_DHCPv4ProfileId* pack_wds_SLQSSSetDHCPv4ClientConfig_t::pProfileId

8.701.2.3 wds_DHCPv4OptionList* pack_wds_SLQSSSetDHCPv4ClientConfig_t::pRequestOptionList

8.702 pack_wds_SLQSSSetLoopback_t Struct Reference

Data Fields

- uint8_t loopbackMode
- uint8_t loopbackMultiplier

8.702.1 Detailed Description

Parameters

loopbackMode	• Loopback Mode. – 0 - Disable – 1 - Enable
loopback-Multiplier	• Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.702.2 Field Documentation

8.702.2.1 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMode

8.702.2.2 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMultiplier

8.703 pack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

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

8.703.1 Detailed Description

Parameters

pTech	• Indicates the technology preference – 1 - UMTS – 2 - CDMA – 3 - eMBMS – 4 - Modem Link Label. Modem Link is an interface for transferring data between entities on AP and modem. • optional
pprofileid3gpp	• pointer to 3GPP profile id • optional
pprofileid3gpp2	• pointer to 3GPPs profile id • optional
pAuth	• Authentication type, it can be PAP or CHAP • optional
pUser	• username for authentication process • optional
pPass	• password for authentication process • optional

8.703.2 Field Documentation

8.703.2.1 uint32_t* pack_wds_SLQSStartDataSession_t::pAuth

8.703.2.2 char* pack_wds_SLQSStartDataSession_t::pPass

8.703.2.3 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp

8.703.2.4 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp2

8.703.2.5 uint8_t* pack_wds_SLQSStartDataSession_t::pTech

8.703.2.6 char* pack_wds_SLQSStartDataSession_t::pUser

8.704 pack_wds_SLQSStopDataSession_t Struct Reference

Data Fields

- uint32_t * psid

8.704.1 Detailed Description

Parameters

sid	session id

8.704.2 Field Documentation

8.704.2.1 uint32_t* pack_wds_SLQSStopDataSession_t::psid

8.705 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

8.705.1 Detailed Description

This structure contains the information about the Set Event Report Request parameters.

Parameters

pCurrChannelRateInd	(optional) • Current Channel Rate Indicator. – 0 - Do not report – 1 - Report channel rate when it changes
pTransferStatInd	(optional) • See TrStatInd for more information.
pDataBearerTechInd	(optional) • Data Bearer Technology Indicator. – 0 - Do not report – 1 - Report radio interface used for data transfer when it changes
pDormancyStatusInd	(optional) • Dormancy Status indicator. – 0 - Do not report – 1 - Report traffic channel state of interface used for data connection
pMIPStatusInd	(optional) • MIP Status Indicator. – 0 - Do not report – 1 - Report MIP status

pCurrData-BearerTechInd	(optional) • Current Data Bearer Technology Indicator. – 0 - Do not report – 1 - Report current data bearer technology when it changes
pDataCallStatus-ChangeInd	(optional) • Data Call Status Change Indicator. – 0 - Do not report – 1 - Report data call status change when it changes
pCurrPrefData-SysInd	(optional) • Current Preferred Data System Indicator. – 0 - Do not report – 1 - Report preferred data system when it changes
pEVDOPage-MonPerChange-Ind	(optional) • EV-DO Page Monitor Period Change Indicator. – 0 - Do not report – 1 - Report EV-DO page monitor period change event
pDataSystem-StatusChange-Ind	(optional) • Data System Status Change Indicator. – 0 - Do not report – 1 - Report data system status change event

Note

At least one parameter should be present.

8.705.2 Field Documentation

8.705.2.1 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd

8.705.2.2 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd

8.705.2.3 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd

8.705.2.4 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd

8.705.2.5 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd

8.705.2.6 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd

8.705.2.7 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd

8.705.2.8 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd

8.705.2.9 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd

8.705.2.10 wds_TrStatInd* pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd

8.706 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

- uint8_t contextId
- uint8_t contextType

8.706.1 Detailed Description

Parameters

contextId	Context Identifier
contextType	Context Type 0-3GPP 1-3GPP2

8.706.2 Field Documentation

8.706.2.1 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.706.2.2 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType

8.707 PackCreateProfileOut Struct Reference

Data Fields

- uint8_t ProfileType
- uint8_t ProfileIndex
- uint16_t ExtErrorCode

8.707.1 Field Documentation

8.707.1.1 uint16_t PackCreateProfileOut::ExtErrorCode

8.707.1.2 uint8_t PackCreateProfileOut::ProfileIndex

8.707.1.3 uint8_t PackCreateProfileOut::ProfileType

8.708 packgetDyingGaspCfg Struct Reference

Data Fields

- uint8_t * pDestSMSNum
- uint8_t * pDestSMSContent

8.708.1 Detailed Description

Parameters

pDestSMSNum[-IN]	• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars. • Optional parameter.

pDestSMS-Content[IN]	• SMS COntent as a string of 8 bit ASCII text characters Max 160 chars. • Optional parameter.

8.708.2 Field Documentation

8.708.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

8.708.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

8.709 packgetDyingGaspStatistics Struct Reference

Data Fields

- uint32_t * pTimeStamp
- uint8_t * pSMSAttemptedFlag

8.709.1 Detailed Description

Parameters

TimeStamp[OU-T]	• Time Stamp.
SMSAttempted-Flag[OUT]	• SMS Attempted Flag.

8.709.2 Field Documentation

8.709.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

8.709.2.2 uint32_t* packgetDyingGaspStatistics::pTimeStamp

8.710 PCMparams Struct Reference

Data Fields

- BYTE iFaceTabLen
- BYTE iFaceTab [255]

8.710.1 Detailed Description

This structure contains the PCM parameters.

Parameters

iFaceTabLen	• Number of sets of iface table

iFaceTab	• Physical Interface Parameters • See qaGobiApiTableSwiAudio.h for more information on physical interface parameters

8.710.2 Field Documentation

8.710.2.1 BYTE PCMparams::iFaceTab[255]

8.710.2.2 BYTE PCMparams::iFaceTabLen

8.711 PCSCFFQDNAddress Struct Reference

Data Fields

- WORD fqdnLen
- CHAR fqdnAddr [256]

8.711.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

Parameters

fqdnLen	• length of the received FQDN address
fqdnAddr	• FQDN address(Max 256 characters)

8.711.2 Field Documentation

8.711.2.1 CHAR PCSCFFQDNAddress::fqdnAddr[256]

8.711.2.2 WORD PCSCFFQDNAddress::fqdnLen

8.712 PCSCFFQDNAddressList Struct Reference

Data Fields

- BYTE numInstances
- struct PCSCFFQDNAddress pcsfFQDNAddress [10]

8.712.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

Parameters

numInstances	• Number of FQDN addresses received
pcsfFQDN-Address	• FQDN address information(Max 10 addresses)

8.712.2 Field Documentation

8.712.2.1 BYTE PCSCFFQDNAddressList::numInstances

8.712.2.2 struct PCSCFFQDNAddress PCSCFFQDNAddressList::pcsfFQDNAddress[10]

8.713 PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- BYTE numInstances
- ULONG pscsfIPv4Addr [64]

8.713.1 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)

8.713.2 Field Documentation

8.713.2.1 BYTE PCSCFIPv4ServerAddressList::numInstances

8.713.2.2 ULONG PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

8.714 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

8.714.1 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: – 0x00 - Unknown – 0x01 - GPS – 0x02 - Cell ID – 0x03 - Enhanced cell ID – 0x04 - WiFi – 0x05 - Terrestrial – 0x06 - Terrestrial hybrid – 0x07 - Other
pTimeType	• Defines the time value set in the pTimeStamp parameter. – 0x00 - UTC Time: starting Jan 1, 1970 – 0x01 - GPS Time: starting Jan 6, 1980 – 0x02 - Age: Age of position information

8.714.2 Field Documentation

8.714.2.1 ULONG* PDSPositionData::pAltitudeWrtEllipsoid

8.714.2.2 ULONG* PDSPositionData::pAltitudeWrtSealevel

8.714.2.3 BYTE* PDSPositionData::pHorizontalConfidence

8.714.2.4 ULONG* PDSPositionData::pHorizontalUncCircular

8.714.2.5 ULONGLONG* PDSPositionData::pLatitude

8.714.2.6 ULONGLONG* PDSPositionData::pLongitude

8.714.2.7 BYTE* PDSPositionData::pPositionSource

8.714.2.8 ULONGLONG* PDSPositionData::pTimeStamp

8.714.2.9 BYTE* PDSPositionData::pTimeType

8.714.2.10 BYTE* PDSPositionData::pVerticalConfidence

8.714.2.11 ULONG* PDSPositionData::pVerticalUnc

8.715 PDSPosMethodStateReq Struct Reference

Data Fields

- BYTE * pXtraTimeState
- BYTE * pXtraDataState
- BYTE * pWifiState

8.715.1 Detailed Description

Parameters to Set state of positioning method for a device.

Parameters

pXtraTimeState	• XTRA Time Position Method State. • Values: – 0x00 - Disable – 0x01 - Enable
pXtraDataState	• XTRA Data Position Method State. • Values: – 0x00 - Disable – 0x01 - Enable
Latitude	• WiFi Position Method State • Values: – 0x00 - Disable – 0x01 - Enable

8.715.2 Field Documentation

8.715.2.1 BYTE* PDSPosMethodStateReq::pWifiState

8.715.2.2 BYTE* PDSPosMethodStateReq::pXtraDataState

8.715.2.3 BYTE* PDSPosMethodStateReq::pXtraTimeState

8.716 peerNumberInfo Struct Reference

Data Fields

- BYTE callID
- BYTE numPI
- BYTE numSI
- BYTE numType
- BYTE numPlan
- BYTE numLen
- BYTE number [81]

8.716.1 Detailed Description

This structure contains information for Connected Peer Numbers.

Parameters

callID	• Unique call identifier for the call.

numPI	• Number presentation indicator. – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
numSI	• Number screening indicator. – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
numType	• Number type. – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
numPlan	• Number plan. – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 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.

8.716.2 Field Documentation

8.716.2.1 BYTE peerNumberInfo::callID

8.716.2.2 BYTE peerNumberInfo::number[81]

8.716.2.3 BYTE peerNumberInfo::numLen

8.716.2.4 BYTE peerNumberInfo::numPI

8.716.2.5 BYTE peerNumberInfo::numPlan

8.716.2.6 BYTE peerNumberInfo::numSI

8.716.2.7 BYTE peerNumberInfo::numType

8.717 personalizationStatus Struct Reference

Data Fields

- BYTE numFeatures
- BYTE feature [12]
- BYTE verifyLeft [12]
- BYTE unblockLeft [12]

8.717.1 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: – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization
verifyLeft	• Number of the remaining attempts to verify the personalization feature.
unblockLeft	• Number of the remaining attempts to unblock the personalization feature.

8.717.2 Field Documentation

8.717.2.1 BYTE personalizationStatus::feature[12]

8.717.2.2 BYTE personalizationStatus::numFeatures

8.717.2.3 BYTE personalizationStatus::unblockLeft[12]

8.717.2.4 BYTE personalizationStatus::verifyLeft[12]

8.718 PhyCaAggPcellInfo Struct Reference

Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- int iLTEbandValue
- BYTE TlvPresent

8.718.1 Detailed Description

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

Parameters

pci	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
freq	• Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
dl_bw_value	• Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB for more information.
scell_state	• Scell state Values. • See NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.718.2 Field Documentation

8.718.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggPcellInfo::dl_bw_value

8.718.2.2 int PhyCaAggPcellInfo::freq

8.718.2.3 int PhyCaAggPcellInfo::iLTEbandValue

8.718.2.4 int PhyCaAggPcellInfo::pci

8.718.2.5 BYTE PhyCaAggPcellInfo::TlvPresent

8.719 PhyCaAggScellDlBw Struct Reference

Data Fields

- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- BYTE TlvPresent

8.719.1 Detailed Description

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

Parameters

dl_bw_value	• Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB for more information.

8.719.2 Field Documentation

8.719.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellDlBw::dl_bw_value

8.719.2.2 BYTE PhyCaAggScellDlBw::TlvPresent

8.720 PhyCaAggScellIndex Struct Reference

Data Fields

- BYTE scell_idx
- BYTE TlvPresent

8.720.1 Detailed Description

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

Parameters

scell_idx	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
TlvPresent	• Tlv Present.

8.720.2 Field Documentation

8.720.2.1 BYTE PhyCaAggScellIndex::scell_idx

8.720.2.2 BYTE PhyCaAggScellIndex::TlvPresent

8.721 PhyCaAggScellIndType Struct Reference

Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_SCELL_STATE scell_state
- BYTE TlvPresent

8.721.1 Detailed Description

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

Parameters

pci	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
freq	• Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
scell_state	• Scell state Values. • See NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.721.2 Field Documentation

8.721.2.1 int PhyCaAggScellIndType::freq

8.721.2.2 int PhyCaAggScellIndType::pci

8.721.2.3 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellIndType::scell_state

8.721.2.4 BYTE PhyCaAggScellIndType::TlvPresent

8.722 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

8.722.1 Detailed Description

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

Parameters

pci	• Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
freq	• Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
dl_bw_value	• Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB for more information.

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

TlvPresent	• Tlv Present.

8.722.2 Field Documentation

8.722.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellInfo::dl_bw_value

8.722.2.2 int PhyCaAggScellInfo::freq

8.722.2.3 int PhyCaAggScellInfo::iLTEbandValue

8.722.2.4 int PhyCaAggScellInfo::pci

8.722.2.5 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellInfo::scell_state

8.722.2.6 BYTE PhyCaAggScellInfo::TlvPresent

8.723 PilotSetData Struct Reference

Data Fields

- BYTE NumPilots
- PilotSetParams * pPilotSetInfo

8.723.1 Detailed Description

This structure contains Pilot Set Data

Parameters

NumPilots(IN/OUT)	• Number of Pilot Sets • As input specifies number of sets of parameter pPilotSetInfo for which memory has been assigned • As output specifies the actual number of sets of parameter pPilotSetInfo returned by device
pPilotSetInfo	• Pilot Set Parameters • 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.

8.723.2 Field Documentation

8.723.2.1 BYTE PilotSetData::NumPilots

8.723.2.2 PilotSetParams* PilotSetData::pPilotSetInfo

8.724 PilotSetParams Struct Reference

Data Fields

- ULONG PilotType
- WORD PilotPN
- WORD PilotStrength

8.724.1 Detailed Description

This structure contains Pilot Set parameters

Parameters

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

8.724.2 Field Documentation

8.724.2.1 WORD PilotSetParams::PilotPN

8.724.2.2 WORD PilotSetParams::PilotStrength

8.724.2.3 ULONG PilotSetParams::PilotType

8.725 pktErrRate Struct Reference

Data Fields

- WORD multiplier
- WORD exponent

8.725.1 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)

8.725.2 Field Documentation

8.725.2.1 WORD pktErrRate::exponent

8.725.2.2 WORD pktErrRate::multiplier

8.726 PlmnID Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- BOOL pcsDigit
- BYTE TlvPresent

8.726.1 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: – TRUE - MNC is a three-digit value; for example, a reported value of 90 corresponds to an MNC value of 090 – 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

8.726.2 Field Documentation

8.726.2.1 WORD PlmnID::mcc

8.726.2.2 WORD PlmnID::mnc

8.726.2.3 BOOL PlmnID::pcsDigit

8.726.2.4 BYTE PlmnID::TlvPresent

8.727 PLMNNetworkName Struct Reference

Data Fields

- BYTE numInstance
- PLMNNetworkNameData PLMNNetName [255]

8.727.1 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).

8.727.2 Field Documentation

8.727.2.1 BYTE PLMNNetworkName::numInstance

8.727.2.2 PLMNNetworkNameData PLMNNetworkName::PLMNNetName[255]

8.728 PLMNNetworkNameData Struct Reference

Data Fields

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

8.728.1 Detailed Description

This structure contains PLMN Network Name Data from multiple sources.

Parameters

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

8.728.2 Field Documentation

8.728.2.1 BYTE PLMNNetworkNameData::codingScheme

8.728.2.2 BYTE PLMNNetworkNameData::countryInitials

8.728.2.3 BYTE PLMNNetworkNameData::longName[255]

8.728.2.4 BYTE PLMNNetworkNameData::longNameLen

8.728.2.5 BYTE PLMNNetworkNameData::longNameSpareBits

8.728.2.6 BYTE PLMNNetworkNameData::shortName[255]

8.728.2.7 BYTE PLMNNetworkNameData::shortNameLen

8.728.2.8 BYTE PLMNNetworkNameData::shortNameSpareBits

8.729 Port Struct Reference

Data Fields

- WORD port
- WORD range

8.729.1 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

8.729.2 Field Documentation

8.729.2.1 WORD Port::port

8.729.2.2 WORD Port::range

8.730 precisionDilution_s Struct Reference

Data Fields

- ULONG PDOP
- ULONG HDOP
- ULONG VDOP

8.730.1 Detailed Description

This structure contains Dilution of precision associated with this position.

Parameters

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

VDOP	• Vertical dilution of precision. • Range- 1 (highest accuracy) to 50 (lowest accuracy)

8.730.2 Field Documentation

8.730.2.1 ULONG precisionDilution_s::HDOP

8.730.2.2 ULONG precisionDilution_s::PDOP

8.730.2.3 ULONG precisionDilution_s::VDOP

8.731 PrefImageList Struct Reference

Data Fields

- BYTE listSize
- struct ImageElement listEntries [2]

8.731.1 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) • See ImageElement

8.731.2 Field Documentation

8.731.2.1 struct ImageElement PrefImageList::listEntries[2]

8.731.2.2 BYTE PrefImageList::listSize

8.732 prefVoiceSO Struct Reference

Data Fields

- BYTE namID
- BYTE evrcCapability
- WORD homePageVoiceSO
- WORD homeOrigVoiceSO
- WORD roamOrigVoiceSO

8.732.1 Detailed Description

This structure contains information about the Preferred Voice Service Options.

Parameters

namID	• Index of the NAM(Number Assignment Module) to be configured. • Range 0 to 3. • Some modems support only 1 or 2 NAMs. • 0xFF,if not available.
evrcCapability	• EVRC capability. • Values: – 0x00 - Disable – 0x01 - Enable – 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. • Values: – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 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. • Values: – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 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. • Values: – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 0xFFFF - Not Available

8.732.2 Field Documentation

8.732.2.1 BYTE prefVoiceSO::evrcCapability

8.732.2.2 WORD prefVoiceSO::homeOrigVoiceSO

8.732.2.3 WORD prefVoiceSO::homePageVoiceSO

8.732.2.4 BYTE prefVoiceSO::namID

8.732.2.5 WORD prefVoiceSO::roamOrigVoiceSO

8.733 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

8.733.1 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 – 0x00 - PDP-IP (IPv4) – 0x01 - PDP-PPP – 0x02 - PDP-IPV6 – 0x03 - PDP-IPV4V6
pPdpHdrComp-Type	• PDP header compression type – 0 - PDP header compression is OFF – 1 - Manufacturer preferred compression – 2 - PDP header compression based on RFC 1144 – 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095

pPdpDataCompType	• PDP data compression type – 0 - PDP data compression is OFF – 1 - Manufacturer preferred compression – 2 - V.42BIS data compression – 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 – Bit map that indicates the authentication algorithm preference * Bit 0 - PAP preference · 0 - PAP is never performed · 1 - PAP may be performed * Bit 1 - CHAP preference · 0 - CHAP is never performed · 1 - CHAP may be performed * If more than one bit is set, then the device decides which authentication procedure is performed while setting up the data session. For example, the device may have a policy to select the most secure authentication mechanism.
pIPv4AddrPref	• IPv4 Address Preference
pPcscfAddr-UsingPCO	• P-CSCF Address using PCO Flag – 1 - (TRUE) implies request PCSCF address using PCO – 0 - (FALSE) implies do not request By default, this value is 0
pPdpAccess-ConFlag	• PDP access control flag – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission
pPcscfAddr-UsingDhcp	• P-CSCF address using DHCP – 1 - (TRUE) implies Request PCSCF address using DHCP – 0 - (FALSE) implies do not request By default, value is 0
pImCnFlag	• IM CN flag – 1 - (TRUE) implies request IM CN flag for this profile – 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 – 1 - (TRUE) implies this is secondary profile – 0 - (FALSE) implies this is not secondary profile

pPrimaryID	• PDP context primary ID • 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 – 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 – This enumerated value may be used to indicate the address allocation preference * 0 - NAS signaling is used for address allocation * 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 • If set, the profile can not be used for making data calls • Any data call is failed locally • Values: – 0 - FALSE(default) – 1 - True • 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 • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

pAPNClass	• Optional 1 Byte numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

8.733.2 Field Documentation

8.733.2.1 BYTE* Profile3GPP::pAddrAllocPref

8.733.2.2 BYTE* Profile3GPP::pAPNClass

8.733.2.3 BYTE* Profile3GPP::pAPNDisabledFlag

8.733.2.4 CHAR* Profile3GPP::pAPNName

8.733.2.5 WORD* Profile3GPP::pAPNnameSize

8.733.2.6 BYTE* Profile3GPP::pAuthenticationPref

8.733.2.7 struct GPRSRequestedQoS* Profile3GPP::pGPRSMinimumQoS

8.733.2.8 struct GPRSRequestedQoS* Profile3GPP::pGPRSRequestedQos

8.733.2.9 BYTE* Profile3GPP::pImCnFlag

8.733.2.10 ULONG* Profile3GPP::pIPv4AddrPref

8.733.2.11 USHORT* Profile3GPP::pIPv6AddPref

8.733.2.12 CHAR* Profile3GPP::pPassword

8.733.2.13 WORD* Profile3GPP::pPasswordSize

8.733.2.14 BYTE* Profile3GPP::pPcscfAddrUsingDhcp

8.733.2.15 BYTE* Profile3GPP::pPcscfAddrUsingPCO

8.733.2.16 ULONG* Profile3GPP::pPDNInactivTimeout

8.733.2.17 BYTE* Profile3GPP::pPdpAccessConFlag

8.733.2.18 BYTE* Profile3GPP::pPdpContext

8.733.2.19 BYTE* Profile3GPP::pPdpDataCompType

8.733.2.20 BYTE* Profile3GPP::pPdpHdrCompType

8.733.2.21 BYTE* Profile3GPP::pPDPtype

8.733.2.22 ULONG* Profile3GPP::pPriDNSIPv4AddPref

8.733.2.23 USHORT* Profile3GPP::pPriDNSIPv6addpref

8.733.2.24 **BYTE*** Profile3GPP::pPrimaryID

8.733.2.25 **CHAR*** Profile3GPP::pProfilename

8.733.2.26 **WORD*** Profile3GPP::pProfilenameSize

8.733.2.27 struct **QosClassID*** Profile3GPP::pQosClassID

8.733.2.28 **ULONG*** Profile3GPP::pSecDNSIPv4AddPref

8.733.2.29 **USHORT*** Profile3GPP::pSecDNSIPv6addpref

8.733.2.30 **BYTE*** Profile3GPP::pSecondaryFlag

8.733.2.31 struct **TFTIDParams*** Profile3GPP::pTFTID1Params

8.733.2.32 struct **TFTIDParams*** Profile3GPP::pTFTID2Params

8.733.2.33 struct **UMTSQoS*** Profile3GPP::pUMTSMinQoS

8.733.2.34 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSMinQosSigInd

8.733.2.35 struct **UMTSQoS*** Profile3GPP::pUMTSReqQoS

8.733.2.36 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSReqQoSSigInd

8.733.2.37 **CHAR*** Profile3GPP::pUsername

8.733.2.38 **WORD*** Profile3GPP::pUsernameSize

8.734 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

8.734.1 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 – 1 - (TRUE)implies request DNS addresses from the PDSN – 0 - (FALSE)implies do not request DNS addresses from the PDSN – Default value is 1 (TRUE)
pPppSessClose-TimerDO	• PPP Session Close Timer for DO – 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 – Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down
pAllowLinger	• Allow/disallow lingering of interface – 1 -(TRUE) implies allow lingering – 0 -(FALSE) implies do not allow lingering
pLcpAckTimeout	• LCP ACK Timeout – Value of LCP ACK Timeout in milliseconds
pIpcpAck-Timeout	• IPCP ACK Timeout – Value of IPCP ACK Timeout in milliseconds
pAuthTimeout	• AUTH Timeout – 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 – 1 - PAP – 2 - CHAP – 3 - PAP or CHAP
pUserId	• User ID to be used during data network authentication • maximum length allowed is 127 bytes; • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
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; • maximum length allowed is 127 bytes • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
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 – 0 - Low (Low speed Service Options (SO15) only) – 1 - Medium (SO33 + low R-SCH) – 2 - High (SO33 + high R-SCH) – Default is 2
pAppType	• Application Type: – 0x00000001 - Default Application Type – 0x00000020 - LBS Application Type – 0x00000040 - Tethered Application Type – This parameter is not used while creating/modifying a profile

pDataMode	• Data Mode to use: – 0 - CDMA or HDR (Hybrid 1X/1xEV-DO) – 1 - CDMA Only (1X only) – 2 - HDR Only (1xEV-DO only) – Default is 0
pAppPriority	• Application Priority – Numerical 1 byte value defining the application priority; higher value implies higher priority – This parameter is not used while creating/modifying a profile
pApnString	• String representing the Access Point Name • maximum length allowed is 100 bytes • QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.
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: – 0 - IPv4 PDN Type – 1 - IPv6 PDN Type – 2 - IPv4 or IPv6 PDN Type – 3 - Unspecified PDN Type (implying no preference)
pIsPcscf-AddressNedded	• This boolean value is used to control if PCSCF address is requested from PDSN – 1 -(TRUE) implies request for PCSCF value from the PDSN – 0 -(FALSE) implies do not request for PCSCF value from the PDSN
pPrimaryV4Dns-Address	• IPv4 Primary DNS address – The Primary IPv4 DNS address that can be statically assigned to the UE
pSecondaryV4-DnsAddress	• IPv4 Secondary DNS address – The Secondary IPv4 DNS address that can be statically assigned to the UE
pPriV6Dns-Address	• Primary IPv6 DNS address – The Primary IPv6 DNS address that can be statically assigned to the UE
pSecV6Dns-Address	• Secondary IPv6 DNS address – The Secondary IPv6 DNS address that can be statically assigned to the UE

pRATType	• Optional 1 Byte Flag indicating RAT Type • Values: – 1 - HRPD – 2 - EHRPD – 3 - HRPD_EHRPD • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pAPNEnabled3GPP2	• Optional 1 Byte Flag indicating if the APN is disabled/enabled • If disabled, the profile can not be used for making data calls • Values: – 0 - Disabled – 1 - Enabled(default value) • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pPDNInactivTimeout3GPP2	• Optional 4 Bytes indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
pAPNClass3GPP2	• Optional 1 Byte numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().

8.734.2 Field Documentation

8.734.2.1 BYTE* Profile3GPP2::pAllowLinger

8.734.2.2 BYTE* Profile3GPP2::pAPNClass3GPP2

8.734.2.3 BYTE* Profile3GPP2::pAPNEnabled3GPP2

8.734.2.4 CHAR* Profile3GPP2::pApnString

8.734.2.5 WORD* Profile3GPP2::pApnStringSize

8.734.2.6 BYTE* Profile3GPP2::pAppPriority

8.734.2.7 ULONG* Profile3GPP2::pAppType

8.734.2.8 CHAR* Profile3GPP2::pAuthPassword

8.734.2.9 WORD* Profile3GPP2::pAuthPasswordSize

8.734.2.10 BYTE* Profile3GPP2::pAuthProtocol

8.734.2.11 BYTE* Profile3GPP2::pAuthRetryCount

8.734.2.12 USHORT* Profile3GPP2::pAuthTimeout

8.734.2.13 BYTE* Profile3GPP2::pDataMode

8.734.2.14 BYTE* Profile3GPP2::pDataRate

8.734.2.15 USHORT* Profile3GPP2::pIpcpAckTimeout

8.734.2.16 BYTE* Profile3GPP2::pIpcpCreqRetryCount

8.734.2.17 BYTE* Profile3GPP2::pIsPcscfAddressNedded

8.734.2.18 USHORT* Profile3GPP2::pLcpAckTimeout

8.734.2.19 BYTE* Profile3GPP2::pLcpCreqRetryCount

8.734.2.20 BYTE* Profile3GPP2::pNegoDnsSrvrPref

8.734.2.21 ULONG* Profile3GPP2::pPDNInactivTimeout3GPP2

8.734.2.22 BYTE* Profile3GPP2::pPdnType

8.734.2.23 ULONG* Profile3GPP2::pPppSessCloseTimer1x

8.734.2.24 ULONG* Profile3GPP2::pPppSessCloseTimerDO

8.734.2.25 ULONG* Profile3GPP2::pPrimaryV4DnsAddress

8.734.2.26 USHORT* Profile3GPP2::pPriV6DnsAddress

8.734.2.27 BYTE* Profile3GPP2::pRATType

8.734.2.28 ULONG* Profile3GPP2::pSecondaryV4DnsAddress

8.734.2.29 USHORT* Profile3GPP2::pSecV6DnsAddress

8.734.2.30 CHAR* Profile3GPP2::pUserId

8.734.2.31 WORD* Profile3GPP2::pUserIdSize

8.735 ProfileIdentifier Struct Reference

Data Fields

- BYTE profileType
- BYTE profileIndex

8.735.1 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

8.735.2 Field Documentation

8.735.2.1 BYTE ProfileIdentifier::profileIndex

8.735.2.2 BYTE ProfileIdentifier::profileType

8.736 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

8.736.1 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

8.736.2 Field Documentation

8.736.2.1 WORD protocolSubtypeElement::AccessMac

8.736.2.2 WORD protocolSubtypeElement::AuthProt

8.736.2.3 WORD protocolSubtypeElement::ControlMac

8.736.2.4 WORD protocolSubtypeElement::EncryptProt

8.736.2.5 WORD protocolSubtypeElement::ForwardMac

8.736.2.6 WORD protocolSubtypeElement::IdleState

8.736.2.7 WORD protocolSubtypeElement::KeyExchange

8.736.2.8 WORD protocolSubtypeElement::MultDisc

8.736.2.9 WORD protocolSubtypeElement::PhysicalLayer

8.736.2.10 WORD protocolSubtypeElement::ReverseMac

8.736.2.11 WORD protocolSubtypeElement::SecProt

8.736.2.12 WORD protocolSubtypeElement::VirtStream

8.737 PSDetachReq Struct Reference

Data Fields

- BYTE * pDetachAction

8.737.1 Detailed Description

This structure contains information about the SLQSSwiPSDetach request parameters.

Parameters

pDetachAction[IN]	• Values – 2- Initiates an immediate packet domain detach.

8.737.2 Field Documentation

8.737.2.1 BYTE* PSDetachReq::pDetachAction

8.738 qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- BYTE leapSeconds
- BYTE localTimeOffset
- BYTE daylightSavings

8.738.1 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 – 0x00 - OFF (daylight savings not in effect) – 0x01 - ON (daylight savings in effect)

8.738.2 Field Documentation

8.738.2.1 BYTE qaQmi3Gpp2TimeZone::daylightSavings

8.738.2.2 BYTE qaQmi3Gpp2TimeZone::leapSeconds

8.738.2.3 BYTE qaQmi3Gpp2TimeZone::localTimeOffset

8.739 qaQmiInterfaceInfo Struct Reference

Data Fields

- BYTE qaQmiinstanceid
- eQaQMIService qaQmisvctype
- ULONG v4sessionId
- ULONG v6sessionId

8.739.1 Detailed Description

Structure used to store the service, interface and session information

Parameters

qaQmiinstanceid	• The interface instance ID – 0x00 - PDP instance ID 0 – 0x01 - PDP instance ID 1 – 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

8.739.2 Field Documentation

8.739.2.1 BYTE qaQmiInterfaceInfo::qaQmiinstanceid

8.739.2.2 eQaQMIService qaQmiInterfaceInfo::qaQmisvctype

8.739.2.3 ULONG qaQmiInterfaceInfo::v4sessionId

8.739.2.4 ULONG qaQmiInterfaceInfo::v6sessionId

8.740 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

8.740.1 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 • See servSystem for more information
roamIndicatorVal	• Optional parameter indicating Roaming Indicator value • Values: – 0x00 - Roaming – 0x01 - Home – 0x02 - Flashing – 0x03 and above - Operator defined values
DataSrv-Capabilities	• Optional parameter indicating Data services capability • See dataSrvCapabilities for more information
CurrentPLMN	• Optional parameter indicating Current PLMN • 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 • See roamIndList for more information
defaultRoamInd	• Optional parameter indicating Default Roaming Indicator • Values: – 0x00 - Roaming – 0x01 - Home
Gpp2TimeZone	• Optional parameter indicating 3GPP2 Time Zone • 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 • Values: – 0x00 - No adjustment for Daylight Saving Time – 0x01 - 1 hr adjustment for Daylight Saving Time – 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 • Values: – 0x00 - Concurrent service not available – 0x01 - Concurrent service available

PRLInd	• Optional parameter indicating 3GPP2 PRL Indicator • Values: – 0x00 - System not in PRL – 0x01 - System is in PRL
DTMInd	• Optional parameter indicating Dual Transfer Mode Indication(GSM Only) • Values: – 0x00 - DTM not supported – 0x01 - DTM supported
DetailedSvcInfo	• Optional parameter indicating Detailed service information • See detailSvcInfo for more information
CDMASystem-InfoExt	• Optional parameter indicating CDMA System Info Ext • See CDMASysInfoExt for more information
hdrPersonality	• Optional parameter indicating HDR Personality Information • Values: – 0x00 - Unknown – 0x01 - HRPD – 0x02 - eHRPD
trackAreaCode	• Optional parameter indicating Tracking area code information for LTE
CallBarStatus	• Optional parameter indicating Call Barring Status • See callBarStatus for more information

8.740.2 Field Documentation

8.740.2.1 WORD qaQmiServingSystemParam::BasestationID

8.740.2.2 ULONG qaQmiServingSystemParam::BasestationLatitude

8.740.2.3 ULONG qaQmiServingSystemParam::BasestationLongitude

8.740.2.4 callBarStatus qaQmiServingSystemParam::CallBarStatus

8.740.2.5 BYTE qaQmiServingSystemParam::CDMA_P_Rev

8.740.2.6 CDMASysInfoExt qaQmiServingSystemParam::CDMASystemInfoExt

8.740.2.7 ULONG qaQmiServingSystemParam::CellID

8.740.2.8 BYTE qaQmiServingSystemParam::concSvcInfo

8.740.2.9 currentPLMN qaQmiServingSystemParam::CurrentPLMN

8.740.2.10 dataSrvCapabilities qaQmiServingSystemParam::DataSrvCapabilities

8.740.2.11 BYTE qaQmiServingSystemParam::defaultRoamInd

8.740.2.12 detailSvcInfo qaQmiServingSystemParam::DetailedSvcInfo

8.740.2.13 BYTE qaQmiServingSystemParam::DTMInd

8.740.2.14 qaQmi3Gpp2TimeZone qaQmiServingSystemParam::Gpp2TimeZone

8.740.2.15 BYTE qaQmiServingSystemParam::GppNetworkDSTAdjustment

8.740.2.16 BYTE qaQmiServingSystemParam::GppTimeZone

8.740.2.17 BYTE qaQmiServingSystemParam::hdrPersonality

8.740.2.18 WORD qaQmiServingSystemParam::Lac

8.740.2.19 WORD qaQmiServingSystemParam::NetworkID

8.740.2.20 BYTE qaQmiServingSystemParam::PRLInd

8.740.2.21 BYTE qaQmiServingSystemParam::roamIndicatorVal

8.740.2.22 roamIndList qaQmiServingSystemParam::RoamingIndicatorList

8.740.2.23 servSystem qaQmiServingSystemParam::ServingSystem

8.740.2.24 WORD qaQmiServingSystemParam::SystemID

8.740.2.25 WORD qaQmiServingSystemParam::trackAreaCode

8.741 QmiCbkCatEventStatusReportInd Struct Reference

Data Fields

- BYTE event_Index
- struct CatCommonEventTlv CCETlv [11]

8.741.1 Field Documentation

8.741.1.1 struct CatCommonEventTlv QmiCbkCatEventStatusReportInd::CCETlv[11]

8.741.1.2 BYTE QmiCbkCatEventStatusReportInd::event_Index

8.742 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

8.742.1 Detailed Description

This structure contains Best Available Position

Parameters

status	• Valid values: – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure

xid	Transaction ID that was specified in the Get Best Available Position request.
pLatitude	• Latitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude
pLongitude	• Longitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude
pHorUncCircular	• Horizontal position uncertainty. • Units - Meters
pAltitudeWrt-Ellipsoid	• Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883
pVertUnc	• Vertical uncertainty. • Units - Meters
pTimestampUtc	• UTC timestamp • Units - Milliseconds since Jan. 1, 1970
pTimeUnc	• Time uncertainty. • Units - Milliseconds
pHorUncEllipse-SemiMinor	• Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters
pHorUncEllipse-SemiMajor	• Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
pHorUncEllipse-OrientAzimuth	• Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180

pHorCirConf	• Horizontal circular uncertainty confidence • Units: Precent • Range: 0 to 99
pHorEllpConf	• Horizontal elliptical uncertainty confidence • Units: Precent • Range: 0 to 99
pHorReliability	• 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
pSpeed-Horizontal	• Horizontal speed. • Units - Meters/second
pSpeedUnc	• 3-D Speed uncertainty. • Units - Meters/second.
pAltitudeWrt-MeanSeaLevel	• Altitude With Respect to Sea Level. • Units - Meters
pVertConfidence	• Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99
pVertReliability	• 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
pSpeedVertical	• Vertical speed. • Units - Meters/second
pSpeedVertical-Unc	• Veritical speed • Units: Meters/second

pHeading	• Heading. • Units - Degree • Range 0 to 359.999
pHeadingUnc	• Heading uncertainty. • Units - Degree • 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 – 0x00000001 - Satellites were used to generate the fix – 0x00000002 - Cell towers were used to generate the fix – 0x00000004 - Wi-Fi access points were used to generate the fix – 0x00000008 - Sensors were used to generate the fix – 0x00000010 - Reference Location was used to generate the fix – 0x00000020 - Coarse position injected into the location engine was used to generate the fix – 0x00000040 - AFLT was used to generate the fix – 0x00000080 - GNSS and network-provided measurements were used to generate the fix
-pPrecision-Dilution	• See precisionDilution for more information
-pGpsTime	• See gpsTime for more information

pTimeSrc	• Values – 0 - Invalid time. – 1 - Time is set by the 1X system. – 2 - Time is set by WCDMA/GSM time tagging. – 3 - Time is set by an external injection. – 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite. – 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites. – 6 - Both time of the week and the GPS week number are known. – 7 - Time is set by the position engine after the fix is obtained – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites.
-pSensorData-Usage	• See sensorDataUsage for more information
-pSvUsedforFix	• See svUsedforFix for more information

8.742.2 Field Documentation

8.742.2.1 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtEllipsoid

8.742.2.2 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtMeanSeaLevel

8.742.2.3 gpsTime* QmiCbkLocBestAvailPosInd::pGpsTime

8.742.2.4 ULONG* QmiCbkLocBestAvailPosInd::pHeading

8.742.2.5 ULONG* QmiCbkLocBestAvailPosInd::pHeadingUnc

8.742.2.6 BYTE* QmiCbkLocBestAvailPosInd::pHorCirConf

8.742.2.7 BYTE* QmiCbkLocBestAvailPosInd::pHorEllpConf

8.742.2.8 ULONG* QmiCbkLocBestAvailPosInd::pHorReliability

8.742.2.9 ULONG* QmiCbkLocBestAvailPosInd::pHorUncCircular

8.742.2.10 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseOrientAzimuth

8.742.2.11 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMajor

8.742.2.12 **ULONG*** QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMinor

8.742.2.13 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pLatitude

8.742.2.14 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pLongitude

8.742.2.15 **ULONG*** QmiCbkLocBestAvailPosInd::pMagneticDeviation

8.742.2.16 **precisionDilution*** QmiCbkLocBestAvailPosInd::pPrecisionDilution

8.742.2.17 **sensorDataUsage*** QmiCbkLocBestAvailPosInd::pSensorDataUsage

8.742.2.18 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedHorizontal

8.742.2.19 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedUnc

8.742.2.20 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedVertical

8.742.2.21 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedVerticalUnc

8.742.2.22 **svUsedforFix*** QmiCbkLocBestAvailPosInd::pSvUsedforFix

8.742.2.23 **ULONG*** QmiCbkLocBestAvailPosInd::pTechnologyMask

8.742.2.24 **ULONG*** QmiCbkLocBestAvailPosInd::pTimeSrc

8.742.2.25 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pTimestampUtc

8.742.2.26 **ULONG*** QmiCbkLocBestAvailPosInd::pTimeUnc

8.742.2.27 **BYTE*** QmiCbkLocBestAvailPosInd::pVertConfidence

8.742.2.28 **ULONG*** QmiCbkLocBestAvailPosInd::pVertReliability

8.742.2.29 **ULONG*** QmiCbkLocBestAvailPosInd::pVertUnc

8.742.2.30 **ULONG*** QmiCbkLocBestAvailPosInd::pXid

8.742.2.31 **ULONG** QmiCbkLocBestAvailPosInd::status

8.743 QmiCbkLocCradleMountInd Struct Reference

Data Fields

- ULONG cradleMountConfigStatus

8.743.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

Parameters

cradleMount-ConfigStatus	• Values – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure

8.743.2 Field Documentation

8.743.2.1 ULONG QmiCbkLocCradleMountInd::cradleMountConfigStatus

8.744 QmiCbkLocEngineStateInd Struct Reference

Data Fields

- ULONG engineState

8.744.1 Detailed Description

This structure contains LOC Engine State field.

Parameters

engineState	• Location engine state. • Valid values – 1 - Location engine is on – 2 - Loction engine is off

8.744.2 Field Documentation

8.744.2.1 ULONG QmiCbkLocEngineStateInd::engineState

8.745 QmiCbkLocEventTimeSyncInd Struct Reference

Data Fields

- ULONG timeSyncRefCounter

8.745.1 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.

8.745.2 Field Documentation

8.745.2.1 ULONG QmiCbkLocEventTimeSyncInd::timeSyncRefCounter

8.746 QmiCbkLocInjectPositionInd Struct Reference

Data Fields

- ULONG status

8.746.1 Detailed Description

Contain the parameters passed for SetLocInjectPositionCallback by the device.

Parameters

status	• UTC Position Injection Status • Valid values: – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure

Note

None

8.746.2 Field Documentation

8.746.2.1 ULONG QmiCbkLocInjectPositionInd::status

8.747 QmiCbkLocInjectSensorDataInd Struct Reference

Data Fields

- ULONG injectSensorDataStatus
- ULONG * pOpaqueIdentifier
- BYTE * pAccelSamplesAccepted
- BYTE * pGyroSamplesAccepted
- BYTE * pAccelTempSamplesAccepted
- BYTE * pGyroTempSamplesAccepted

8.747.1 Detailed Description

This structure contains LOC Inject Sensor Data

Parameters

injectSensorDataStatus	• Values – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure
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. • 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. • 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. • 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. • This field is present only if the gyroscope temperature samples were sent in the request.

8.747.2 Field Documentation

8.747.2.1 ULONG QmiCbkLocInjectSensorDataInd::injectSensorDataStatus

8.747.2.2 BYTE* QmiCbkLocInjectSensorDataInd::pAccelSamplesAccepted

8.747.2.3 BYTE* QmiCbkLocInjectSensorDataInd::pAccelTempSamplesAccepted

8.747.2.4 BYTE* QmiCbkLocInjectSensorDataInd::pGyroSamplesAccepted

8.747.2.5 BYTE* QmiCbkLocInjectSensorDataInd::pGyroTempSamplesAccepted

8.747.2.6 ULONG* QmiCbkLocInjectSensorDataInd::pOpaqueIdentifier

8.748 QmiCbkLocInjectTimeInd Struct Reference

Data Fields

- ULONG injectTimeSyncStatus

8.748.1 Detailed Description

This structure contains LOC Inject Time Sync Data Status

Parameters

injectTimeSync-Status	• Values – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure

8.748.2 Field Documentation

8.748.2.1 ULONG QmiCbkLocInjectTimeInd::injectTimeSyncStatus

8.749 QmiCbkLocInjectUTCTimeInd Struct Reference

Data Fields

- ULONG status

8.749.1 Detailed Description

Contain the parameters passed for SetLocInjectUTCTimeCallback by the device.

Parameters

status	• Status of the UTC Time Injection request • Valid values: – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out

Note

None

8.749.2 Field Documentation

8.749.2.1 ULONG QmiCbkLocInjectUTCTimeInd::status

8.750 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

8.750.1 Detailed Description

This structure contains Event Position Report

Parameters

sessionStatus	• Values – 0 - Session was successful – 1 - Session is still in progress; further position reports will be generated until either the fix criteria specified by the client are met or the client response timeout occurs. – 2 - Session failed.. – 3 - Fix request failed because the session timed out. – 4 - Fix request failed because the session was ended by the user. – 5 - Fix request failed due to bad parameters in the request. – 6 - Fix request failed because the phone is offline. – 7 - Fix request failed because the engine is locked
sessionId	• ID of the session that was specified in the Start request • Range - 0 to 255
pLatitude	• Latitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude
pLongitude	• Longitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude
pHorUncCircular	• Horizontal position uncertainty. • Units - Meters

pHorUncEllipse-SemiMinor	• Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters
pHorUncEllipse-SemiMajor	• Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
pHorUncEllipse-OrientAzimuth	• Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180
pHorConfidence	• Horizontal uncertainty confidence. • If both elliptical and horizontal uncertainties are specified in this message, the confidence corresponds to the elliptical uncertainty. • Units - Percentage • Range 0-99
pHorReliability	• 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
pSpeed-Horizontal	• Horizontal speed. • Units - Meters/second
pSpeedUnc	• 3-D Speed uncertainty. • Units - Meters/second.
pAltitudeWrt-Ellipsoid	• Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883
pAltitudeWrt-MeanSeaLevel	• Altitude With Respect to Sea Level. • Units - Meters
pVertUnc	• Vertical uncertainty. • Units - Meters

pVertConfidence	• Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99
pVertReliability	• 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
pSpeedVertical	• Vertical speed. • Units - Meters/second
pHeading	• Heading. • Units - Degree • Range 0 to 359.999
pHeadingUnc	• Heading uncertainty. • Units - Degree • 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 – 0x00000001 - Satellites were used to generate the fix – 0x00000002 - Cell towers were used to generate the fix – 0x00000004 - Wi-Fi access points were used to generate the fix – 0x00000008 - Sensors were used to generate the fix – 0x00000010 - Reference Location was used to generate the fix – 0x00000020 - Coarse position injected into the location engine was used to generate the fix – 0x00000040 - AFLT was used to generate the fix – 0x00000080 - GNSS and network-provided measurements were used to generate the fix
-pPrecision-Dilution	• See precisionDilution for more information
pTimestampUtc	• UTC timestamp • 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. • Units - Seconds
-pGpsTime	• See gpsTime for more information
pTimeUnc	• Time uncertainty. • Units - Milliseconds
pTimeSrc	• Values – 0 - Invalid time. – 1 - Time is set by the 1X system. – 2 - Time is set by WCDMA/GSM time tagging. – 3 - Time is set by an external injection. – 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite. – 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites. – 6 - Both time of the week and the GPS week number are known. – 7 - Time is set by the position engine after the fix is obtained – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites.
-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

8.750.2 Field Documentation

8.750.2.1 BYTE* QmiCbkLocPositionReportInd::pAltitudeAssumed

8.750.2.2 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtEllipsoid

8.750.2.3 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtMeanSeaLevel

8.750.2.4 ULONG* QmiCbkLocPositionReportInd::pFixId

8.750.2.5 gpsTime* QmiCbkLocPositionReportInd::pGpsTime

8.750.2.6 ULONG* QmiCbkLocPositionReportInd::pHeading

8.750.2.7 ULONG* QmiCbkLocPositionReportInd::pHeadingUnc

8.750.2.8 BYTE* QmiCbkLocPositionReportInd::pHorConfidence

8.750.2.9 ULONG* QmiCbkLocPositionReportInd::pHorReliability

8.750.2.10 ULONG* QmiCbkLocPositionReportInd::pHorUncCircular

8.750.2.11 ULONG* QmiCbkLocPositionReportInd::pHorUncEllipseOrientAzimuth

8.750.2.12 ULONG* QmiCbkLocPositionReportInd::pHorUncEllipseSemiMajor

8.750.2.13 ULONG* QmiCbkLocPositionReportInd::pHorUncEllipseSemiMinor

8.750.2.14 ULONGLONG* QmiCbkLocPositionReportInd::pLatitude

8.750.2.15 BYTE* QmiCbkLocPositionReportInd::pLeapSeconds

8.750.2.16 ULONGLONG* QmiCbkLocPositionReportInd::pLongitude

8.750.2.17 ULONG* QmiCbkLocPositionReportInd::pMagneticDeviation

8.750.2.18 precisionDilution* QmiCbkLocPositionReportInd::pPrecisionDilution

8.750.2.19 sensorDataUsage* QmiCbkLocPositionReportInd::pSensorDataUsage

8.750.2.20 ULONG* QmiCbkLocPositionReportInd::pSpeedHorizontal

8.750.2.21 ULONG* QmiCbkLocPositionReportInd::pSpeedUnc

8.750.2.22 ULONG* QmiCbkLocPositionReportInd::pSpeedVertical

8.750.2.23 svUsedforFix* QmiCbkLocPositionReportInd::pSvUsedforFix

8.750.2.24 ULONG* QmiCbkLocPositionReportInd::pTechnologyMask

8.750.2.25 ULONG* QmiCbkLocPositionReportInd::pTimeSrc

8.750.2.26 ULONGLONG* QmiCbkLocPositionReportInd::pTimestampUtc

8.750.2.27 ULONG* QmiCbkLocPositionReportInd::pTimeUnc

8.750.2.28 BYTE* QmiCbkLocPositionReportInd::pVertConfidence

8.750.2.29 ULONG* QmiCbkLocPositionReportInd::pVertReliability

8.750.2.30 ULONG* QmiCbkLocPositionReportInd::pVertUnc

8.750.2.31 BYTE QmiCbkLocPositionReportInd::sessionId

8.750.2.32 ULONG QmiCbkLocPositionReportInd::sessionStatus

8.751 QmiCbkLocSensorStreamingInd Struct Reference

Data Fields

- accelAcceptReady * pAccelAcceptReady
- gyroAcceptReady * pGyroAcceptReady
- accelTempAcceptReady * pAccelTempAcceptReady
- gyroTempAcceptReady * pGyroTempAcceptReady

8.751.1 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

8.751.2 Field Documentation

8.751.2.1 accelAcceptReady* QmiCbkLocSensorStreamingInd::pAccelAcceptReady

8.751.2.2 accelTempAcceptReady* QmiCbkLocSensorStreamingInd::pAccelTempAcceptReady

8.751.2.3 gyroAcceptReady* QmiCbkLocSensorStreamingInd::pGyroAcceptReady

8.751.2.4 gyroTempAcceptReady* QmiCbkLocSensorStreamingInd::pGyroTempAcceptReady

8.752 QmiCbkLocSetExtPowerConfigInd Struct Reference

Data Fields

- ULONG status

8.752.1 Detailed Description

This structure contains LOC Set External Power Config Status

Parameters

status	• Values – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure

8.752.2 Field Documentation

8.752.2.1 ULONG QmiCbkLocSetExtPowerConfigInd::status

8.753 QmiCbkNasLTECphyCaInfo Struct Reference

Data Fields

- PhyCaAggScellIndType sPhyCaAggScellIndType
- PhyCaAggScellDlBw sPhyCaAggScellDlBw
- PhyCaAggScellInfo sPhyCaAggScellInfo
- PhyCaAggPcellInfo sPhyCaAggPcellInfo
- PhyCaAggScellIndex sPhyCaAggScellIndex

8.753.1 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.

8.753.2 Field Documentation

8.753.2.1 PhyCaAggPcellInfo QmiCbkNasLTECphyCaInfo::sPhyCaAggPcellInfo

8.753.2.2 PhyCaAggScellDlBw QmiCbkNasLTECphyCaInfo::sPhyCaAggScellDlBw

8.753.2.3 PhyCaAggScellIndex QmiCbkNasLTECphyCaInfo::sPhyCaAggScellIndex

8.753.2.4 PhyCaAggScellIndType QmiCbkNasLTECphyCaInfo::sPhyCaAggScellIndType

8.753.2.5 PhyCaAggScellInfo QmiCbkNasLTECphyCaInfo::sPhyCaAggScellInfo

8.754 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference

Data Fields

- struct sessionInfoTlv SITlv

8.754.1 Field Documentation

8.754.1.1 struct sessionInfoTlv QmiCbkSwiOmaDmEventStatusReportInd::SITlv

8.755 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference

Data Fields

- struct sessionInfoTlvExt SITlv

8.755.1 Field Documentation

8.755.1.1 struct sessionInfoTlvExt QmiCbkSwiOmaDmEventStatusReportIndExt::SITlv

8.756 QmiCbkTmdMitiLvlRptInd Struct Reference

Data Fields

- _MitigationDevInfo MitigationDevInfo
- BYTE currentMitigationLvl

8.756.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

Parameters

MitigationDevInfo	• See MitigationDevInfo for more information.
currentMitigationLvl	• Current Thermal Mitigation Level

8.756.2 Field Documentation

8.756.2.1 BYTE QmiCbkTmdMitiLvlRptInd::currentMitigationLvl

8.756.2.2 _MitigationDevInfo QmiCbkTmdMitiLvlRptInd::MitigationDevInfo

8.757 QmiCbkWdsStatisticsIndState Struct Reference

Data Fields

- DataULongTlv TxOkConutTlv
- DataULongTlv RxOkConutTlv
- DataULongLongTlv TxOkByteCountTlv
- DataULongLongTlv RxOkByteCountTlv
- DataULongTlv TxDropConutTlv
- DataULongTlv RxDropConutTlv

8.757.1 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.
TxOkByteCountTlv	• Tx Ok Byte Count Packet Tlv Value.
RxOkByteCountTlv	• Rx Ok Byte Count Packet Tlv Value.
TxDropConutTlv	• Tx Drop Count Packet Tlv Value.
RxDropConutTlv	• Rx Drop Count Packet Tlv Value.

8.757.2 Field Documentation

8.757.2.1 DataULongTlv QmiCbkWdsStatisticsIndState::RxDropConutTlv

8.757.2.2 DataULongLongTlv QmiCbkWdsStatisticsIndState::RxOkByteCountTlv

8.757.2.3 DataULongTlv QmiCbkWdsStatisticsIndState::RxOkConutTlv

8.757.2.4 DataULongTlv QmiCbkWdsStatisticsIndState::TxDropConutTlv

8.757.2.5 DataULongLongTlv QmiCbkWdsStatisticsIndState::TxOkByteCountTlv

8.757.2.6 DataULongTlv QmiCbkWdsStatisticsIndState::TxOkConutTlv

8.758 qmifwinfo_s Struct Reference

Data Fields

- union {
  struct fwinfo_s g
  struct slqsfwinfo_s s
  } dev

8.758.1 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

8.758.2 Field Documentation

8.758.2.1 union { ... } qmifwinfo_s::dev

8.758.2.2 struct fwinfo_s qmifwinfo_s::g

8.758.2.3 struct slqsfwinfo_s qmifwinfo_s::s

8.759 QmiNas3GppNetworkInfo Struct Reference

Data Fields

- WORD pMCC
- WORD pMNC
- ULONG pInUse
- ULONG pRoaming
- ULONG pForbidden
- ULONG pPreferred
- CHAR pDesription [255]

8.759.1 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 – 0 – Unknown – 1 – Current serving network – 2 – Not current serving network, available
pRoaming	• Home/Roam Status of the Network – 0 – Unknown – 1 – Home – 2 – Roam
pForbidden	• Is the Network in the forbidden network list – 0 – Unknown – 1 – Forbidden – 2 – Not Forbidden

pPreferred	• Is the Network in the Preferred network list – 0 – Unknown – 1 – Preferred – 2 – Not Preferred
pDesription	• Network Name/Description

8.759.2 Field Documentation

8.759.2.1 CHAR QmiNas3GppNetworkInfo::pDesription[255]

8.759.2.2 ULONG QmiNas3GppNetworkInfo::pForbidden

8.759.2.3 ULONG QmiNas3GppNetworkInfo::pInUse

8.759.2.4 WORD QmiNas3GppNetworkInfo::pMCC

8.759.2.5 WORD QmiNas3GppNetworkInfo::pMNC

8.759.2.6 ULONG QmiNas3GppNetworkInfo::pPreferred

8.759.2.7 ULONG QmiNas3GppNetworkInfo::pRoaming

8.760 QmiNasGetRFBandInfoResp Struct Reference

Data Fields

- struct qmTlvResult results
- BYTE * pInstancesSize
- struct RFBandInfoElements * pRFBandInfoElements

8.760.1 Field Documentation

8.760.1.1 BYTE* QmiNasGetRFBandInfoResp::pInstancesSize

8.760.1.2 struct RFBandInfoElements* QmiNasGetRFBandInfoResp::pRFBandInfoElements

8.760.1.3 struct qmTlvResult QmiNasGetRFBandInfoResp::results

8.761 QmiNasPerformNetworkScanResp Struct Reference

Data Fields

- struct qmTlvResult results
- BYTE * pInstanceSize
- struct QmiNas3GppNetworkInfo * pInstances

8.761.1 Field Documentation

8.761.1.1 struct QmiNas3GppNetworkInfo* QmiNasPerformNetworkScanResp::pInstances

8.761.1.2 BYTE* QmiNasPerformNetworkScanResp::pInstanceSize

8.761.1.3 struct qmTlvResult QmiNasPerformNetworkScanResp::results

8.762 qmiSmsMessageList Struct Reference

Data Fields

- uint32_t messageIndex
- uint32_t messageTag

8.762.1 Detailed Description

Parameters

messageIndex	• Message index of each matched message
messageTag	• Messagetag

8.762.2 Field Documentation

8.762.2.1 uint32_t qmiSmsMessageList::messageIndex

8.762.2.2 uint32_t qmiSmsMessageList::messageTag

8.763 qmiWDSDataBearerTechnology Struct Reference

Data Fields

- uint8_t currentNetwork
- uint32_t ratMask
- uint32_t soMask

8.763.1 Detailed Description

Parameters

currentNetwork	current selected network
Radio	Access Technology (RAT) mask
soMask	Service Option (SO) mask

8.763.2 Field Documentation

8.763.2.1 uint8_t qmiWDSDataBearerTechnology::currentNetwork

8.763.2.2 uint32_t qmiWDSDataBearerTechnology::ratMask

8.763.2.3 uint32_t qmiWDSDataBearerTechnology::soMask

8.764 QmiWdsIpAddressInfo Struct Reference

Data Fields

- ULONG * pIPAddressV4
- USHORT * pIPAddressV6
- BYTE * pIPv6prefixlen

8.764.1 Detailed Description

Parameters

pIPAddressV4[-OUT]	• Current IPv4 address • default value of 0 if not reported by the device.
pIPAddressV6[-OUT]	• Current IPv6 address Space for storing the 8 element array of type USHORT for the IPv6 address is allocated by the application. The IP Address is stored in the user supplied buffer as follows: User buffer: [<U0>..<U7>] IPv6 address from the network: 1234:2A01:....:5678 User buffer contents: U0 corresponds to 1234 U1 corresponds to 2A01 – ----------— – -— – ----------— – -— U7 corresponds to 5678
pIPv6prefixlen[-OUT]	• IPv6 prefix length in number of bits

8.764.2 Field Documentation

8.764.2.1 ULONG* QmiWdsIpAddressInfo::pIPAddressV4

8.764.2.2 USHORT* QmiWdsIpAddressInfo::pIPAddressV6

8.764.2.3 BYTE* QmiWdsIpAddressInfo::pIPv6prefixlen

8.765 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

8.765.1 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 One or more bytes describing the profile. Description may be a user-defined name for the profile. QMI_ERR_ARG_TOO_LONG is returned if profile_name is too long.
pPDPType	• PDP type – 0 – PDP-IP (IPv4) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6 – 0xffffffff - invalid
pAPNName	• Access point name String parameter that is a logical name used to select the GGSN and external packet data network. If value is NULL or omitted, then the subscription default value is requested. 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 – Bit 0 – PAP preference * 0 – PAP is never performed * 1 – PAP may be performed – Bit 1 – CHAP preference * 0 – CHAP is never performed * 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 – 1 – (TRUE) implies request PCSCF address using PCO – 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
mtu	• MTU

pDomainList	• Domain-Name List
pIPFamily-Preference	• IP family – 0x04 – IPV4 ADDR – 0x06 – IPV6 ADDR
pIMCNflag	• IM CN Flag – 0x00 – FALSE – 0x01 – TRUE
pTechnology	• Technology – CDMA – 0x8001 – UMTS – 0x8004
pIPV6Address-Info	• IPV6 Address Information
pIPV6GW-AddressInfo	• IPV6 Gateway Address Information

8.765.2 Field Documentation

8.765.2.1 CHAR* qmiWdsRunTimeSettings::pAPNName

8.765.2.2 ULONG* qmiWdsRunTimeSettings::pAuthentication

8.765.2.3 struct DomainNameList* qmiWdsRunTimeSettings::pDomainList

8.765.2.4 struct GPRSQoS* qmiWdsRunTimeSettings::pGPRSGrantedQoS

8.765.2.5 ULONG* qmiWdsRunTimeSettings::pGWAddressV4

8.765.2.6 BYTE* qmiWdsRunTimeSettings::pIMCNflag

8.765.2.7 ULONG* qmiWdsRunTimeSettings::pIPAddressV4

8.765.2.8 BYTE* qmiWdsRunTimeSettings::pIPFamilyPreference

8.765.2.9 struct IPV6AddressInfo* qmiWdsRunTimeSettings::pIPV6AddrInfo

8.765.2.10 struct IPV6GWAddressInfo* qmiWdsRunTimeSettings::pIPV6GWAddrInfo

8.765.2.11 ULONG* qmiWdsRunTimeSettings::pMtu

8.765.2.12 BYTE* qmiWdsRunTimeSettings::pPCSCFAddrPCO

8.765.2.13 struct PCSCFFQDNAddressList* qmiWdsRunTimeSettings::pPCSCFFQDNAddrList

8.765.2.14 **ULONG*** qmiWdsRunTimeSettings::pPDPType

8.765.2.15 **ULONG*** qmiWdsRunTimeSettings::pPrimaryDNSV4

8.765.2.16 **USHORT*** qmiWdsRunTimeSettings::pPrimaryDNSV6

8.765.2.17 struct **ProfileIdentifier*** qmiWdsRunTimeSettings::pProfileID

8.765.2.18 **CHAR*** qmiWdsRunTimeSettings::pProfileName

8.765.2.19 **ULONG*** qmiWdsRunTimeSettings::pSecondaryDNSV4

8.765.2.20 **USHORT*** qmiWdsRunTimeSettings::pSecondaryDNSV6

8.765.2.21 struct **PCSCFIPv4ServerAddressList*** qmiWdsRunTimeSettings::pServerAddrList

8.765.2.22 **ULONG*** qmiWdsRunTimeSettings::pSubnetMaskV4

8.765.2.23 **WORD*** qmiWdsRunTimeSettings::pTechnology

8.765.2.24 struct **UMTSQoS*** qmiWdsRunTimeSettings::pUMTSGrantedQoS

8.765.2.25 **CHAR*** qmiWdsRunTimeSettings::pUsername

8.766 QosClassID Struct Reference

Data Fields

- BYTE QCI
- ULONG gDlBitRate
- ULONG maxDlBitRate
- ULONG gUlBitRate
- ULONG maxUlBitRate

8.766.1 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

8.766.2 Field Documentation

8.766.2.1 ULONG QosClassID::gDlBitRate

8.766.2.2 ULONG QosClassID::gUlBitRate

8.766.2.3 ULONG QosClassID::maxDlBitRate

8.766.2.4 ULONG QosClassID::maxUlBitRate

8.766.2.5 BYTE QosClassID::QCI

8.767 QosEventInfo Struct Reference

Data Fields

- ULONG * pDataBearer
- ULONG * pPacketsCountTX
- ULONG * pPacketsCountRX
- ULONGLONG * pTotalBytesTX
- ULONGLONG * pTotalBytesRX

8.767.1 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) – 0x00 - Indicates that this field is ignored – 0x01 - CDMA 1X – 0x02 - EV-DO Rev 0 – 0x03 - GPRS – 0x04 - WCDMA – 0x05 - EV-DO Rev A – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - EV-DO Rev A EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x8000 - NULL Bearer – 0xFF - Unknown Technology
pDormancy-Status	• Dormancy status (NULL if not present) – 1 - traffic channel dormant – 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)

8.767.2 Field Documentation

8.767.2.1 ULONG* QosEventInfo::pDataBearer

8.767.2.2 ULONG* QosEventInfo::pPacketsCountRX

8.767.2.3 ULONG* QosEventInfo::pPacketsCountTX

8.767.2.4 ULONGLONG* QosEventInfo::pTotalBytesRX

8.767.2.5 ULONGLONG* QosEventInfo::pTotalBytesTX

8.768 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

8.768.1 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 • Bearer ID or Radio Link Protocol (RLP) ID of the activated flow. • Valid Values - 0 to 16 • 0xFF - Invalid value.

8.768.2 Field Documentation

8.768.2.1 BYTE* QosFlowInfo::pBearerID

8.768.2.2 QosFlowInfoState* QosFlowInfo::pQFlowState

8.768.2.3 swiQosFilter* QosFlowInfo::pRxQFilter[MAX_QOS_FILTER_TLV]

8.768.2.4 swiQosFlow* QosFlowInfo::pRxQFlowGranted

8.768.2.5 swiQosFilter* QosFlowInfo::pTxQFilter[MAX_QOS_FILTER_TLV]

8.768.2.6 swiQosFlow* QosFlowInfo::pTxQFlowGranted

8.769 QosFlowInfoState Struct Reference

Data Fields

- ULONG id
- BYTE isNewFlow
- BYTE state

8.769.1 Detailed Description

This structure contains QoS flow state

Parameters

id	QoS identifier
isNewFlow	• 1 – Newly added flow • 0 – Existing flow
state	This indicates that the flow that was added/modified/deleted: • 0x01 – Flow activated • 0x02 – Flow modified • 0x03 – Flow deleted • 0x04 – Flow suspended • 0x05 – Flow enabled • 0x06 – Flow disabled

8.769.2 Field Documentation

8.769.2.1 ULONG QosFlowInfoState::id

8.769.2.2 BYTE QosFlowInfoState::isNewFlow

8.769.2.3 BYTE QosFlowInfoState::state

8.770 QosMap Struct Reference

Data Fields

- BYTE dscp
- ULONG qos_id
- BYTE state

8.770.1 Detailed Description

This structure contains the SLQSQosDumpMap Information

Parameters

dscp	• Differential Service Code Point(DSCP) value
qos_id	• QoS identifier
state	• QoS Flow state

8.770.2 Field Documentation

8.770.2.1 BYTE QosMap::dscp

8.770.2.2 ULONG QosMap::qos_id

8.770.2.3 BYTE QosMap::state

8.771 RankIndicatorInd Struct Reference

Data Fields

- WORD Count1
- WORD Count2

8.771.1 Field Documentation

8.771.1.1 WORD RankIndicatorInd::Count1

8.771.1.2 WORD RankIndicatorInd::Count2

8.772 readResult Struct Reference

Data Fields

- WORD contentLen
- BYTE content [255+1]

8.772.1 Detailed Description

This structure contains the information for write operation.

Parameters

contentLen	• Number of sets of content.
content[MAX_DESCRIPTION_LENGTH]	• Read content. • The content is the sequence of bytes as read from the card.

8.772.2 Field Documentation

8.772.2.1 BYTE readResult::content[255+1]

8.772.2.2 WORD readResult::contentLen

8.773 readTransparentInfo Struct Reference

Data Fields

- WORD offset
- WORD length

8.773.1 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. • The value 0 is used to read the complete file.

8.773.2 Field Documentation

8.773.2.1 WORD readTransparentInfo::length

8.773.2.2 WORD readTransparentInfo::offset

8.774 redirNumInfo Struct Reference

Data Fields

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

8.774.1 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 – Caller Id
SI	• Number screening indicator. • Values: – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
numType	• Number type. • Values: – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
numPlan	• Number plan. • Values: – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 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.

8.774.2 Field Documentation

8.774.2.1 BYTE redirNumInfo::number[255]

8.774.2.2 BYTE redirNumInfo::numLen

8.774.2.3 BYTE redirNumInfo::numPlan

8.774.2.4 BYTE redirNumInfo::numType

8.774.2.5 BYTE redirNumInfo::PI

8.774.2.6 BYTE redirNumInfo::reason

8.774.2.7 BYTE redirNumInfo::SI

8.775 registerRefresh Struct Reference

Data Fields

- BYTE registerFlag
- BYTE voteForInit
- WORD numFiles
- fileInfo arrfileInfo [255]

8.775.1 Detailed Description

This structure contains paramaters of refresh Information

Parameters

registerFlag	• Flag that indicates whether to register or deregister for refresh indications. Valid values: – 0 - Deregister – 1 - Register
voteForInit	• Flag that indicates whether to vote for the init when there is a refresh. Valid values: – 0 - Client does not vote for initialization – 1 - Client votes for initialization
numFiles	• Number of sets of the following elements: – file_id – path_len – path
arrfileInfo	• Array of file Information structure. • See /ref fileInfo for more information

8.775.2 Field Documentation

8.775.2.1 fileInfo registerRefresh::arrfileInfo[255]

8.775.2.2 WORD registerRefresh::numFiles

8.775.2.3 BYTE registerRefresh::registerFlag

8.775.2.4 BYTE registerRefresh::voteForInit

8.776 remainingRetries Struct Reference

Data Fields

- BYTE verifyLeft
- BYTE unblockLeft

8.776.1 Detailed Description

This structure contains the information about the retries remaining.

Parameters

verifyLeft	• Number of remaining attempts to verify the PIN. • 0xFF, if unavailable.
unblockLeft	• Number of remaining attempts to unblock the PIN. • 0xFF, if unavailable.

Note

This value is returned only when the enable/disable operation has failed. This information is not sent for a hidden key PIN type.

8.776.2 Field Documentation

8.776.2.1 BYTE remainingRetries::unblockLeft

8.776.2.2 BYTE remainingRetries::verifyLeft

8.777 remotePartyName Struct Reference

Data Fields

- BYTE namePI
- BYTE codingScheme
- BYTE nameLen
- BYTE callerName [255]

8.777.1 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. – 0x00 - PRESENTATION_NAME_PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_NAME_PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NAME_UNAVAILABLE - Unavailable presentation – 0x03 - PRESENTATION_NAME_NAME_PRESENTATION_RESTRICTED - Restricted name presentation – 0xFF - Not Available
codingScheme	• Refer to Table10 qaGobiApiTableCodingScheme.h for coding schemes • 0xFF - Not Available
nameLen	• Provides the length of name which follow. • If zero(0) then no further information exists.
callerName[MAX_DESCRIPTION_LENGTH]	• Name in ASCII, NULL ending.

8.777.2 Field Documentation

8.777.2.1 BYTE remotePartyName::callerName[255]

8.777.2.2 BYTE remotePartyName::codingScheme

8.777.2.3 BYTE remotePartyName::nameLen

8.777.2.4 BYTE remotePartyName::namePI

8.778 remotePartyNum Struct Reference

Data Fields

- BYTE presentationInd
- BYTE numLen
- BYTE remPartyNumber [81]

8.778.1 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. – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
numLen	• Provides the length of number which follow. • If zero(0) then no further information exists.
remParty-Number[MAX_-CALL_NO_LEN]	• Array of numbers in ASCII, NULL ending.

8.778.2 Field Documentation

8.778.2.1 BYTE remotePartyNum::numLen

8.778.2.2 BYTE remotePartyNum::presentationInd

8.778.2.3 BYTE remotePartyNum::remPartyNumber[81]

8.779 ReqFieldsList Struct Reference

Data Fields

- BYTE requestFieldsLen
- BYTE requestFields [256]

8.779.1 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. • Array of uint8 is a bitmask where each bit represents a field ID. • Field 0-15 are mandatory, First Bit represents field ID 16, • Starting with the LSB, bit 0 represents Field ID 16, bit 1 represents ID 17.

8.779.2 Field Documentation

8.779.2.1 BYTE ReqFieldsList::requestFields[256]

8.779.2.2 BYTE ReqFieldsList::requestFieldsLen

8.780 RespFieldsList Struct Reference

Data Fields

- BYTE responseFieldsLen
- BYTE responseFields [256]

8.780.1 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. • Format is same as request field.

8.780.2 Field Documentation

8.780.2.1 BYTE RespFieldsList::responseFields[256]

8.780.2.2 BYTE RespFieldsList::responseFieldsLen

8.781 RFBandInfoElements Struct Reference

Data Fields

- BYTE radioInterface
- WORD activeBandClass
- WORD activeChannel
- uint8_t radioInterface
- uint16_t activeBandClass
- uint16_t activeChannel

8.781.1 Detailed Description

This structure contains the RFBandInfo response parameters.

Parameters

radioInterface	• Radio interface technology – See Tables for Radio Interface
activeBandClass	• Active Band Class – See Tables for Band Classes

activeChannel	• Active channel (0 if channel is not relevant to the reported technology)
radioInterface	radio interface technology
activeBandClass	active band class
activeChannel	active channel

8.781.2 Field Documentation

8.781.2.1 WORD RFBandInfoElements::activeBandClass

8.781.2.2 uint16_t RFBandInfoElements::activeBandClass

8.781.2.3 WORD RFBandInfoElements::activeChannel

8.781.2.4 uint16_t RFBandInfoElements::activeChannel

8.781.2.5 BYTE RFBandInfoElements::radioInterface

8.781.2.6 uint8_t RFBandInfoElements::radioInterface

8.782 rmTrasnferStaticsReq Struct Reference

Data Fields

- uint8_t bResetStatistics
- uint32_t ulMask

8.782.1 Detailed Description

Parameters

bResetStatistics	Clear RM statistics
ulMask	Requested statistic bit mask

8.782.2 Field Documentation

8.782.2.1 uint8_t rmTrasnferStaticsReq::bResetStatistics

8.782.2.2 uint32_t rmTrasnferStaticsReq::ulMask

8.783 roamIndList Struct Reference

Data Fields

- BYTE numInstances
- BYTE radioInterface [0x0A]
- BYTE roamIndicator [0x0A]

8.783.1 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 – defaults to zero
radioInterface	• Radio Interface currently in use • Values: – 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE
roamIndicator	• Roaming Indicator • Values: – 0x00 - Roaming – 0x01 - Home

8.783.2 Field Documentation

8.783.2.1 BYTE roamIndList::numInstances

8.783.2.2 BYTE roamIndList::radioInterface[0x0A]

8.783.2.3 BYTE roamIndList::roamIndicator[0x0A]

8.784 RoamingInfo Struct Reference

Data Fields

- BYTE TlvPresent
- BYTE roaming_ind

8.784.1 Field Documentation

8.784.1.1 BYTE RoamingInfo::roaming_ind

8.784.1.2 BYTE RoamingInfo::TlvPresent

8.785 roamTimer Struct Reference

Data Fields

- BYTE namID
- ULONG roamTimerValue

8.785.1 Detailed Description

This structure contains information about the Roam Timer.

Parameters

namID	• Index of the NAM(Number Assignment Module) to be configured. • Range 0 to 3. • Some modems support only 1 or 2 NAMs. • 0xFF,if not available.
roamTimerValue	• Time in minutes. • Cumulative air time is slammed. • 0xFFFFFFFF,if not available.

8.785.2 Field Documentation

8.785.2.1 BYTE roamTimer::namID

8.785.2.2 ULONG roamTimer::roamTimerValue

8.786 RSRPThresh Struct Reference

Data Fields

- BYTE RSRPThresListLen
- SHORT * pRSRPThresList

8.786.1 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 • 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 – RSRP values are applicable only for LTE – RSRP values are measured in dBm, with a range of -44 dBm to -140 dBm – Each RSRP threshold value is a signed byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.786.2 Field Documentation

8.786.2.1 SHORT* RSRPThresh::pRSRPThresList

8.786.2.2 BYTE RSRPThresh::RSRPThresListLen

8.787 rsrqInformation Struct Reference

Data Fields

- INT8 rsrq
- BYTE radioIf

8.787.1 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 – 0x08 - LTE

8.787.2 Field Documentation

8.787.2.1 BYTE rsrqInformation::radioIf

8.787.2.2 INT8 rsrqInformation::rsrq

8.788 RSRQThresh Struct Reference

Data Fields

- BYTE RSRQThresListLen
- INT8 * pRSRQThresList

8.788.1 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 • 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 – RSRQ values are applicable only for LTE – RSRQ values are measured in dBm, with a range of -20 dBm to -3 dBm – Each RSRQ threshold value is a signed byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.788.2 Field Documentation

8.788.2.1 INT8* RSRQThresh::pRSRQThresList

8.788.2.2 BYTE RSRQThresh::RSRQThresListLen

8.789 RSSIThresh Struct Reference

Data Fields

- BYTE RSSIThresListLen
- INT8 * pRSSIThresList

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

8.789.2 Field Documentation

8.789.2.1 INT8* RSSIThresh::pRSSIThresList

8.789.2.2 BYTE RSSIThresh::RSSIThresListLen

8.790 RXAGCList Struct Reference

Data Fields

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

8.790.1 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

8.790.2 Field Documentation

8.790.2.1 WORD* RXAGCList::pRXAIG

8.790.2.2 WORD* RXAGCList::pRXComprSlope

8.790.2.3 WORD* RXAGCList::pRXComprThres

8.790.2.4 WORD* RXAGCList::pRXExpSlope

8.790.2.5 WORD* RXAGCList::pRXExpThres

8.790.2.6 WORD* RXAGCList::pRXStaticGain

8.791 RXAVCList Struct Reference

Data Fields

- WORD * pAVRXAVCSens
- WORD * pAVRXAVCHeadroom

8.791.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXAVCLIST.

Parameters

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

8.791.2 Field Documentation

8.791.2.1 WORD* RXAVCList::pAVRXAVCHeadroom

8.791.2.2 WORD* RXAVCList::pAVRXAVCSens

8.792 rxInfo Struct Reference

Data Fields

- BYTE isRadioTuned

- INT32 rxPower
- INT32 ecio
- INT32 rscp
- INT32 rsrp
- ULONG phase

8.792.1 Detailed Description

This structure contains the Rx Information.

Parameters

isRadioTuned	• Whether Rx is tuned to a channel: – 0x00 - Not tuned – 0x01 - Tuned – 0xFF - Not Available • If the radio is tuned, instantaneous values are set for the signal information fields below. • 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 dbm; valid for CDMA, HDR, GSM, WCDMA, and LTE.
rscp	• Received signal code power in 1/10 dbm. • 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. • When the phase is unknown, 0xFFFFFFFF is used.

8.792.2 Field Documentation

8.792.2.1 INT32 rxInfo::ecio

8.792.2.2 BYTE rxInfo::isRadioTuned

8.792.2.3 ULONG rxInfo::phase

8.792.2.4 INT32 rxInfo::rscp

8.792.2.5 INT32 rxInfo::rsrp

8.792.2.6 INT32 rxInfo::rxPower

8.793 RXPCMIIRFltr Struct Reference

Data Fields

- WORD * pFlag
- WORD * pStageCnt
- BYTE * pStage0Val
- BYTE * pStage1Val
- BYTE * pStage2Val
- BYTE * pStage3Val
- BYTE * pStage4Val

8.793.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXPCMIIRFLTR.

Parameters

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

pStage3Val	• A 20 BYTE sized parameter indicating Stage 3 value – A1 – A2 – B0 – B1 – B2
pStage4Val	• A 20 BYTE sized parameter indicating Stage 4 value – A1 – A2 – B0 – B1 – B2

8.793.2 Field Documentation

8.793.2.1 WORD* RXPCMIIRFltr::pFlag

8.793.2.2 BYTE* RXPCMIIRFltr::pStage0Val

8.793.2.3 BYTE* RXPCMIIRFltr::pStage1Val

8.793.2.4 BYTE* RXPCMIIRFltr::pStage2Val

8.793.2.5 BYTE* RXPCMIIRFltr::pStage3Val

8.793.2.6 BYTE* RXPCMIIRFltr::pStage4Val

8.793.2.7 WORD* RXPCMIIRFltr::pStageCnt

8.794 RxSigInfo Struct Reference

Data Fields

- BYTE rxChainIndex
- BYTE isRadioTuned
- INT32 rxPower
- INT32 rsrp

8.794.1 Detailed Description

This structure contains the parameters for Rx Signal Info.

Parameters

rxChainIndex	• Rx antenna path • Valid Values – 0 - Primary Rx – 1 - Diversity Rx
isRadioTuned	• Rx path is tuned to a channel or Not • Values – 0x00 - Not tuned – 0x01 - Tuned

Note

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

Parameters

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

8.794.2 Field Documentation

8.794.2.1 BYTE RxSigInfo::isRadioTuned

8.794.2.2 INT32 RxSigInfo::rsrp

8.794.2.3 BYTE RxSigInfo::rxChainIndex

8.794.2.4 INT32 RxSigInfo::rxPower

8.795 rxSignalStrengthListElement Struct Reference

Data Fields

- SHORT rxSignalStrength
- BYTE radioIf

8.795.1 Detailed Description

This structure contains the Received Signal Strength Information

Parameters

rxSignalStrength	• Received signal strength in dBm – For CDMA and UMTS, this indicates forward link pilotEc. – For GSM, the received signal strength. – For LTE, this indicates the total received wideband power observed by UE.
radioIf	• Radio interface technology of the signal being radio_if measured – 0x00 - RADIO_IF_NO_SVC - None (no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000Â® 1X – 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE

Note

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

8.795.2 Field Documentation

8.795.2.1 BYTE rxSignalStrengthListElement::radioIf

8.795.2.2 SHORT rxSignalStrengthListElement::rxSignalStrength

8.796 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

8.796.1 Detailed Description

This structure contains the Extra Apn Params

Parameters

apnId	• APN id • ID identifying the APN that the client would like to query the AMBR params
ambr_ul	• APN AMBR uplink • APN AMBR uplink values from 1 kbps to 8640 kbps

ambr_dl	• APN AMBR downlink • APN AMBR downlink values from 1 kbps to 8640 kbps
ambr_ul_ext	• Extended APN AMBR uplink • APN AMBR uplink values from 8700 kbps to 256 Mbps
ambr_dl_ext	• Extended APN AMBR downlink • APN AMBR downlink values from 8700 kbps to 256 Mbps
ambr_ul_ext2	• Second extended APN AMBR uplink • APN AMBR uplink values from 256 Mbps to 65280 Mbps
ambr_dl_ext2	• Second extended APN AMBR downlink • APN AMBR downlink values from 256 Mbps to 65280 Mbps

8.796.2 Field Documentation

8.796.2.1 BYTE sApnExtraParams::ambr_dl

8.796.2.2 BYTE sApnExtraParams::ambr_dl_ext

8.796.2.3 BYTE sApnExtraParams::ambr_dl_ext2

8.796.2.4 BYTE sApnExtraParams::ambr_ul

8.796.2.5 BYTE sApnExtraParams::ambr_ul_ext

8.796.2.6 BYTE sApnExtraParams::ambr_ul_ext2

8.796.2.7 ULONG sApnExtraParams::apnId

8.797 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

8.797.1 Detailed Description

Contain fields in struct satelliteInfo

Parameters

svListLen	• number of sets of the following elements: – validMask – system – gnssSvid – healthStatus – svStatus – svInfoMask – elevation – azimuth – snr
validMask	• Bitmask indicating which of the fields in this TLV are valid. Valid bitmasks: – 0x00000001 - VALID_SYSTEM – 0x00000002 - VALID_GNSS_SVID – 0x00000004 - VALID_HEALTH_STATUS – 0x00000008 - VALID_PROCESS_STATUS – 0x00000010 - VALID_SVINFO_MASK – 0x00000020 - VALID_ELEVATION – 0x00000040 - VALID_AZIMUTH – 0x00000080 - VALID_SNR
system	• Indicates to which constellation this SV belongs. Valid values: – eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite – eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite – eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite – eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite – eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite – 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: – FOR GPS: 1 to 32 – FOR GLONASS: 1 to 32 – FOR SBAS: 120 to 151 – for BDS: 201 to 237
healthStatus	• health status. Range: 0 - 1 – 0 - unhealthy – 1 - healthy

svStatus	• SV process status. Valid values: – eQMI_LOC_SV_STATUS_IDLE (1) - SV is not being actively processed – eQMI_LOC_SV_STATUS_SEARCH (2) - The system is searching for this SV – eQMI_LOC_SV_STATUS_TRACK (3) - SV is being tracked
svInfoMask	• Indicates whether almanac and ephemeris information is available. Valid bitmasks: – 0x01 - SVINFO_HAS_EPHEMERIS – 0x02 - SVINFO_HAS_ALMANAC
elevation	• SV elevation angle. – Units: Degrees – Range: 0 to 90
azimuth	• SV azimuth angle. – Units: Degrees – Range: 0 to 360
snr	• SV signal-to-noise ratio – Units: dB-Hz

Note

None

8.797.2 Field Documentation

8.797.2.1 FLOAT satelliteInfo::azimuth

8.797.2.2 FLOAT satelliteInfo::elevation

8.797.2.3 WORD satelliteInfo::gnssSvId

8.797.2.4 BYTE satelliteInfo::healthStatus

8.797.2.5 FLOAT satelliteInfo::snr

8.797.2.6 BYTE satelliteInfo::svInfoMask

8.797.2.7 BYTE satelliteInfo::svListLen

8.797.2.8 ULONG satelliteInfo::svStatus

8.797.2.9 ULONG satelliteInfo::system

8.797.2.10 ULONG satelliteInfo::validMask

8.798 SccRxInfo Struct Reference

Data Fields

- INT32 rsrq
- SHORT snr
- BYTE numInstances
- RxSigInfo sigInfo [255]
- BYTE TlvPresent

8.798.1 Detailed Description

This structure contains information about the SccRxInfo parameters.

Parameters

rsrq	• Current reference signal • Receive quality in 1/10 dB resolution
snr	• Reference signal signal-to-noise ratio in dB. • Range -10 to 30
numInstances	• Number of sets of the following – rxChainIndex – isRadioTuned – rxPower – rsrp
sigInfo	• See RxSigInfo for more information
TlvPresent	• Tlv Present.

8.798.2 Field Documentation

8.798.2.1 BYTE SccRxInfo::numInstances

8.798.2.2 INT32 SccRxInfo::rsrq

8.798.2.3 RxSigInfo SccRxInfo::sigInfo[255]

8.798.2.4 SHORT SccRxInfo::snr

8.798.2.5 BYTE SccRxInfo::TlvPresent

8.799 sensorData Struct Reference

Data Fields

- ULONG timeOfFirstSample
- BYTE flags
- BYTE sensorDataLen
- WORD timeOffset [64]
- ULONG xAxis [64]
- ULONG yAxis [64]
- ULONG zAxis [64]

8.799.1 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. • The time stamp is in the time reference scale that is used by the sensor time source. • Units - Milliseconds
flags	• Flags to indicate any deviation from the default measurement assumptions. • All unused bits in this field must be set to 0. • Valid bitmasks – 0x01 - Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples – 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 – timeOffset – xAxis – yAxis – zAxis
timeOffset	• Sample time offset • Units - Milliseconds
xAxis	• Sensor x-axis sample. • Units Accelerometer - Meters/seconds square • Units Gyroscope - Radians/second
yAxis	• Sensor Y-axis sample. • Units Accelerometer - Meters/seconds square • Units Gyroscope - Radians/second

xAxis	• Sensor Z-axis sample. • Units Accelerometer - Meters/seconds square • Units Gyroscope - Radians/second

8.799.2 Field Documentation

8.799.2.1 BYTE sensorData::flags

8.799.2.2 BYTE sensorData::sensorDataLen

8.799.2.3 ULONG sensorData::timeOfFirstSample

8.799.2.4 WORD sensorData::timeOffset[64]

8.799.2.5 ULONG sensorData::xAxis[64]

8.799.2.6 ULONG sensorData::yAxis[64]

8.799.2.7 ULONG sensorData::zAxis[64]

8.800 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]

8.800.1 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. • The time stamp is in the time reference scale that is used by the sensor time source. • Units - Milliseconds

flags	• Flags to indicate any deviation from the default measurement assumptions. • All unused bits in this field must be set to 0. • Valid bitmasks – 0x01 - Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples – 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 – timeOffset – xAxis – yAxis – zAxis
timeOffset	• Sample time offset • Units - Milliseconds
xAxis	• Sensor x-axis sample. • Units Accelerometer - Meters/seconds square • Units Gyroscope - Radians/second
yAxis	• Sensor Y-axis sample. • Units Accelerometer - Meters/seconds square • Units Gyroscope - Radians/second
xAxis	• Sensor Z-axis sample. • Units Accelerometer - Meters/seconds square • Units Gyroscope - Radians/second

8.800.2 Field Documentation

8.800.2.1 uint8_t sensorData_t::flags

8.800.2.2 uint8_t sensorData_t::sensorDataLen

8.800.2.3 uint32_t sensorData_t::timeOfFirstSample

8.800.2.4 uint16_t sensorData_t::timeOffset[64]

8.800.2.5 uint32_t sensorData_t::xAxis[64]

8.800.2.6 uint32_t sensorData_t::yAxis[64]

8.800.2.7 uint32_t sensorData_t::zAxis[64]

8.801 sensorDataUsage_s Struct Reference

Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

8.801.1 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

8.801.2 Field Documentation

8.801.2.1 ULONG sensorDataUsage_s::aidingIndicatorMask

8.801.2.2 ULONG sensorDataUsage_s::usageMask

8.802 serialNumbersInfo Struct Reference

Data Fields

- BYTE esnSize
- CHAR * pESNString
- BYTE imeiSize
- CHAR * pIMEIString
- BYTE meidSize
- CHAR * pMEIDString
- BYTE imeiSvnSize
- CHAR * pImeiSvnString

8.802.1 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.

8.802.2 Field Documentation

8.802.2.1 BYTE serialNumbersInfo::esnSize

8.802.2.2 BYTE serialNumbersInfo::imeiSize

8.802.2.3 BYTE serialNumbersInfo::imeiSvnSize

8.802.2.4 BYTE serialNumbersInfo::meidSize

8.802.2.5 CHAR* serialNumbersInfo::pESNString

8.802.2.6 CHAR* serialNumbersInfo::pIMEIString

8.802.2.7 CHAR* serialNumbersInfo::pImeiSvnString

8.802.2.8 CHAR* serialNumbersInfo::pMEIDString

8.803 serviceProviderName Struct Reference

Data Fields

- BYTE displayCondition
- BYTE spnLength
- BYTE spn [255]

8.803.1 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.

8.803.2 Field Documentation

8.803.2.1 BYTE serviceProviderName::displayCondition

8.803.2.2 BYTE serviceProviderName::spn[255]

8.803.2.3 BYTE serviceProviderName::spnLength

8.804 ServingSystemInfo Struct Reference

Data Fields

- BYTE registrationState
- BYTE csAttachState
- BYTE psAttachState
- BYTE selectedNetwork
- BYTE radioInterfaceNo
- BYTE radioInterfaceList [255]
- BYTE hdrPersonality

8.804.1 Detailed Description

This structure will hold the serving system parameters information

Parameters

registrationState	- Registration state of the mobile • 0 - QMI_NAS_NOT_REGISTERED Not registered;mobile is not currently searching for a new network to provide service • 1 - QMI_NAS_REGISTERED Registered with a network • 2 - QMI_NAS_NOT_REGISTERED_SEARCHING Not registered, but mobile is currently searching for a new network to provide service • 3 - QMI_NAS_REGISTRATION_DENIED Registration denied by the visible network • 4 - QMI_NAS_REGISTRATION_UNKNOWN Registration state is unknown
csAttachState	- Circuit Switch domain attach state of the mobile • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached
psAttachState	- Packet domain attach state of the mobile • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached
selectedNetwork	- Type of selected radio access network • 0x00 - Unknown • 0x01 - 3GPP2 network • 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) • 0x00 - None (no service) • 0x01 - cdma2000 1X • 0x02 - cdma2000 HRPD (1xEV-DO) • 0x03 - AMPS • 0x04 - GSM • 0x05 - UMTS • 0x08 - LTE
hdrPersonality	- HDR personality information (valid only for EVDO) • 0x00 - Unknown • 0x01 - HRPD • 0x02 - eHRPD

Note: None

8.804.2 Field Documentation

8.804.2.1 BYTE ServingSystemInfo::csAttachState

8.804.2.2 BYTE ServingSystemInfo::hdrPersonality

8.804.2.3 BYTE ServingSystemInfo::psAttachState

8.804.2.4 BYTE ServingSystemInfo::radioInterfaceList[255]

8.804.2.5 BYTE ServingSystemInfo::radioInterfaceNo

8.804.2.6 BYTE ServingSystemInfo::registrationState

8.804.2.7 BYTE ServingSystemInfo::selectedNetwork

8.805 servSystem Struct Reference

Data Fields

- BYTE regState
- BYTE csAttachState
- BYTE psAttachState
- BYTE selNetwork
- BYTE numRadioInterfaces
- BYTE radioInterface [0x0A]

8.805.1 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 • Values: – 0 - Not Registered; mobile is not currently searching for a new network to provide service – 1 - Registered with a network – 2 - Not registered, but mobile is currently searching for a new network to provide service – 3 - Registration denied by visible network – 4 - Registration state is unknown
csAttachState	• CS Attach State - Circuit-switched domain attach state of the mobile • Values: – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached
psAttachState	• PS Attach State - Packet-switched domain attach state of the mobile • Values: – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached

selNetwork	• Selected Network - Type of selected radio access network • Values: – 0 - Unknown – 1 - 3GPP2 network – 2 - 3GPP network
numRadio-Interfaces	• In Use Radio Interfaces Number – Number of radio interfaces currently in use – defaults to zero
radioInterface	• Radio Interface(s) modem discovered • Values: – 0x00 - RADIO_IF_NO_SVC - None(no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE

8.805.2 Field Documentation

8.805.2.1 BYTE servSystem::csAttachState

8.805.2.2 BYTE servSystem::numRadioInterfaces

8.805.2.3 BYTE servSystem::psAttachState

8.805.2.4 BYTE servSystem::radioInterface[0x0A]

8.805.2.5 BYTE servSystem::regState

8.805.2.6 BYTE servSystem::selNetwork

8.806 sessionInfo Union Reference

Data Fields

- struct omaDmFotaTlv omaDmFota
- struct omaDmConfigTlv omaDmConfig
- struct omaDmNotificationsTlv omaDmNotifications

8.806.1 Detailed Description

This union sessionInfo consist of omaDmFotaTlv, omaDmConfigTlv and omaDmNotificationsTlv, out of which one will be unpacked against pEventFields.

8.806.2 Field Documentation

8.806.2.1 struct omaDmConfigTlv sessionInfo::omaDmConfig

8.806.2.2 struct omaDmFotaTlv sessionInfo::omaDmFota

8.806.2.3 struct omaDmNotificationsTlv sessionInfo::omaDmNotifications

8.807 sessionInfoExt Union Reference

Data Fields

- struct omaDmFotaTlvExt omaDmFota
- struct omaDmConfigTlvExt omaDmConfig

8.807.1 Detailed Description

This union sessionInfo consist of omaDmFotaTlv and omaDmConfigTlv, out of which one will be unpacked against pEventFields.

8.807.2 Field Documentation

8.807.2.1 struct omaDmConfigTlvExt sessionInfoExt::omaDmConfig

8.807.2.2 struct omaDmFotaTlvExt sessionInfoExt::omaDmFota

8.808 sessionInfoTlv Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformation sessionInfo

8.808.1 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

8.808.2 Field Documentation

8.808.2.1 sessionInformation sessionInfoTlv::sessionInfo

8.808.2.2 ULONG sessionInfoTlv::sessionType

8.808.2.3 BYTE sessionInfoTlv::TlvPresent

8.809 sessionInfoTlvExt Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformationExt sessionInfo

8.809.1 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

8.809.2 Field Documentation

8.809.2.1 sessionInformationExt sessionInfoTlvExt::sessionInfo

8.809.2.2 ULONG sessionInfoTlvExt::sessionType

8.809.2.3 BYTE sessionInfoTlvExt::TlvPresent

8.810 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

8.810.1 Detailed Description

This structure contains the SLQSSetAudioPathConfig request parameters.

Parameters

Profile	[Mandatory] • Audio Profile – 0-9

pECMode	[Optional] • AV_EC – 0 - Echo cancellation off – 1 - Handset echo mode – 2 - Headset mode – 3 - Car kit mode – 4 - Speaker Mode
pNSEnable	[Optional] • Noise Suppression – 0 - Noise suppression off – 1 - Noise suppression on
pTXGain	[Optional] • TX Voice volume – 0x0000 - 0xffff
pDTMFTXGain	[Optional] • AV_DTMFTXG – 0x0000 - 0xffff
pCodecSTGain	[Optional] • AV_CODECSTG – 0x0000 - 0xffff
pTXPCMIIRFltr	[Optional] • See TXPCMIIRFltr for more information
pRXPCMIIRFltr	[Optional] • See RXPCMIIRFltr for more information
pRXAVCAGC-Switch	[Optional] • RX AVC/AGC Switch
pTXAVCSwitch	[Optional] • TX AVC Switch
pRXAGCList	[Optional] • See RXAGCList for more information
pRXAVCList	[Optional] • See RXAVCList for more information
pTXAGCList	[Optional] • See TXAGCList for more information

8.810.2 Field Documentation

8.810.2.1 WORD* SetAudioPathConfigReq::pCodecSTGain

8.810.2.2 WORD* SetAudioPathConfigReq::pDTMFTXGain

8.810.2.3 BYTE* SetAudioPathConfigReq::pECMode

8.810.2.4 BYTE* SetAudioPathConfigReq::pNSEnable

8.810.2.5 BYTE SetAudioPathConfigReq::Profile

8.810.2.6 RXAGCList* SetAudioPathConfigReq::pRXAGCList

8.810.2.7 BYTE* SetAudioPathConfigReq::pRXAVCAGCSwitch

8.810.2.8 RXAVCList* SetAudioPathConfigReq::pRXAVCList

8.810.2.9 RXPCMIIRFltr* SetAudioPathConfigReq::pRXPCMIIRFltr

8.810.2.10 TXAGCList* SetAudioPathConfigReq::pTXAGCList

8.810.2.11 BYTE* SetAudioPathConfigReq::pTXAVCSwitch

8.810.2.12 WORD* SetAudioPathConfigReq::pTXGain

8.810.2.13 TXPCMIIRFltr* SetAudioPathConfigReq::pTXPCMIIRFltr

8.811 SetAudioProfileReq Struct Reference

Data Fields

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

8.811.1 Detailed Description

This structure contains the SLQSSetAudioProfile request parameters.

Parameters

Profile	• Audio Profile – 0 - Handset – 1 - Headset – 2 - Car Kit – 3 - Speaker phone – 4 - Auxiliary – 5 - TTY – 6 - Auxiliary external PCM – 7 - Primary external PCM – 8 - External slave PCM – 9 - I2S

EarMute	• Ear Mute Setting – 0 - unmuted – 1 - muted
MicMute	• MIC Mute Setting – 0 - unmuted – 1 - muted
Generator	• Audio Generator – 0 - Voice – 1 - Key Beep – 2 - MIDI
Volume	• Audio Volume Level – 0 to 7

8.811.2 Field Documentation

8.811.2.1 BYTE SetAudioProfileReq::EarMute

8.811.2.2 BYTE SetAudioProfileReq::Generator

8.811.2.3 BYTE SetAudioProfileReq::MicMute

8.811.2.4 BYTE SetAudioProfileReq::Profile

8.811.2.5 BYTE SetAudioProfileReq::Volume

8.812 SetAudioVolTLBConfigReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Volume
- BYTE Item
- WORD VolValue

8.812.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig request parameters

Parameters

Profile	• Audio Profile – 0-9

Generator	• Audio Generator – 0-2
Volume	• Audio Volume Level – 0-7
Item	• Item – 13 - AV_RXVOLDB – 14 - AV_DTMFVOLDB – 15 - AV_PAD
Value	• Value to be set to the volume table

8.812.2 Field Documentation

8.812.2.1 BYTE SetAudioVolTLBConfigReq::Generator

8.812.2.2 BYTE SetAudioVolTLBConfigReq::Item

8.812.2.3 BYTE SetAudioVolTLBConfigReq::Profile

8.812.2.4 BYTE SetAudioVolTLBConfigReq::Volume

8.812.2.5 WORD SetAudioVolTLBConfigReq::VolValue

8.813 SetAudioVolTLBConfigResp Struct Reference

Data Fields

- WORD ResCode

8.813.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig response parameters.

Parameters

ResCode	• Result of requested item

8.813.2 Field Documentation

8.813.2.1 WORD SetAudioVolTLBConfigResp::ResCode

8.814 setCustomSettingV2 Struct Reference

Data Fields

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

8.814.1 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)

8.814.2 Field Documentation

8.814.2.1 CHAR setCustomSettingV2::cust_id[64+1]

8.814.2.2 BYTE setCustomSettingV2::cust_value[8+1]

8.814.2.3 WORD setCustomSettingV2::value_length

8.815 setDyingGaspCfg Struct Reference

Data Fields

- BYTE * pDestSMSNum
- BYTE * pDestSMSContent

8.815.1 Detailed Description

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

Parameters

OUT]	pDestSMSNum[OUT] • SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars. • Optional parameter.
OUT]	pDestSMSContent[OUT] • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars. • Optional parameter.

8.815.2 Field Documentation

8.815.2.1 BYTE* setDyingGaspCfg::pDestSMSContent

8.815.2.2 BYTE* setDyingGaspCfg::pDestSMSNum

8.816 SetIMSSMSConfigReq Struct Reference

Data Fields

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

8.816.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig request parameters.

Parameters

pSMSFormat	• SMS format – 0 - 3GPP – 1 - 3GPP2
pSMSOverIPNw-Ind	• SMS over IP Network Indication Flag – TRUE - Turn on mobile-originated SMS – FALSE - Turn off mobile-originated SMS
pPhoneCtxtURI-Len	• Length of Phone context Universal Resource Identifier to follow
pPhoneCtxtURI	• Phone context universal resource identifier • Length of this string must be specified in pPhoneCtxtURILen parameter

8.816.2 Field Documentation

8.816.2.1 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURI

8.816.2.2 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURILen

8.816.2.3 BYTE* SetIMSSMSConfigReq::pSMSFormat

8.816.2.4 BYTE* SetIMSSMSConfigReq::pSMSOverIPNwInd

8.817 SetIMSSMSConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.817.1 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

8.817.2 Field Documentation

8.817.2.1 BYTE* SetIMSSMSConfigResp::pSettingResp

8.818 SetIMSUserConfigReq Struct Reference

Data Fields

- BYTE * pIMSDomainLen
- BYTE * pIMSDomain

8.818.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig request parameters.

Parameters

pIMSDomainLen	• Length of IMS Domain Name to follow
pIMSDomain	• IMS domain name

8.818.2 Field Documentation

8.818.2.1 BYTE* SetIMSUserConfigReq::pIMSDomain

8.818.2.2 BYTE* SetIMSUserConfigReq::pIMSDomainLen

8.819 SetIMSUserConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.819.1 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

8.819.2 Field Documentation

8.819.2.1 BYTE* SetIMSUserConfigResp::pSettingResp

8.820 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

8.820.1 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 • Values: – True - Enable – False - Disable

pScrAmrEnable	• Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB) • Values: – True - Enable – False - Disable
pScrAmrWb-Enable	• Flag to enable/disable SCR for AMR WB Audio • Values: – True - Enable – False - Disable
pAmrMode	• BitMask for AMR NB modes allowed • Values: – 0x1 - 4.75 kbps – 0x2 - 5.15 kbps – 0x4 - 5.9 kbps – 0x8 - 6.17 kbps – 0x10 - 7.4 kbps – 0x20 - 7.95 kbps – 0x40 - 10.2 kbps – 0x80 - 12.2 kbps
pAmrWBMode	• BitMask for AMR WB modes allowed • Values: – 0x1 - 6.60 kbps – 0x2 - 8.85 kbps – 0x4 - 12.65 kbps – 0x8 - 14.25 kbps – 0x10 - 15.85 kbps – 0x20 - 18.25 kbps – 0x40 - 19.85 kbps – 0x80 - 23.05 kbps – 0x100 - 23.85 kbps
pAmrOctet-Aligned	• Flag to indicate if the octet is aligned for AMR NB Audio • Values: – True - Aligned – False - Not aligned, Bandwidth Efficient mode
pAmrWBOctet-Aligned	• Flag to indicate if the octet is aligned for AMR WB Audio • Values: – True - Aligned – 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.

8.820.2 Field Documentation

8.820.2.1 BYTE* SetIMSVoIPConfigReq::pAmrMode

8.820.2.2 BYTE* SetIMSVoIPConfigReq::pAmrOctetAligned

8.820.2.3 BYTE* SetIMSVoIPConfigReq::pAmrWbEnable

8.820.2.4 WORD* SetIMSVoIPConfigReq::pAmrWBMode

8.820.2.5 BYTE* SetIMSVoIPConfigReq::pAmrWBOctetAligned

8.820.2.6 WORD* SetIMSVoIPConfigReq::pMinSessionExpiryTimer

8.820.2.7 WORD* SetIMSVoIPConfigReq::pRingBackTimer

8.820.2.8 WORD* SetIMSVoIPConfigReq::pRingingTimer

8.820.2.9 WORD* SetIMSVoIPConfigReq::pRTPRTCPInactTimer

8.820.2.10 BYTE* SetIMSVoIPConfigReq::pScrAmrEnable

8.820.2.11 BYTE* SetIMSVoIPConfigReq::pScrAmrWbEnable

8.820.2.12 WORD* SetIMSVoIPConfigReq::pSessionExpiryTimer

8.821 SetIMSVoIPConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.821.1 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

8.821.2 Field Documentation

8.821.2.1 BYTE* SetIMSVoIPConfigResp::pSettingResp

8.822 SetM2MAudioAVCFGReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Device
- BYTE PIFACEId
- PCMparams * pPCMParams

8.822.1 Detailed Description

This structure contains the SLQSSetM2MAudioAVCFG request parameters.

Parameters

Profile	• Audio Profile – 0-5
Device	• ACDB Device • See qaGobiApiTableSwiAudio.h for more information on ACDB Device
PIFACEId	• Physical Interface • See qaGobiApiTableSwiAudio.h for more information on physical interface
pPCMParams	• PCM parameters • See PCMparams for more information

8.822.2 Field Documentation

8.822.2.1 BYTE SetM2MAudioAVCFGReq::Device

8.822.2.2 BYTE SetM2MAudioAVCFGReq::PIFACEId

8.822.2.3 PCMparams* SetM2MAudioAVCFGReq::pPCMParams

8.822.2.4 BYTE SetM2MAudioAVCFGReq::Profile

8.823 SetM2MAudioLPBKReq Struct Reference

Data Fields

- BYTE Enable

8.823.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK request parameters.

Parameters

Enable	• Operation to be performed – 0 - stop – 1 - VOCODER loop – 2 - internal codec loop

8.823.2 Field Documentation

8.823.2.1 BYTE SetM2MAudioLPBKReq::Enable

8.824 SetM2MAudioProfileReq Struct Reference

Data Fields

- BYTE Profile
- BYTE * pEarMute
- BYTE * pMicMute
- BYTE * pGenerator
- BYTE * pVolume
- BYTE * pCwtMute

8.824.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile request parameters.

Parameters

Profile	• Audio Profile Number – 0-5
pEarMute	• Ear Mute – 0 - mute – 1 - unmute
pMicMute	• Mic Mute – 0 - mute – 1 - unmute

pGenerator	• Generator – 0 - voice
pVolume	• Set RX Volume level – 0-5
pCwtMute	• Call Waiting Tone Mute – 0 - Mute – 1 - UnMute

8.824.2 Field Documentation

8.824.2.1 BYTE* SetM2MAudioProfileReq::pCwtMute

8.824.2.2 BYTE* SetM2MAudioProfileReq::pEarMute

8.824.2.3 BYTE* SetM2MAudioProfileReq::pGenerator

8.824.2.4 BYTE* SetM2MAudioProfileReq::pMicMute

8.824.2.5 BYTE SetM2MAudioProfileReq::Profile

8.824.2.6 BYTE* SetM2MAudioProfileReq::pVolume

8.825 SetM2MAudioVolumeReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Level

8.825.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile request parameters.

Parameters

Profile	• Audio Profile Number – 0-5
Generator	• Generator – 0 - voice

Level	• Audio volume level – 0-5

8.825.2 Field Documentation

8.825.2.1 BYTE SetM2MAudioVolumeReq::Generator

8.825.2.2 BYTE SetM2MAudioVolumeReq::Level

8.825.2.3 BYTE SetM2MAudioVolumeReq::Profile

8.826 SetM2MAVMuteReq Struct Reference

Data Fields

- BYTE Profile
- BYTE EarMute
- BYTE MicMute
- BYTE * pCwtMute

8.826.1 Detailed Description

This structure contains the SLQSSetM2MAVMute request parameters.

Parameters

Profile	• Audio Profile Number – 0-5
EarMute	• Ear Mute – 0-1
MicMute	• Mic Mute – 0-1
pCwtMute	[Optional] • Call Waiting Tone Mute – 0-1

8.826.2 Field Documentation

8.826.2.1 BYTE SetM2MAVMuteReq::EarMute

8.826.2.2 BYTE SetM2MAVMuteReq::MicMute

8.826.2.3 BYTE* SetM2MAVMuteReq::pCwtMute

8.826.2.4 BYTE SetM2MAVMuteReq::Profile

8.827 SetM2MSpkrGainReq Struct Reference

Data Fields

- BYTE Profile
- WORD Value

8.827.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain request parameters.

Parameters

Profile	• Audio Profile Number – 0-5
Value	• RX speakerphone gain – 0x0 - 0x7fff

8.827.2 Field Documentation

8.827.2.1 BYTE SetM2MSpkrGainReq::Profile

8.827.2.2 WORD SetM2MSpkrGainReq::Value

8.828 setPINProtection Struct Reference

Data Fields

- BYTE pinID
- BYTE pinOperation
- BYTE pinLength
- BYTE pinValue [255]

8.828.1 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. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN – 4 - Hidden key
pinOperation	• Indicates whether the PIN is enabled or disabled. – 0 - Disable the PIN – 1 - Enable the PIN
pinLength	• Length of the following elements i.e. pin value.
pinValue[MAX_-DESCRIPTION-_LENGTH]	• PIN value. • This value is a sequence of ASCII characters.

8.828.2 Field Documentation

8.828.2.1 BYTE setPINProtection::pinID

8.828.2.2 BYTE setPINProtection::pinLength

8.828.2.3 BYTE setPINProtection::pinOperation

8.828.2.4 BYTE setPINProtection::pinValue[255]

8.829 SetRegMgrConfigReq Struct Reference

Data Fields

- WORD * pPriCSCFPort
- BYTE * pCSCFPortNameLen
- BYTE * pCSCFPortName
- BYTE * pIMSTestMode

8.829.1 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 • Length of this string must be specified in pCSCFPortNameLen parameter
pIMSTestMode	• IMS Test mode Enabled. – TRUE - Enable, no IMS registration – FALSE - Disable, IMS registration is initiated

8.829.2 Field Documentation

8.829.2.1 BYTE* SetRegMgrConfigReq::pCSCFPortName

8.829.2.2 BYTE* SetRegMgrConfigReq::pCSCFPortNameLen

8.829.2.3 BYTE* SetRegMgrConfigReq::pIMSTestMode

8.829.2.4 WORD* SetRegMgrConfigReq::pPriCSCFPort

8.830 SetRegMgrConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.830.1 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

8.830.2 Field Documentation

8.830.2.1 BYTE* SetRegMgrConfigResp::pSettingResp

8.831 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

8.831.1 Detailed Description

This structure contains the Signal Strength reporting thresholds Item information.

Parameters

pCDMARSSI-Thresh	• CDMA RSSI threshold List • See CDMARSSIThresh for more details
pCDMARSSI-Delta	• RSSI delta (in units of 0.1 dBm). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pCDMAECIO-Thresh	• CDMA ECIO Threshold List • See CDMAECIOThresh for more details
pCDMAECIO-Delta	• ECIO delta (in units of 0.1 dB). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.

pHDRRSSI-Thresh	• HDR RSSI Threshold List • See HDRRSSIThresh for more details
pHDRRSSIDelta	• RSSI delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pHDRECIO-Thresh	• HDR ECIO Threshold List • See HDRECIOThresh for more details
pHDRECIODelta	• ECIO delta (in units of 0.1 dB) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pHDRSINR-Thresh	• HDR SINR Threshold List • See HDRSINRThreshold for more details
pHDRSINRDelta	• SINR delta (in units of 1 SINR level) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pHDRIOThresh	• HDR IO Threshold List • See HDRIOThresh for more details
pHDRIODelta	• IO delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pGSMRSSI-Thresh	• GSM RSSI Threshold List • See GSMRSSIThresh for more details
pGSMRSSIDelta	• RSSI delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pWCDMARSSI-Thresh	• WCDMA RSSI Threshold List • See WCDMARSSIThresh for more details
pWCDMARSSI-Delta	• RSSI delta (in units of 0.1 dBm). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pWCDMAECIO-Thresh	• WCDMA ECIO Threshold List • See WCDMAECIOThresh for more details

pWCDMAECIO-Delta	• ECIO delta (in units of 0.1 dB) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pLTERSSI-Thresh	• LTE RSSI Threshold List • See LTERSSIThresh for more details
pLTERSSIDelta	• RSSI delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pLTESNR-Thresh	• LTE SNR Threshold List • See LTESNRThreshold for more details
pLTESNRDelta	• SNR delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pLTERSRQ-Thresh	• LTE RSRQ Threshold List • See LTERSRQThresh for more details
pLTERSRQ-Delta	• RSRQ delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pLTERSRP-Thresh	• LTE RSRP Threshold List • See LTERSRPThresh for more details
pLTERSRPDelta	• RSRP delta (in units of 0.1 dBm). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pLTESigRpt-Config	• LTE Signal Report Config • See LTESigRptConfig for more details
pTDSCDMARS-CPThresh	• TDSCDMA RSCP Threshold List • See TDSCDMARSCPThresh for more details
pTDSCDMARS-CPDelta	• RSCP delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
pTDSCDMARS-SIThresh	• TDSCDMA RSSI Threshold List • See TDSCDMARSSIThresh for more details

pTDSCDMARSSIDelta	• RSSI delta (in dBm) used by TD-SCDMA.
pTDSCDMAECIOThresh	• TDSCDMA ECIO Threshold List • See TDSCDMAECIOThresh for more details
pTDSCDMAECIODelta	• ECIO delta (in dB) used by TD-SCDMA
pTDSCDMASINRThresh	• TDSCDMA SINR Threshold List • See TDSCDMASINRThresh for more details
pTDSCDMASINRDelta	• SINR delta (in dB) used by TD-SCDMA.

8.831.2 Field Documentation

8.831.2.1 WORD* setSignalStrengthInfo::pCDMAECIODelta

8.831.2.2 CDMAECIOThresh* setSignalStrengthInfo::pCDMAECIOThresh

8.831.2.3 WORD* setSignalStrengthInfo::pCDMARSSIDelta

8.831.2.4 CDMARSSIThresh* setSignalStrengthInfo::pCDMARSSIThresh

8.831.2.5 WORD* setSignalStrengthInfo::pGSMRSSIDelta

8.831.2.6 GSMRSSIThresh* setSignalStrengthInfo::pGSMRSSIThresh

8.831.2.7 WORD* setSignalStrengthInfo::pHDRECIODelta

8.831.2.8 HDRECIOThresh* setSignalStrengthInfo::pHDRECIOThresh

8.831.2.9 WORD* setSignalStrengthInfo::pHDRIODelta

8.831.2.10 HDRIOThresh* setSignalStrengthInfo::pHDRIOThresh

8.831.2.11 WORD* setSignalStrengthInfo::pHDRRSSIDelta

8.831.2.12 HDRRSSIThresh* setSignalStrengthInfo::pHDRRSSIThresh

8.831.2.13 WORD* setSignalStrengthInfo::pHDRSINRDelta

8.831.2.14 HDRSINRThreshold* setSignalStrengthInfo::pHDRSINRThresh

8.831.2.15 WORD* setSignalStrengthInfo::pLTERSRPDelta

8.831.2.16 LTERSRPThresh* setSignalStrengthInfo::pLTERSRPThresh

8.831.2.17 WORD* setSignalStrengthInfo::pLTERSRQDelta

8.831.2.18 **LTERSRQThresh**∗ setSignalStrengthInfo::pLTERSRQThresh

8.831.2.19 **WORD**∗ setSignalStrengthInfo::pLTERSSIDelta

8.831.2.20 **LTERSSIThresh**∗ setSignalStrengthInfo::pLTERSSIThresh

8.831.2.21 **LTESigRptConfig**∗ setSignalStrengthInfo::pLTESigRptConfig

8.831.2.22 **WORD**∗ setSignalStrengthInfo::pLTESNRDelta

8.831.2.23 **LTESNRThreshold**∗ setSignalStrengthInfo::pLTESNRThresh

8.831.2.24 **ULONG**∗ setSignalStrengthInfo::pTDSCDMAECIODelta

8.831.2.25 **TDSCDMAECIOThresh**∗ setSignalStrengthInfo::pTDSCDMAECIOThresh

8.831.2.26 **WORD**∗ setSignalStrengthInfo::pTDSCDMARSCPDelta

8.831.2.27 **TDSCDMARSCPThresh**∗ setSignalStrengthInfo::pTDSCDMARSCPThresh

8.831.2.28 **ULONG**∗ setSignalStrengthInfo::pTDSCDMARSSIDelta

8.831.2.29 **TDSCDMARSSIThresh**∗ setSignalStrengthInfo::pTDSCDMARSSIThresh

8.831.2.30 **ULONG**∗ setSignalStrengthInfo::pTDSCDMASINRDelta

8.831.2.31 **TDSCDMASINRThresh**∗ setSignalStrengthInfo::pTDSCDMASINRThresh

8.831.2.32 **WORD**∗ setSignalStrengthInfo::pWCDMAECIODelta

8.831.2.33 **WCDMAECIOThresh**∗ setSignalStrengthInfo::pWCDMAECIOThresh

8.831.2.34 **WORD**∗ setSignalStrengthInfo::pWCDMARSSIDelta

8.831.2.35 **WCDMARSSIThresh**∗ setSignalStrengthInfo::pWCDMARSSIThresh

8.832 SetSIPConfigReq Struct Reference

Data Fields

- WORD ∗ pSIPLocalPort
- ULONG ∗ pTimerSIPReg
- ULONG ∗ pSubscribeTimer
- ULONG ∗ pTimerT1
- ULONG ∗ pTimerT2
- ULONG ∗ pTimerTf
- BYTE ∗ pSigCompEnabled

8.832.1 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 – TRUE - Enable – FALSE - Disable

8.832.2 Field Documentation

8.832.2.1 BYTE* SetSIPConfigReq::pSigCompEnabled

8.832.2.2 WORD* SetSIPConfigReq::pSIPLocalPort

8.832.2.3 ULONG* SetSIPConfigReq::pSubscribeTimer

8.832.2.4 ULONG* SetSIPConfigReq::pTimerSIPReg

8.832.2.5 ULONG* SetSIPConfigReq::pTimerT1

8.832.2.6 ULONG* SetSIPConfigReq::pTimerT2

8.832.2.7 ULONG* SetSIPConfigReq::pTimerTf

8.833 SetSIPConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.833.1 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

8.833.2 Field Documentation

8.833.2.1 BYTE* SetSIPConfigResp::pSettingResp

8.834 sGetDeviceSeriesResult Struct Reference

Data Fields

- enum eGobiDeviceSeries eDevice
- ULONG uResult

8.834.1 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

8.834.2 Field Documentation

8.834.2.1 enum eGobiDeviceSeries sGetDeviceSeriesResult::eDevice

8.834.2.2 ULONG sGetDeviceSeriesResult::uResult

8.835 sidNid Struct Reference

Data Fields

- WORD nid
- WORD sid

8.835.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

nid	• Network ID
sid	• System ID

8.835.2 Field Documentation

8.835.2.1 WORD sidNid::nid

8.835.2.2 WORD sidNid::sid

8.836 sigInfo Struct Reference

Data Fields

- RSSIThresh * pRSSIThresh
- ECIOThresh * pECIOThresh
- HDRSINRThresh * pHDRSINRThresh
- LTESNRThresh * pLTESNRThresh
- IOThresh * pIOThresh
- RSRQThresh * pRSRQThresh
- RSRPThresh * pRSRPThresh
- LTESigRptCfg * pLTESigRptCfg
- TDSCDMASINRCONFThresh * pTDSCDMASINRCONFThresh

8.836.1 Detailed Description

This structure contains the 3gpp Configuration Item information.

Parameters

pRSSIThresh	• RSSI threshold List • See RSSIThresh for more details
pECIOThresh	• ECIO Threshold List • See ECIOThresh for more details
pHDRSINR-Thresh	• HDR SINR Threshold List • See HDRSINRThresh for more details
pLTESNR-Thresh	• LTE SNR Threshold List • See LTESNRThresh for more details
pIOThresh	• IO Threshold List • See IOThresh for more details
pRSRQThresh	• RSRQ Threshold List • See RSRQThresh for more details
pRSRPThresh	• RSRP Threshold List • See RSRPThresh for more details

pLTESigRptCfg	• LTE signal report config • See LTESigRptCfg for more details
pTDSCDMASIN-RCONFThresh	• TD-SCDMA SINR Threshold List • See TDSCDMASINRCONFThresh for more details

8.836.2 Field Documentation

8.836.2.1 ECIOThresh∗ sigInfo::pECIOThresh

8.836.2.2 HDRSINRThresh∗ sigInfo::pHDRSINRThresh

8.836.2.3 IOThresh∗ sigInfo::pIOThresh

8.836.2.4 LTESigRptCfg∗ sigInfo::pLTESigRptCfg

8.836.2.5 LTESNRThresh∗ sigInfo::pLTESNRThresh

8.836.2.6 RSRPThresh∗ sigInfo::pRSRPThresh

8.836.2.7 RSRQThresh∗ sigInfo::pRSRQThresh

8.836.2.8 RSSIThresh∗ sigInfo::pRSSIThresh

8.836.2.9 TDSCDMASINRCONFThresh∗ sigInfo::pTDSCDMASINRCONFThresh

8.837 signalInfo Struct Reference

Data Fields

- BYTE signalType
- BYTE alertPitch
- BYTE signal

8.837.1 Detailed Description

This structure contains Signal Information

Parameters

signalType	• Call identifier for the call.
alertPitch	• Signal Information
signal	• Caller ID Information

8.837.2 Field Documentation

8.837.2.1 BYTE signalInfo::alertPitch

8.837.2.2 BYTE signalInfo::signal

8.837.2.3 BYTE signalInfo::signalType

8.838 SignalStrengthDataType Struct Reference

Data Fields

- BYTE thresholdsSize
- INT8 thresholds [5]

8.838.1 Field Documentation

8.838.1.1 INT8 SignalStrengthDataType::thresholds[5]

8.838.1.2 BYTE SignalStrengthDataType::thresholdsSize

8.839 slot_t Struct Reference

Data Fields

- uint32_t uPhyCardStatus
- uint32_t uPhySlotStatus
- uint8_t bLogicalSlot
- uint8_t bICCIDLength
- uint8_t bICCID [255]

8.839.1 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. – 0x00 - Unknown. – 0x01 - Absent. – 0x02 - Present.
uPhySlotStatus	• State of the Physical Slot status. – 0x00 Inactive. – 0x01 Activate.

bLogicalSlot	• Logical Slot associated with this physical slot. THis is valid if the physical slot is active. – 1 - Slot 1. – 2 - Slot 2. – 3 - Slot 3. – 4 - Slot 4. – 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.

8.839.2 Field Documentation

8.839.2.1 uint8_t slot_t::bICCID[255]

8.839.2.2 uint8_t slot_t::bICCIDLength

8.839.2.3 uint8_t slot_t::bLogicalSlot

8.839.2.4 uint32_t slot_t::uPhyCardStatus

8.839.2.5 uint32_t slot_t::uPhySlotStatus

8.840 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]

8.840.1 Detailed Description

This structure contains information about the SLOTS present.

Parameters

cardState	• Indicates the state of the card for each slot. – 0 - Absent – 1 - Present – 2 - Error

upinState	• Indicates the state of UPIN. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked – 0xFF - Not Available
upinRetries	• Indicates the number of retries remaining to verify the UPIN. • If 0xFF, information not available.
upukRetries	• Indicates the number of retries remaining to unblock the UPIN. • If 0xFF, information not available.
errorState	• Indicates the reason for the card error, and is valid only when the card state is Error – 0 - Unknown – 1 - Power down – 2 - Poll error – 3 - No ATR received – 4 - Volt mismatch – 5 - Parity error – 6 - Unknown; possibly removed – 7 - Card returned technical problems – 0xFF - Not Available • Other values are possible and reserved for future use. • When an unknown value is received, it is to be handled as "Unknown".
numApp	• Indicates the number of applications available on the card. • The following block is repeated for each application. i.e. AppStatus. • If zero(0) then no AppStatus information exists.
AppStatus[MAX_NO_OF_APPLICATIONS]	• See appStats for more information.

8.840.2 Field Documentation

8.840.2.1 appStats slotInf::AppStatus[10]

8.840.2.2 uint8_t slotInf::cardState

8.840.2.3 uint8_t slotInf::errorState

8.840.2.4 uint8_t slotInf::numApp

8.840.2.5 uint8_t slotInf::upinRetries

8.840.2.6 uint8_t slotInf::upinState

8.840.2.7 uint8_t slotInf::upukRetries

8.841 slotInfo Struct Reference

Data Fields

- BYTE cardState
- BYTE upinState
- BYTE upinRetries
- BYTE upukRetries
- BYTE errorState
- BYTE numApp
- appStatus AppStatus [10]

8.841.1 Detailed Description

This structure contains information about the SLOTS present.

Parameters

cardState	• Indicates the state of the card for each slot. – 0 - Absent – 1 - Present – 2 - Error
upinState	• Indicates the state of UPIN. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked – 0xFF - Not Available
upinRetries	• Indicates the number of retries remaining to verify the UPIN. • If 0xFF, information not available.
upukRetries	• Indicates the number of retries remaining to unblock the UPIN. • If 0xFF, information not available.

errorState	• Indicates the reason for the card error, and is valid only when the card state is Error – 0 - Unknown – 1 - Power down – 2 - Poll error – 3 - No ATR received – 4 - Volt mismatch – 5 - Parity error – 6 - Unknown; possibly removed – 7 - Card returned technical problems – 0xFF - Not Available • Other values are possible and reserved for future use. • When an unknown value is received, it is to be handled as "Unknown".
numApp	• Indicates the number of applications available on the card. • The following block is repeated for each application. i.e. AppStatus. • If zero(0) then no AppStatus information exists.
AppStatus[MAX_NO_OF_APPLICATIONS]	• See appStatus for more information.

8.841.2 Field Documentation

8.841.2.1 appStatus slotInfo::AppStatus[10]

8.841.2.2 BYTE slotInfo::cardState

8.841.2.3 BYTE slotInfo::errorState

8.841.2.4 BYTE slotInfo::numApp

8.841.2.5 BYTE slotInfo::upinRetries

8.841.2.6 BYTE slotInfo::upinState

8.841.2.7 BYTE slotInfo::upukRetries

8.842 slots_t Struct Reference

Data Fields

- slot_t uimSlotStatus [255]

8.842.1 Field Documentation

8.842.1.1 slot_t slots_t::uimSlotStatus[255]

8.843 slqsautoconnect Struct Reference

Data Fields

- BOOL action
- BYTE acsetting
- BYTE acroamsetting

8.843.1 Detailed Description

structure contains autoconnect settings parameters

Parameters

action	• 0 - get autoconnect settings • 1 - set autoconnect settings
acsetting	• Current autoconnect setting: – 0x00 - Autoconnect disabled – 0x01 - Autoconnect enabled – 0x02 - Autoconnect paused (resume on powercycle)
acroamsetting	• Current autoconnect roaming status – 0x00 - Autoconnect always allowed – 0x01 - Autoconnect while in home service area only

8.843.2 Field Documentation

8.843.2.1 BYTE slqsautoconnect::acroamsetting

8.843.2.2 BYTE slqsautoconnect::acsetting

8.843.2.3 BOOL slqsautoconnect::action

8.844 SLQSDeleteProfileParams Struct Reference

Data Fields

- BYTE profileType
- BYTE profileIndex

8.844.1 Detailed Description

This structure contains the information about the profile to be deleted.

Parameters

profileType	• Identifies the type of profile – 0x00 – 3GPP • Note: Deletion of 3GPP2 profiles is not supported.

profileIndex	• Index of the configured profile to be deleted – Value between 1 - 16 (EM/MC73xx or earlier) – Value between 1 - 24 (EM/MC74xx onwards)

8.844.2 Field Documentation

8.844.2.1 BYTE SLQSDeleteProfileParams::profileIndex

8.844.2.2 BYTE SLQSDeleteProfileParams::profileType

8.845 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]

8.845.1 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 • deprecated on EM/MC74xx(9x30) devices
carrier_str	• carrier string • 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

8.845.2 Field Documentation

8.845.2.1 CHAR slqsfwinfo_s::appversion_str[85]

8.845.2.2 CHAR slqsfwinfo_s::bootversion_str[85]

8.845.2.3 CHAR slqsfwinfo_s::carrier_str[20]

8.845.2.4 CHAR slqsfwinfo_s::cur_carr_name[17]

8.845.2.5 CHAR slqsfwinfo_s::cur_carr_rev[13]

8.845.2.6 CHAR slqsfwinfo_s::modelid_str[20]

8.845.2.7 CHAR slqsfwinfo_s::packageid_str[85]

8.845.2.8 CHAR slqsfwinfo_s::priversion_str[16]

8.845.2.9 CHAR slqsfwinfo_s::sku_str[15]

8.846 SlqsNas3GppNetworkInfo Struct Reference

Data Fields

- WORD MCC
- WORD MNC
- ULONG InUse
- ULONG Roaming
- ULONG Forbidden
- ULONG Preferred
- CHAR Description [255]

8.846.1 Detailed Description

Contain the 3GPP network information.

Parameters

MCC	• Mobile Country Code
MNC	• Mobile Network Code

InUse	• Is the Network the current serving Network – 0 - Unknown – 1 - Current serving network – 2 - Not current serving network, available
Roaming	• Home/Roam Status of the Network – 0 - Unknown – 1 - Home – 2 - Roam
Forbidden	• Is the Network in the forbidden network list – 0 - Unknown – 1 - Forbidden – 2 - Not Forbidden
Preferred	• Is the Network in the Preferred network list – 0 - Unknown – 1 - Preferred – 2 - Not Preferred
Description	• Network Name/Description • This is a NULL terminated string.

8.846.2 Field Documentation

8.846.2.1 CHAR SlqsNas3GppNetworkInfo::Description[255]

8.846.2.2 ULONG SlqsNas3GppNetworkInfo::Forbidden

8.846.2.3 ULONG SlqsNas3GppNetworkInfo::InUse

8.846.2.4 WORD SlqsNas3GppNetworkInfo::MCC

8.846.2.5 WORD SlqsNas3GppNetworkInfo::MNC

8.846.2.6 ULONG SlqsNas3GppNetworkInfo::Preferred

8.846.2.7 ULONG SlqsNas3GppNetworkInfo::Roaming

8.847 SlqsNasPcsDigit Struct Reference

Data Fields

- WORD MCC
- WORD MNC
- BYTE includes_pcs_digit

8.847.1 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 • 0x01 - MNC is a three-digit value • 0x00 - MNC is a two-digit value

8.847.2 Field Documentation

8.847.2.1 BYTE SlqsNasPcsDigit::includes_pcs_digit

8.847.2.2 WORD SlqsNasPcsDigit::MCC

8.847.2.3 WORD SlqsNasPcsDigit::MNC

8.848 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

8.848.1 Detailed Description

This structure contains SMS parameters

Parameters

messageFormat	• Message format • Values: – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 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. • Values: – 0x00 - Do not care about the channel on which the message is sent – 0x01 - Request to send the message over the dedicated channel
pServiceOption	• Service option: • Values: – 0x00 - SO_AUTO - AUTO (choose the best service option) – 0x06 - SO_6 - Service option 6 – 0x0E - SO_14 - Service option 14
pFollowOnDC	• Flag to request not to disconnect the CDMA dedicated channel after the send operation is complete. • This TLV can be included if more messages are expected to follow. • Values: – 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. • Values: – 0x00 - Message is not to be sent on IMS – 0x01 - Message is to be sent on IMS – 0x02 to 0xFF - Reserved
pRetryMessage	• Indicates this message is a retry message. • Values: – 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. • 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. • The control point might send numerous requests. • This TLV will help the control point to identify the request for which the received indication belongs.

8.848.2 Field Documentation

8.848.2.1 ULONG slqssendasyncsmsparams_s::messageFormat

8.848.2.2 ULONG slqssendasyncsmsparams_s::messageSize

8.848.2.3 BYTE* slqssendasyncsmsparams_s::pFollowOnDC

8.848.2.4 BYTE* slqssendasyncsmsparams_s::pForceOnDC

8.848.2.5 BYTE* slqssendasyncsmsparams_s::pLinktimer

8.848.2.6 BYTE* slqssendasyncsmsparams_s::pMessage

8.848.2.7 BYTE* slqssendasyncsmsparams_s::pRetryMessage

8.848.2.8 ULONG* slqssendasyncsmsparams_s::pRetryMessageId

8.848.2.9 BYTE* slqssendasyncsmsparams_s::pServiceOption

8.848.2.10 BYTE* slqssendasyncsmsparams_s::pSmsOnIms

8.848.2.11 ULONG* slqssendasyncsmsparams_s::pUserData

8.849 slqssendsmsparams_s Struct Reference

Data Fields

- ULONG messageFormat
- ULONG messageSize
- BYTE * pMessage
- USHORT messageID
- ULONG messageFailureCode
- BYTE * pLinktimer
- BYTE * pSmsOnIms

8.849.1 Detailed Description

This structure contains SMS parameters

Parameters

messageFormat[IN]	• Message format – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 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 • If cause code is not provided, then value will be 0xFFFFFFFF
pSmsOnIms[IN]	• (Optional) SMS on IMS • Indicates whether the message is to be sent on IMS. • Values: – 0x00 - Message is not to be sent on IMS – 0x01 - Message is to be sent on IMS – 0x02 to 0xFF - Reserved
pLinktimer[IN]	• (Optional) Link Timer • Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow

8.849.2 Field Documentation

8.849.2.1 ULONG slqssendsmsparams_s::messageFailureCode

8.849.2.2 ULONG slqssendsmsparams_s::messageFormat

8.849.2.3 USHORT slqssendsmsparams_s::messageID

8.849.2.4 ULONG slqssendsmsparams_s::messageSize

8.849.2.5 BYTE* slqssendsmsparams_s::pLinktimer

8.849.2.6 BYTE* slqssendsmsparams_s::pMessage

8.849.2.7 BYTE* slqssendsmsparams_s::pSmsOnIms

8.850 slqsSessionStateInfo Struct Reference

Data Fields

- qaQmiInterfaceInfo * pQmiInterfaceInfo
- ULONG reconfiguration_required
- ULONG state
- ULONG sessionEndReason

8.850.1 Detailed Description

Contains the session state information and information about the interface

Parameters

pQmiInterface-Info	• See qaQmiInterfaceInfo for more information
state	• Current Link Status – 1 Disconnected – 2 Connected – 3 Suspended (Unsupported) – 4 Authenticating
reconfiguration_-required	• Indicates if host needs to be reconfigured – 0 No need to reconfigure – 1 Reconfiguration required
sessionEnd-Reason	• See qaGobiApiTableCallEndReasons.h for Call End Reason

8.850.2 Field Documentation

8.850.2.1 qaQmiInterfaceInfo* slqsSessionStateInfo::pQmiInterfaceInfo

8.850.2.2 ULONG slqsSessionStateInfo::reconfiguration_required

8.850.2.3 ULONG slqsSessionStateInfo::sessionEndReason

8.850.2.4 ULONG slqsSessionStateInfo::state

8.851 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

8.851.1 Detailed Description

This structure contains the Signal Strength Information

Parameters

signalStrengthReqMask[IN]	• Request Mask – Request additional signal information for: Bit 0 - RSSI Information bit Valid values are: 0 - Do Not Request Additional Info for RSSI 1 - Request Additional Info for RSSI Bit 1 - ECIO Information bit Valid values are: 0 - Do Not Request Additional Info for ECIO 1 - Request Additional Info for ECIO Bit 2 - IO Information bit Valid values are: 0 - Do Not Request Additional Info for IO 1 - Request Additional Info for IO Bit 3 - SINR Information bit Valid values are: 0 - Do Not Request Additional Info for SINR 1 - Request Additional Info for SINR Bit 4 - ERROR RATE Information bit Valid values are: 0 - Do Not Request Additional Info for Error Rate 1 - Request Additional Info for Error Rate Bit 5 - RSRQ Information bit Valid values are: 0 - Do Not Request Additional Info for RSRQ 1 - Request Additional Info for RSRQ Bit 6 - LTE SNR information bit Valid values are: 0 - Do not request additional information for LTE SNR 1 - Request additional information for LTE SNR Bit 7 - LTE RSRP Information bit Valid values are: 0 - Do not request additional information for LTE RSRP 1 - Request additional information for LTE RSRP
rxSignalStrengthListLen[OUT]	• Number of elements in Receive Signal Strength List
rxSignalStrengthList[OUT]	• See rxSignalStrengthListElement for more information
ecioListLen[OUT]	• 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 – 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

8.851.2 Field Documentation

8.851.2.1 struct ecioListElement slqsSignalStrengthInfo::ecioList[18]

8.851.2.2 USHORT slqsSignalStrengthInfo::ecioListLen

8.851.2.3 struct errorRateListElement slqsSignalStrengthInfo::errorRateList[18]

8.851.2.4 USHORT slqsSignalStrengthInfo::errorRateListLen

8.851.2.5 INT32 slqsSignalStrengthInfo::Io

8.851.2.6 SHORT slqsSignalStrengthInfo::ltersrp

8.851.2.7 SHORT slqsSignalStrengthInfo::ltesnr

8.851.2.8 struct rsrqInformation slqsSignalStrengthInfo::rsrqInfo

8.851.2.9 struct rxSignalStrengthListElement slqsSignalStrengthInfo::rxSignalStrengthList[18]

8.851.2.10 USHORT slqsSignalStrengthInfo::rxSignalStrengthListLen

8.851.2.11 USHORT slqsSignalStrengthInfo::signalStrengthReqMask

8.851.2.12 BYTE slqsSignalStrengthInfo::sinr

8.852 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

8.852.1 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. • 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: – Maximum number of threshold values is 10 – 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: – Maximum number of threshold values is 5 – 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

8.852.2 Field Documentation

8.852.2.1 BYTE SLQSSignalStrengthsIndReq::ecioDelta

8.852.2.2 SHORT SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.852.2.3 BYTE SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.852.2.4 BYTE SLQSSignalStrengthsIndReq::ioDelta

8.852.2.5 BYTE SLQSSignalStrengthsIndReq::lteRsrpDelta

8.852.2.6 WORD SLQSSignalStrengthsIndReq::lteSnrDelta

8.852.2.7 BYTE SLQSSignalStrengthsIndReq::rsrqDelta

8.852.2.8 BYTE SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.852.2.9 BYTE SLQSSignalStrengthsIndReq::sinrDelta

8.852.2.10 BYTE SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.852.2.11 BYTE SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.853 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

8.853.1 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 – 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

8.853.2 Field Documentation

8.853.2.1 struct ecioListElement SLQSSignalStrengthsInformation::ecioInfo

8.853.2.2 struct errorRateListElement SLQSSignalStrengthsInformation::errorRateInfo

8.853.2.3 ULONG SLQSSignalStrengthsInformation::io

8.853.2.4 struct lteRsrpinformation SLQSSignalStrengthsInformation::lteRsrpinfo

8.853.2.5 struct lteSnrinformation SLQSSignalStrengthsInformation::lteSnrinfo

8.853.2.6 struct rsrqInformation SLQSSignalStrengthsInformation::rsrqInfo

8.853.2.7 struct rxSignalStrengthListElement SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.853.2.8 BYTE SLQSSignalStrengthsInformation::sinr

8.854 slqsWdsEventInfo Struct Reference

Data Fields

- qaQmiInterfaceInfo * pQmiInterfaceInfo
- ULONG * pDormancyStatus
- ULONG * pDataBearer
- ULONG * pPacketsCountTX
- ULONG * pPacketsCountRX
- ULONGLONG * pTotalBytesTX
- ULONGLONG * pTotalBytesRX

8.854.1 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) • 0x00 - Indicates that this field is ignored • 0x01 - CDMA 1X • 0x02 - EV-DO Rev 0 • 0x03 - GPRS • 0x04 - WCDMA • 0x05 - EV-DO Rev A • 0x06 - EDGE • 0x07 - HSDPA and WCDMA • 0x08 - WCDMA and HSUPA • 0x09 - HSDPA and HSUPA • 0x0A - LTE • 0x0B - EV-DO Rev A EHRPD • 0x0C - HSDPA+ and WCDMA • 0x0D - HSDPA+ and HSUPA • 0x0E - DC_HSDPA+ and WCDMA • 0x0F - DC_HSDPA+ and HSUPA • 0x8000 - NULL Bearer • 0xFF - Unknown Technology
pDormancy-Status	• Dormancy status (NULL if not present) – 1 - traffic channel dormant – 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)

8.854.2 Field Documentation

8.854.2.1 ULONG* slqsWdsEventInfo::pDataBearer

8.854.2.2 ULONG* slqsWdsEventInfo::pDormancyStatus

8.854.2.3 ULONG* slqsWdsEventInfo::pPacketsCountRX

8.854.2.4 ULONG* slqsWdsEventInfo::pPacketsCountTX

8.854.2.5 qaQmiInterfaceInfo* slqsWdsEventInfo::pQmiInterfaceInfo

8.854.2.6 ULONGLONG* slqsWdsEventInfo::pTotalBytesRX

8.854.2.7 ULONGLONG* slqsWdsEventInfo::pTotalBytesTX

8.855 sms_BroadcastConfig Struct Reference

Data Fields

- uint16_t fromServiceId
- uint16_t toServiceId
- uint8_t selected

8.855.1 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 – 0x00 – Not selected – 0x01 – Selected

8.855.2 Field Documentation

8.855.2.1 uint16_t sms_BroadcastConfig::fromServiceId

8.855.2.2 uint8_t sms_BroadcastConfig::selected

8.855.2.3 uint16_t sms_BroadcastConfig::toServiceId

8.856 sms_CDMABroadcastConfig Struct Reference

Data Fields

- uint16_t serviceCategory
- uint16_t language
- uint8_t selected

8.856.1 Detailed Description

This structure contains CDMABroadcastConfig parameters

Parameters

serviceCategory	• Service category
language	• Language
selected	• Specified service_category and language – 0x00 – Not selected – 0x01 – Selected

8.856.2 Field Documentation

8.856.2.1 uint16_t sms_CDMABroadcastConfig::language

8.856.2.2 uint8_t sms_CDMABroadcastConfig::selected

8.856.2.3 uint16_t sms_CDMABroadcastConfig::serviceCategory

8.857 sms_getIndicationReg Struct Reference

Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

8.857.1 Detailed Description

This structure contains Get Indication Register Response parameters

Parameters

pRegTrans-LayerInfoEvt	- • Optional parameter indicating registration status of transport layer information events • Values: – 0x00 - Disabled – 0x01 - Enabled • returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.
pRegTransNW-RegInfoEvt	- • Optional parameter indicating registration status of transport network registration information events • Values: – 0x00 - Disabled – 0x01 - Enabled • 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
	- Values:
	  – 0x00 - Disabled
	  – 0x01 - Enabled
	- returns a default value 0xFF if this parameter is allocated memory in the structure and no response is received from the device.

8.857.2 Field Documentation

8.857.2.1 uint8_t* sms_getIndicationReg::pRegCallStatInfoEvt

8.857.2.2 uint8_t* sms_getIndicationReg::pRegTransLayerInfoEvt

8.857.2.3 uint8_t* sms_getIndicationReg::pRegTransNWRegInfoEvt

8.858 sms_getMsgWaitingInfo Struct Reference

Data Fields

- uint8_t numInstances
- sms_messageWaitingInfoContent msgWaitInfo [0xFF]

8.858.1 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. – See sms_messageWaitingInfoContent for more information.

8.858.2 Field Documentation

8.858.2.1 sms_messageWaitingInfoContent sms_getMsgWaitingInfo::msgWaitInfo[0xFF]

8.858.2.2 uint8_t sms_getMsgWaitingInfo::numInstances

8.859 sms_getTransLayerInfo Struct Reference

Data Fields

- uint8_t * pRegInd
- sms_transLayerInfo * pTransLayerInfo

8.859.1 Detailed Description

This structure contains Get Transport Layer Info Response parameters

Parameters

pRegInd	- • Optional parameter indicating if transport layer is registered • Values: – 0x00 - Transport layer is not registered – 0x01 - Transport layer is registered • returns a default value 0xFF if no response is received from the device.
pTransLayerInfo	• Pointer to structure of sms_transLayerInfo. – Optional parameter – See sms_transLayerInfo for more information • returns a default value 0xFF for parameter values if no response is received from the device.

8.859.2 Field Documentation

8.859.2.1 uint8_t* sms_getTransLayerInfo::pRegInd

8.859.2.2 sms_transLayerInfo* sms_getTransLayerInfo::pTransLayerInfo

8.860 sms_getTransNWRegInfo Struct Reference

Data Fields

- uint8_t * pRegStatus

8.860.1 Detailed Description

This structure contains transport network registration info parameter

Parameters

pRegStatus	- • Optional parameter indicating transport layer network registration status • Values: – 0x00 - No service – 0x01 - In progress – 0x02 - Failed – 0x03 - Limited Service – 0x04 - Full Service • returns a default value 0xFF if no response is received from the device.

8.860.2 Field Documentation

8.860.2.1 uint8_t* sms_getTransNWRegInfo::pRegStatus

8.861 sms_maxStorageSizeReq Struct Reference

Data Fields

- uint8_t storageType
- uint8_t * pMessageMode

8.861.1 Detailed Description

This structure contains get store max size request parameters

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
pMessage-Mode(optional	parameter) • 0x00 - CDMA, LTE (if network type is CDMA) • 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

8.861.2 Field Documentation

8.861.2.1 uint8_t* sms_maxStorageSizeReq::pMessageMode

8.861.2.2 uint8_t sms_maxStorageSizeReq::storageType

8.862 sms_maxStorageSizeResp Struct Reference

Data Fields

- uint32_t maxStorageSize
- uint32_t freeSlots

8.862.1 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 - returns a default value 0xFFFFFFFF for parameter values if no response is received from the device.

8.862.2 Field Documentation

8.862.2.1 uint32_t sms_maxStorageSizeResp::freeSlots

8.862.2.2 uint32_t sms_maxStorageSizeResp::maxStorageSize

8.863 sms_messageWaitingInfoContent Struct Reference

Data Fields

- uint8_t msgType
- uint8_t activeInd
- uint8_t msgCount

8.863.1 Detailed Description

This structure contains message waiting information per instance

Parameters

msgType	- Message type
	  – 0x00 - MWI_MESSAGE_TYPE_VOICMAIL - Voicemail
	  – 0x01 - MWI_MESSAGE_TYPE_FAX - Fax
	  – 0X02 - MWI_MESSAGE_TYPE_EMAIL - Email
	  – 0x03 - MWI_MESSAGE_TYPE_OTHER - Other
	  – 0x04 - MWI_MESSAGE_TYPE_VIDEOMAIL - Videomail

activeInd	- Indicates whether the indication is active
	  – 0x00 - Inactive
	  – 0x01 - Active

msgCount	- Number of messages

8.863.2 Field Documentation

8.863.2.1 uint8_t sms_messageWaitingInfoContent::activeInd

8.863.2.2 uint8_t sms_messageWaitingInfoContent::msgCount

8.863.2.3 uint8_t sms_messageWaitingInfoContent::msgType

8.864 sms_msgProtocolResp Struct Reference

Data Fields

- uint8_t msgProtocol

8.864.1 Detailed Description

This structure contains get message protocol response parameters

Parameters

msgProtocol	- • Message Protocol • Values: – 0x00 - MESSAGE_PROTOCOL_CDMA – 0x01 - MESSAGE_PROTOCOL_WCDMA

8.864.2 Field Documentation

8.864.2.1 uint8_t sms_msgProtocolResp::msgProtocol

8.865 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference

Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_CDMABroadcastConfig CDMABroadcastConfig [0x05]

8.865.1 Detailed Description

This structure contains the 3GPP2 Broadcast Configuration Information parameters

Parameters

activated_ind	• Broadcast SMS – 0x00 - Deactivated – 0x01 - Activated
num_instances	• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. – serviceCategory – language – selected
broadcastConfig	• A CDMABroadcastConfig structure array. • Further defined by the structure sms_CDMABroadcastConfig

8.865.2 Field Documentation

8.865.2.1 uint8_t sms_qaQmi3GPP2BroadcastCfgInfo::activated_ind

8.865.2.2 sms_CDMABroadcastConfig sms_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]

8.865.2.3 uint16_t sms_qaQmi3GPP2BroadcastCfgInfo::num_instances

8.866 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference

Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_BroadcastConfig broadcastConfig [0x05]

8.866.1 Detailed Description

This structure contains the 3GPP Broadcast Configuration Information parameters

Parameters

activated_ind	• Broadcast SMS – 0x00 - Deactivated – 0x01 - Activated
num_instances	• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. – fromServiceId – toServiceId – selected
broadcastConfig	• A BroadcastConfig structure array. • Further defined by the structure sms_BroadcastConfig

8.866.2 Field Documentation

8.866.2.1 uint8_t sms_qaQmi3GPPBroadcastCfgInfo::activated_ind

8.866.2.2 sms_BroadcastConfig sms_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

8.866.2.3 uint16_t sms_qaQmi3GPPBroadcastCfgInfo::num_instances

8.867 sms_routeEntry Struct Reference

Data Fields

- uint8_t messageType
- uint8_t messageClass

- uint8_t routeStorage
- uint8_t receiptAction

8.867.1 Detailed Description

This structure contains SMS route entry details

Parameters

messageType	- • Message type matching this route • Values: – 0x00 - MESSAGE_TYPE_POINT_TO_POINT
messageClass	- • Message Class • Values: – 0x00 - MESSAGE_CLASS_0 – 0x01 - MESSAGE_CLASS_1 – 0x02 - MESSAGE_CLASS_2 – 0x03 - MESSAGE_CLASS_3 – 0x04 - MESSAGE_CLASS_NONE – 0x05 - MESSAGE_CLASS_CDMA
routeStorage	- • If the receiptAction is store where to store the message • Values: – 0x00 - STORAGE_TYPE_UIM – 0x01 - STORAGE_TYPE_NV – 0xFF - STORAGE_TYPE_NONE
receiptAction	- • Action to be taken on receipt of a message matching the specified type and class for this route • Values: – 0x00 - DISCARD (discarded without notification) – 0x01 - STORE AND NOTIFY (stored and notified to the registered clients) – 0x02 - TRANSFER ONLY (transferred to the client, client expected to send the ACK) – 0x03 - TRANSFER AND ACK (transferred to the client, device expected to send the ACK)

8.867.2 Field Documentation

8.867.2.1 uint8_t sms_routeEntry::messageClass

8.867.2.2 uint8_t sms_routeEntry::messageType

8.867.2.3 uint8_t sms_routeEntry::receiptAction

8.867.2.4 uint8_t sms_routeEntry::routeStorage

8.868 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

8.868.1 Detailed Description

This structure contains SMS parameters

Parameters

messageFormat	• Message format • Values: – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 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. • Values: – 0x00 - Do not care about the channel on which the message is sent – 0x01 - Request to send the message over the dedicated channel
pServiceOption	• Service option: • Values: – 0x00 - SO_AUTO - AUTO (choose the best service option) – 0x06 - SO_6 - Service option 6 – 0x0E - SO_14 - Service option 14
pFollowOnDC	• Flag to request not to disconnect the CDMA dedicated channel after the send operation is complete. • This TLV can be included if more messages are expected to follow. • Values: – 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. • Values: – 0x00 - Message is not to be sent on IMS – 0x01 - Message is to be sent on IMS – 0x02 to 0xFF - Reserved
pRetryMessage	• Indicates this message is a retry message. • Values: – 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. • 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. • The control point might send numerous requests. • This TLV will help the control point to identify the request for which the received indication belongs.

8.868.2 Field Documentation

8.868.2.1 uint32_t sms_sendAsyncsmsParams::messageFormat

8.868.2.2 uint32_t sms_sendAsyncsmsParams::messageSize

8.868.2.3 uint8_t* sms_sendAsyncsmsParams::pFollowOnDC

8.868.2.4 uint8_t* sms_sendAsyncsmsParams::pForceOnDC

8.868.2.5 uint8_t* sms_sendAsyncsmsParams::pLinktimer

8.868.2.6 uint8_t* sms_sendAsyncsmsParams::pMessage

8.868.2.7 uint8_t* sms_sendAsyncsmsParams::pRetryMessage

8.868.2.8 uint32_t* sms_sendAsyncsmsParams::pRetryMessageId

8.868.2.9 uint8_t* sms_sendAsyncsmsParams::pServiceOption

8.868.2.10 uint8_t* sms_sendAsyncsmsParams::pSmsOnIms

8.868.2.11 uint32_t* sms_sendAsyncsmsParams::pUserData

8.869 sms_setIndicationReg Struct Reference

Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

8.869.1 Detailed Description

This structure contains Indication Register request parameters

Parameters

pRegTransLayerInfoEvt	- Optional parameter indicating registration status of transport layer information events - Values: – 0x00 - Disabled – 0x01 - Enabled – 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 - Values: – 0x00 - Disabled – 0x01 - Enabled – 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 - Values: – 0x00 - Disabled – 0x01 - Enabled – 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

8.869.2 Field Documentation

8.869.2.1 uint8_t* sms_setIndicationReg::pRegCallStatInfoEvt

8.869.2.2 uint8_t* sms_setIndicationReg::pRegTransLayerInfoEvt

8.869.2.3 uint8_t* sms_setIndicationReg::pRegTransNWRegInfoEvt

8.870 sms_setRoutesReq Struct Reference

Data Fields

- uint16_t numOfRoutes

- sms_routeEntry routeList [0x0A]
- uint8_t * pTransferStatusReport

8.870.1 Detailed Description

This structure contains SMS route request parameters

Parameters

numOfRoutes	- • Number of sets of the following element
routeList	- • Array containing the set of sms_routeEntry
pTransferStatus-Report	- • 0x01 - Status report are transferred to the client (optional)

8.870.2 Field Documentation

8.870.2.1 uint16_t sms_setRoutesReq::numOfRoutes

8.870.2.2 uint8_t* sms_setRoutesReq::pTransferStatusReport

8.870.2.3 sms_routeEntry sms_setRoutesReq::routeList[0x0A]

8.871 sms_transLayerInfo Struct Reference

Data Fields

- uint8_t TransType
- uint8_t TransCap

8.871.1 Detailed Description

This structure contains Transport Layer Information

Parameters

TransType	• Transport Type – 0x00 - IMS
TransCap	• Transport Capability • Values: – 0x00 - CDMA – 0x01 - GW

8.871.2 Field Documentation

8.871.2.1 uint8_t sms_transLayerInfo::TransCap

8.871.2.2 uint8_t sms_transLayerInfo::TransType

8.872 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

8.872.1 Detailed Description

This structure contains SMS parameters

Parameters

sendStatus	• Send Status • Values: – QMI_ERR_NONE – No error in the request – QMI_ERR_CAUSE_CODE - SMS cause code – QMI_ERR_MESSAGE_DELIVERY_FAILURE - Message could not be delivered – 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 • Values: – 0x00 - ERROR_CLASS_TEMPORARY – 0x01 - ERROR_CLASS_PERMANENT
RPCause	• GW RP cause
TPCause	• GW TP Cause

msgDelFailure-Type	• Message delivery failure type • Values: – 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY – 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT
msgDelFailure-Cause	• Message delivery failure cause • Values: – 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.

8.872.2 Field Documentation

8.872.2.1 BYTE SMSAsyncRawSend_s::alphaIDLen

8.872.2.2 WORD SMSAsyncRawSend_s::causeCode

8.872.2.3 BYTE SMSAsyncRawSend_s::errorClass

8.872.2.4 WORD SMSAsyncRawSend_s::messageID

8.872.2.5 BYTE SMSAsyncRawSend_s::msgDelFailureCause

8.872.2.6 BYTE SMSAsyncRawSend_s::msgDelFailureType

8.872.2.7 BYTE* SMSAsyncRawSend_s::pAlphaID

8.872.2.8 WORD SMSAsyncRawSend_s::RPCause

8.872.2.9 WORD SMSAsyncRawSend_s::sendStatus

8.872.2.10 BYTE SMSAsyncRawSend_s::TPCause

8.872.2.11 ULONG SMSAsyncRawSend_s::userData

8.873 sMSCAddress Struct Reference

Data Fields

- uint8_t length
- uint8_t data [256]

8.873.1 Detailed Description

Parameters

length	• Number of sets of following element
data	• SMSC address

8.873.2 Field Documentation

8.873.2.1 uint8_t sMSCAddress::data[256]

8.873.2.2 uint8_t sMSCAddress::length

8.874 SMSCAddress Struct Reference

Data Fields

- BYTE length
- BYTE data [256]

8.874.1 Detailed Description

This structure holds SMSC information

Parameters

length	• Number of sets of following element
data	• SMSC address

8.874.2 Field Documentation

8.874.2.1 BYTE SMSCAddress::data[256]

8.874.2.2 BYTE SMSCAddress::length

8.875 sMSCAddressTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSCAddressInfo SMSCInfo

8.875.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
SMSCInfo	• SMSC Address • See sMSCAddressInfo for more information

8.875.2 Field Documentation

8.875.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

8.875.2.2 uint8_t sMSCAddressTlv::TlvPresent

8.876 sMSEtwsMessage Struct Reference

Data Fields

- uint8_t notificationType
- uint16_t length
- uint8_t data [1254]

8.876.1 Detailed Description

Parameters

notificationType	• Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS
length	• Number of sets of following elements
data	• Raw message data

8.876.2 Field Documentation

8.876.2.1 uint8_t sMSEtwsMessage::data[1254]

8.876.2.2 uint16_t sMSEtwsMessage::length

8.876.2.3 uint8_t sMSEtwsMessage::notificationType

8.877 SMSEtwsMessage Struct Reference

Data Fields

- BYTE notificationType
- WORD length
- BYTE data [1254]

8.877.1 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

8.877.2 Field Documentation

8.877.2.1 BYTE SMSEtwsMessage::data[1254]

8.877.2.2 WORD SMSEtwsMessage::length

8.877.2.3 BYTE SMSEtwsMessage::notificationType

8.878 sMSEtwsMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSEtwsMessageInfo EtwsMessageInfo

8.878.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
EtwsMessage-Info	• ETWS Message • See sMSEtwsMessageInfo for more information

8.878.2 Field Documentation

8.878.2.1 sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo

8.878.2.2 uint8_t sMSEtwsMessageTlv::TlvPresent

8.879 SMSEtwsPlmn Struct Reference

Data Fields

- WORD mobileCountryCode
- WORD mobileNetworkCode

8.879.1 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

8.879.2 Field Documentation

8.879.2.1 WORD SMSEtwsPlmn::mobileCountryCode

8.879.2.2 WORD SMSEtwsPlmn::mobileNetworkCode

8.880 sMSEtwsPlmn Struct Reference

Data Fields

- uint16_t mobileCountryCode
- uint16_t mobileNetworkCode

8.880.1 Detailed Description

Parameters

mobileCountry-Code	• 16 bit representation of MCC value range : 0 -999
mobileNetwork-Code	• 16 bit representation of MNC value range : 0 -999

8.880.2 Field Documentation

8.880.2.1 uint16_t sMSEtwsPlmn::mobileCountryCode

8.880.2.2 uint16_t sMSEtwsPlmn::mobileNetworkCode

8.881 SMSEventInfo_s Struct Reference

Data Fields

- BYTE smsEventType

- SMSMTMessageInfo * pMTMessageInfo
- SMSTransferRouteMTMessageInfo * pTransferRouteMTMessageInfo
- SMSMessageModeInfo * pMessageModeInfo
- SMSEtwsMessageInfo * pEtwsMessageInfo
- SMSEtwsPlmnInfo * pEtwsPlmnInfo
- SMSCAddressInfo * pSMSCAddressInfo
- SMSOnIMSInfo * pSMSOnIMSInfo

8.881.1 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 NULL, if this event is not present in the smsEventType parameter
pTransferRoute-MTMessageInfo	• pointer to the SMSTransferRouteMTMessageInfo structure . NULL, if this event is not present in the smsEventType parameter
pMessageMode-Info	• pointer to the SMSMessageModeInfo structure NULL, if this event is not present in the smsEventType parameter
pEtwsMessage-Info	• pointer to the SMSEtwsMessageInfo structure NULL, if this event is not present in the smsEventType parameter
pEtwsPlmnInfo	• pointer to the SMSEtwsPlmnInfo structure NULL, if this event is not present in the smsEventType parameter
pSMSCAddress-Info	• pointer to the SMSCAddressInfo structure NULL, if this event is not present in the smsEventType parameter
pSMSOnIMSInfo	• pointer to the SMSOnIMSInfo structure NULL, if this event is not present in the smsEventType parameter Note: None

8.881.2 Field Documentation

8.881.2.1 SMSEtwsMessageInfo* SMSEventInfo_s::pEtwsMessageInfo

8.881.2.2 SMSEtwsPlmnInfo* SMSEventInfo_s::pEtwsPlmnInfo

8.881.2.3 SMSMessageModeInfo* SMSEventInfo_s::pMessageModeInfo

8.881.2.4 SMSMTMessageInfo* SMSEventInfo_s::pMTMessageInfo

8.881.2.5 SMSCAddressInfo* SMSEventInfo_s::pSMSCAddressInfo

8.881.2.6 SMSOnIMSInfo* SMSEventInfo_s::pSMSOnIMSInfo

8.881.2.7 SMSTransferRouteMTMessageInfo* SMSEventInfo_s::pTransferRouteMTMessageInfo

8.881.2.8 BYTE SMSEventInfo_s::smsEventType

8.882 smsMaxStorageSizeReq Struct Reference

Data Fields

- BYTE storageType
- BYTE * pMessageMode

8.882.1 Detailed Description

This structure contains get store max size resquest parameters

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
pMessage-Mode(optional	parameter) • 0x00 - CDMA, LTE (if network type is CDMA) • 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

8.882.2 Field Documentation

8.882.2.1 BYTE* smsMaxStorageSizeReq::pMessageMode

8.882.2.2 BYTE smsMaxStorageSizeReq::storageType

8.883 smsMaxStorageSizeResp Struct Reference

Data Fields

- ULONG maxStorageSize
- ULONG freeSlots

8.883.1 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.

8.883.2 Field Documentation

8.883.2.1 ULONG smsMaxStorageSizeResp::freeSlots

8.883.2.2 ULONG smsMaxStorageSizeResp::maxStorageSize

8.884 SMSMemoryInfo Struct Reference

Data Fields

- BYTE storageType
- BYTE messageMode

8.884.1 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 - MESSAGE_MODE_GW - GW

8.884.2 Field Documentation

8.884.2.1 BYTE SMSMemoryInfo::messageMode

8.884.2.2 BYTE SMSMemoryInfo::storageType

8.885 SMSMessageMode Struct Reference

Data Fields

- BYTE messageMode

8.885.1 Detailed Description

This structure holds information related to message mode

Parameters

messageMode	• Message mode 0x00 - CDMA 0x01 - GW

8.885.2 Field Documentation

8.885.2.1 BYTE SMSMessageMode::messageMode

8.886 sMSMessageMode Struct Reference

Data Fields

- uint8_t messageMode

8.886.1 Detailed Description

Parameters

messageMode	Message Mode

8.886.2 Field Documentation

8.886.2.1 uint8_t sMSMessageMode::messageMode

8.887 smsMsgprotocolResp Struct Reference

Data Fields

- BYTE msgProtocol

8.887.1 Detailed Description

This structure contains get message protocol response parameters

Parameters

msgProtocol	- • Message Protocol • Values: – 0x00 - MESSAGE_PROTOCOL_CDMA – 0x01 - MESSAGE_PROTOCOL_WCDMA

8.887.2 Field Documentation

8.887.2.1 BYTE smsMsgprotocolResp::msgProtocol

8.888 SMSMTMessage Struct Reference

Data Fields

- ULONG storageType
- ULONG messageIndex

8.888.1 Detailed Description

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

8.888.2 Field Documentation

8.888.2.1 ULONG SMSMTMessage::messageIndex

8.888.2.2 ULONG SMSMTMessage::storageType

8.889 sMSMTMessage Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex

8.889.1 Detailed Description

Parameters

storageType	memory storage 0x00-UIM 0x01-NV
messageIndex	MT Message index

8.889.2 Field Documentation

8.889.2.1 uint32_t sMSMTMessage::messageIndex

8.889.2.2 uint32_t sMSMTMessage::storageType

8.890 SMSOnIMS Struct Reference

Data Fields

- BYTE smsOnIMS

8.890.1 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.

8.890.2 Field Documentation

8.890.2.1 BYTE SMSOnIMS::smsOnIMS

8.891 sMSOnIMS Struct Reference

Data Fields

- uint8_t smsOnIMS

8.891.1 Detailed Description

Parameters

smsOnIMS	SMS on IMS

8.891.2 Field Documentation

8.891.2.1 uint8_t sMSOnIMS::smsOnIMS

8.892 sMSOnIMSTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSOnIMSInfo IMSInfo

8.892.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
IMSInfo	• SMS on IMS • See sMSOnIMSInfo for more information

8.892.2 Field Documentation

8.892.2.1 sMSOnIMSInfo sMSOnIMSTlv::IMSInfo

8.892.2.2 uint8_t sMSOnIMSTlv::TlvPresent

8.893 smsRouteEntry Struct Reference

Data Fields

- BYTE messageType
- BYTE messageClass
- BYTE routeStorage
- BYTE receiptAction

8.893.1 Detailed Description

This structure contains SMS route entry details

Parameters

messageType	- • Message type matching this route • Values: – 0x00 - MESSAGE_TYPE_POINT_TO_POINT
messageClass	- • Message Class • Values: – 0x00 - MESSAGE_CLASS_0 – 0x01 - MESSAGE_CLASS_1 – 0x02 - MESSAGE_CLASS_2 – 0x03 - MESSAGE_CLASS_3 – 0x04 - MESSAGE_CLASS_NONE – 0x05 - MESSAGE_CLASS_CDMA
routeStorage	- • If the receiptAction is store where to store the message • Values: – 0x00 - STORAGE_TYPE_UIM – 0x01 - STORAGE_TYPE_NV – 0xFF - STORAGE_TYPE_NONE

receiptAction	- • Action to be taken on receipt of a message matching the specified type and class for this route • Values: – 0x00 - DISCARD (discarded without notification) – 0x01 - STORE AND NOTIFY (stored and notified to the registered clients) – 0x02 - TRANSFER ONLY (transferred to the client, client expected to send the ACK) – 0x03 - TRANSFER AND ACK (transferred to the client, device expected to send the ACK)

8.893.2 Field Documentation

8.893.2.1 BYTE smsRouteEntry::messageClass

8.893.2.2 BYTE smsRouteEntry::messageType

8.893.2.3 BYTE smsRouteEntry::receiptAction

8.893.2.4 BYTE smsRouteEntry::routeStorage

8.894 smsSetRoutesReq Struct Reference

Data Fields

- WORD numOfRoutes
- smsRouteEntry routeList [0x0A]
- BYTE * pTransferStatusReport

8.894.1 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)

8.894.2 Field Documentation

8.894.2.1 WORD smsSetRoutesReq::numOfRoutes

8.894.2.2 BYTE* smsSetRoutesReq::pTransferStatusReport

8.894.2.3 smsRouteEntry smsSetRoutesReq::routeList[0x0A]

8.895 SMSTransferRouteMTMessage Struct Reference

Data Fields

- BYTE ackIndicator
- ULONG transactionID
- BYTE format
- WORD length
- BYTE data [256]

8.895.1 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

8.895.2 Field Documentation

8.895.2.1 BYTE SMSTransferRouteMTMessage::ackIndicator

8.895.2.2 BYTE SMSTransferRouteMTMessage::data[256]

8.895.2.3 BYTE SMSTransferRouteMTMessage::format

8.895.2.4 WORD SMSTransferRouteMTMessage::length

8.895.2.5 ULONG SMSTransferRouteMTMessage::transactionID

8.896 sMSTransferRouteMTMessage Struct Reference

Data Fields

- uint8_t ackIndicator
- uint32_t transactionID
- uint8_t format
- uint16_t length

- uint8_t data [256]

8.896.1 Detailed Description

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

8.896.2 Field Documentation

8.896.2.1 uint8_t sMSTransferRouteMTMessage::ackIndicator

8.896.2.2 uint8_t sMSTransferRouteMTMessage::data[256]

8.896.2.3 uint8_t sMSTransferRouteMTMessage::format

8.896.2.4 uint16_t sMSTransferRouteMTMessage::length

8.896.2.5 uint32_t sMSTransferRouteMTMessage::transactionID

8.897 sQosFlowStat Struct Reference

Data Fields

- ULONG bearerId
- ULONG tx_pkt
- ULONG tx_pkt_drp
- ULONGLONG tx_bytes
- ULONGLONG tx_bytes_drp

8.897.1 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

8.897.2 Field Documentation

8.897.2.1 ULONG sQosFlowStat::bearerId

8.897.2.2 ULONGLONG sQosFlowStat::tx_bytes

8.897.2.3 ULONGLONG sQosFlowStat::tx_bytes_drp

8.897.2.4 ULONG sQosFlowStat::tx_pkt

8.897.2.5 ULONG sQosFlowStat::tx_pkt_drp

8.898 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)]

8.898.1 Detailed Description

This structure contains the Data statistic per QoS flow

Parameters

apnId	• APN id • 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 • See sQosFlowStat for more information • See MAX_QOS_SPEC_PER_APN for more information

8.898.2 Field Documentation

8.898.2.1 ULONG sQosStat::apnId

8.898.2.2 ULONG sQosStat::numQosFlow

8.898.2.3 sQosFlowStat sQosStat::qosFlow[(10)]

8.898.2.4 ULONGLONG sQosStat::total_rx_bytes

8.898.2.5 ULONG sQosStat::total_rx_pkt

8.898.2.6 ULONGLONG sQosStat::total_tx_bytes

8.898.2.7 ULONGLONG sQosStat::total_tx_bytes_drp

8.898.2.8 ULONG sQosStat::total_tx_pkt

8.898.2.9 ULONG sQosStat::total_tx_pkt_drp

8.899 SrvStatusInfo Struct Reference

Data Fields

- BYTE srvStatus
- BYTE isPrefDataPath

8.899.1 Detailed Description

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

Parameters

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

8.899.2 Field Documentation

8.899.2.1 BYTE SrvStatusInfo::isPrefDataPath

8.899.2.2 BYTE SrvStatusInfo::srvStatus

8.900 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

8.900.1 Detailed Description

This structure contains the start/stop data session params Information

Parameters

action	• 1 - Start Session • 0 - Stop Session
instanceId	• PDP Instance. • Instance ID corresponding to the session ID
pTechnology	• Indicates the technology preference (optional) – 1 - UMTS – 2 - CDMA – 3 - eMBMS – 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 • [OUT] - Assigned session ID when starting a data session
failureReason	• Reason data session failed to be established • See qaGobiApiTableCallEndReasons.h for Call End Reason
failureReasonv4	• Reason v4 data session failed to be established • See qaGobiApiTableCallEndReasons.h for Call End Reason
failureReasonv6	• Reason v6 data session failed to be established • See qaGobiApiTableCallEndReasons.h for Call End Reason
rc4	• v4 result code • See qmerrno.h
rc6	• v6 result code • See qmerrno.h

v4sessionId	• Do not modify - used for internal management of data sessions • Non zero value indicates that a session is active
v6sessionId	• Do not modify - used for internal management of data sessions • Non zero value indicates that a session is active
ipfamily	• 4 for an IPv4 data session • 6 for an IPv6 data session • 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 • See qaGobiApiTableCallEndReasons.h for Call End Reason Type
verbFailReason	• Verbose reason explaining why call failed. Depends on verbFailReasonType parameter • See qaGobiApiTableCallEndReasons.h for Call End Reason

8.900.2 Field Documentation

8.900.2.1 BOOL ssdatasession_params::action

8.900.2.2 ULONG ssdatasession_params::failureReason

8.900.2.3 ULONG ssdatasession_params::failureReasonv4

8.900.2.4 ULONG ssdatasession_params::failureReasonv6

8.900.2.5 BYTE ssdatasession_params::instanceId

8.900.2.6 BYTE ssdatasession_params::ipfamily

8.900.2.7 ULONG* ssdatasession_params::pAuthentication

8.900.2.8 CHAR* ssdatasession_params::pPassword

8.900.2.9 ULONG* ssdatasession_params::pProfileId3GPP

8.900.2.10 ULONG* ssdatasession_params::pProfileId3GPP2

8.900.2.11 ULONG* ssdatasession_params::pTechnology

8.900.2.12 CHAR* ssdatasession_params::pUsername

8.900.2.13 ULONG ssdatasession_params::rcv4

8.900.2.14 ULONG ssdatasession_params::rcv6

8.900.2.15 ULONG ssdatasession_params::sessionId

8.900.2.16 ULONG ssdatasession_params::v4sessionId

8.900.2.17 ULONG ssdatasession_params::v6sessionId

8.900.2.18 ULONG ssdatasession_params::verbFailReason

8.900.2.19 ULONG ssdatasession_params::verbFailReasonType

8.901 SupportedMsgList Struct Reference

Data Fields

- WORD supportedMsgLen
- BYTE supportedMsgs [256]

8.901.1 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. • Starting with the LSB, bit 0 represents message ID 0, bit 1 represents message ID 1.

8.901.2 Field Documentation

8.901.2.1 WORD SupportedMsgList::supportedMsgLen

8.901.2.2 BYTE SupportedMsgList::supportedMsgs[256]

8.902 SUPSInfo Struct Reference

Data Fields

- BYTE svcType
- BYTE isModByCC

8.902.1 Detailed Description

This structure contains information about the Supplementary Services.

Parameters

svcType	• Service type. – 0x01 - SERVICE_TYPE_ACTIVATE - Activate – 0x02 - SERVICE_TYPE_DEACTIVATE - Deactivate – 0x03 - SERVICE_TYPE_REGISTER - Register – 0x04 - SERVICE_TYPE_ERASE - Erase – 0x05 - SERVICE_TYPE_INTERROGATE - Interrogate – 0x06 - SERVICE_TYPE_REGISTER_PASSWORD - Register password – 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: – 0 - False – 1 - True

8.902.2 Field Documentation

8.902.2.1 BYTE SUPSInfo::isModByCC

8.902.2.2 BYTE SUPSInfo::svcType

8.903 SV Struct Reference

Data Fields

- WORD id
- ULONG system
- BYTE mask

8.903.1 Detailed Description

This structure contains the Delete SV Info

Parameters

id	• SV ID of the satellite whose data is to be deleted • Range: – For GPS: 1 to 32 – For SBAS: 33 to 64 – For GLONASS: 65 to 96

system	• Indicates to which constellation this SV belongs • Valid values: – eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite – eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite – eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite – eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite – eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite – eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite
mask	• Indicates if the ephemeris or almanac for a satellite is to be deleted • Valid values: – 0x01 - DELETE_EPHEMERIS – 0x02 - DELETE_ALMANAC

8.903.2 Field Documentation

8.903.2.1 WORD SV::id

8.903.2.2 BYTE SV::mask

8.903.2.3 ULONG SV::system

8.904 SVInfo Struct Reference

Data Fields

- BYTE len
- SV * pSV

8.904.1 Detailed Description

This structure contains the elements of Delete SV Info

Parameters

len	• Number of sets of the following elements in struct SV: – gnssSvId – system – deleteSvInfoMask
pSV	• Pointer to struct SV. See SV for more information

8.904.2 Field Documentation

8.904.2.1 BYTE SVInfo::len

8.904.2.2 SV* SVInfo::pSV

8.905 svUsedforFix_s Struct Reference

Data Fields

- BYTE gnssSvUsedList_len
- WORD gnssSvUsedList [255]

8.905.1 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. • Following information is associated with each SV ID: – GPS - 1 to 32 – SBAS - 33 to 64 – GLONASS - 65 to 96 – QZSS - 193 to 197 – BDS - 201 to 237

8.905.2 Field Documentation

8.905.2.1 WORD svUsedforFix_s::gnssSvUsedList[255]

8.905.2.2 BYTE svUsedforFix_s::gnssSvUsedList_len

8.906 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]

8.906.1 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 • Values – 0 - Device – 1 - Host
m_FwImageId	• Firmware image ID
m_FwBuildId	• Firmware build ID
m_PriImageId	• PRI image ID
m_PriBuildId	• PRI build ID

8.906.2 Field Documentation

8.906.2.1 BYTE SWI_STRUCT_CarrierImage::m_FwBuildId[100]

8.906.2.2 BYTE SWI_STRUCT_CarrierImage::m_FwImageId[16]

8.906.2.3 ULONG SWI_STRUCT_CarrierImage::m_nCarrierId

8.906.2.4 ULONG SWI_STRUCT_CarrierImage::m_nFolderId

8.906.2.5 ULONG SWI_STRUCT_CarrierImage::m_nStorage

8.906.2.6 BYTE SWI_STRUCT_CarrierImage::m_PriBuildId[100]

8.906.2.7 BYTE SWI_STRUCT_CarrierImage::m_PriImageId[16]

8.907 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

8.907.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

Parameters

function	• Setting to indicate when modem should start an automatic GNSS fix – 0 - disabled – 1 - At bootup – 2 - When NMEA port is opened
function_reported	• 0 - not reported by modem • 1 - reported by modem
fix_type	• Type of GNSS fix: – 1 - Default Engine mode – 2 - MS-Based – 3 - MS-Assisted – 4 - Standalone
fix_type_reported	• 0 - not reported by modem • 1 - reported by modem
max_time	• Maximum time allowed for the receiver to get a fix in seconds • Valid range: 1-255
max_time_reported	• 0 - not reported by modem • 1 - reported by modem
max_dist	• Maximum uncertainty of a fix measured by distance in meters • Valid range: 1 - 4294967280
max_dist_reported	• 0 - not reported by modem • 1 - reported by modem
fix_rate	• Time between fixes in seconds • Valid range: 1–65535

fix_rate_-reported	• 0 - not reported by modem • 1 - reported by modem

8.907.2 Field Documentation

8.907.2.1 ULONG SwiLocGetAutoStartResp::fix_rate

8.907.2.2 BOOL SwiLocGetAutoStartResp::fix_rate_reported

8.907.2.3 BYTE SwiLocGetAutoStartResp::fix_type

8.907.2.4 BOOL SwiLocGetAutoStartResp::fix_type_reported

8.907.2.5 BYTE SwiLocGetAutoStartResp::function

8.907.2.6 BOOL SwiLocGetAutoStartResp::function_reported

8.907.2.7 ULONG SwiLocGetAutoStartResp::max_dist

8.907.2.8 BOOL SwiLocGetAutoStartResp::max_dist_reported

8.907.2.9 BYTE SwiLocGetAutoStartResp::max_time

8.907.2.10 BOOL SwiLocGetAutoStartResp::max_time_reported

8.908 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

8.908.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

Parameters

function	• Setting to indicate when modem should start an automatic GNSS fix – 0 - disabled – 1 - At bootup – 2 - When NMEA port is opened

set_function	• 0 - do not set to modem • 1 - set to modem
fix_type	• Type of GNSS fix: – 1 - Default Engine mode – 2 - MS-Based – 3 - MS-Assisted – 4 - Standalone
set_fix_type	• 0 - do not set to modem • 1 - set to modem
max_time	• Maximum time allowed for the receiver to get a fix in seconds • Valid range: 1-255
set_max_time	• 0 - do not set to modem • 1 - set to modem
max_dist	• Maximum uncertainty of a fix measured by distance in meters • Valid range: 1 - 4294967280
set_max_dist	• 0 - do not set to modem • 1 - set to modem
fix_rate	• Time between fixes in seconds • Valid range: 1–65535
set_fix_rate	• 0 - do not set to modem • 1 - set to modem

8.908.2 Field Documentation

8.908.2.1 ULONG SwiLocSetAutoStartReq::fix_rate

8.908.2.2 BYTE SwiLocSetAutoStartReq::fix_type

8.908.2.3 BYTE SwiLocSetAutoStartReq::function

8.908.2.4 ULONG SwiLocSetAutoStartReq::max_dist

8.908.2.5 BYTE SwiLocSetAutoStartReq::max_time

8.908.2.6 BOOL SwiLocSetAutoStartReq::set_fix_rate

8.908.2.7 BOOL SwiLocSetAutoStartReq::set_fix_type

8.908.2.8 BOOL SwiLocSetAutoStartReq::set_function

8.908.2.9 BOOL SwiLocSetAutoStartReq::set_max_dist

8.908.2.10 BOOL SwiLocSetAutoStartReq::set_max_time

8.909 swiModemStatusResp Struct Reference

Data Fields

- CommInfo commonInfo
- LTEInfo * pLTEInfo

8.909.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

Parameters

commonInfo	(mandatory) • See CommInfo for more information
pLTEInfo	(optional) • See LTEInfo for more information

8.909.2 Field Documentation

8.909.2.1 CommInfo swiModemStatusResp::commonInfo

8.909.2.2 LTEInfo* swiModemStatusResp::pLTEInfo

8.910 SwiOTAMsg_s Struct Reference

Data Fields

- ULONG type
- WORD data_len
- BYTE data [2048]
- LteNasReleaseInfo * pLteNasRelInfo
- ULONGLONG * pTime

8.910.1 Detailed Description

This structure contains OTA message

Parameters

type	• message type – 0 - LTE ESM uplink – 1 - LTE ESM downlink – 2 - LTE EMM uplink – 3 - LTE EMM downlink – 4 - GSM/UMTS uplink – 5 - GSM/UMTS downlink
data_len	• OTA Message Content Length
data	• OTA Message Content
pLteNasRelInfo	• LTE NAS Release Info • see LteNasReleaseInfo for details
pTime	• Seconds in local time since Jan. 6th 1980 00:00:00 UTC

8.910.2 Field Documentation

8.910.2.1 BYTE SwiOTAMsg_s::data[2048]

8.910.2.2 WORD SwiOTAMsg_s::data_len

8.910.2.3 LteNasReleaseInfo* SwiOTAMsg_s::pLteNasRelInfo

8.910.2.4 ULONGLONG* SwiOTAMsg_s::pTime

8.910.2.5 ULONG SwiOTAMsg_s::type

8.911 swiPDPRuntimeSettingsReq Struct Reference

Data Fields

- BYTE contextId
- BYTE contextType

8.911.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

Parameters

contextId	• Context Identifier

v4sessionId	• The v4 session ID for which the runtime settings are to be retrieved • provide a NULL pointer if not applicable
v6sessionId	• The v6 session ID for which the runtime settings are to be retrieved • provide a NULL pointer if not applicable

8.911.2 Field Documentation

8.911.2.1 BYTE swiPDPRuntimeSettingsReq::contextId

8.911.2.2 BYTE swiPDPRuntimeSettingsReq::contextType

8.912 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

8.912.1 Detailed Description

This structure contains the response parameters retrieved by the API SLQSWdsSwiPDPRuntimeSettings

Parameters

pContextId	(optional) • Context Identifier – 0xFF - Not Available
pBearerId	(optional) • Bearer Identity • 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. – 0xFF - Not Available

pAPNName	(optional) • APN name associated with the context id – NULL terminated by default.
pIPv4Address	(optional) • IPv4 Address – 0xFFFF - Not Available
pIPv4GW-Address	(optional) • IPv4 Gateway Address – 0xFFFF - Not Available
pPrDNSIPv4-Address	(optional) • Primary DNS IPv4 Address – 0xFFFF - Not Available
pSeDNSIPv4-Address	(optional) • Secondary DNS IPv4 Address – 0xFFFF - Not Available
pIPv6Address	(optional) • IPv6 Address • See IPV6AddressInfo for more information
pIPv6GW-Address	(optional) • IPv6 Gateway Address • See IPV6AddressInfo for more information
pPrDNSIPv6-Address	(optional) • Primary IPv6 DNS Address(in network byte order) • This is an 8-element array of 16-bit numbers, each of which is in big-endian format
pSeDNSIPv6-Address	(optional) • Secondary IPv6 DNS Address(in network byte order) • This is an 8-element array of 16-bit numbers, each of which is in big-endian format
pPrPCSCFIPv4-Address	(optional) • Primary PCSCF IPv4 Address
pSePCSCFIPv4-Address	(optional) • Secondary PCSCF IPv4 Address
pPrPCSCFIPv6-Address	(optional) • Primary PCSCF IPv6 Address • This is an 8-element array of 16-bit numbers, each of which is in big-endian format
pSePCSCFIPv6-Address	(optional) • Secondary PCSCF IPv6 Address • 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.

8.912.2 Field Documentation

8.912.2.1 CHAR* swiPDPRuntimeSettingsResp::pAPNName

8.912.2.2 BYTE* swiPDPRuntimeSettingsResp::pBearerId

8.912.2.3 BYTE* swiPDPRuntimeSettingsResp::pContextId

8.912.2.4 ULONG* swiPDPRuntimeSettingsResp::pIPv4Address

8.912.2.5 ULONG* swiPDPRuntimeSettingsResp::pIPv4GWAddress

8.912.2.6 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6Address

8.912.2.7 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6GWAddress

8.912.2.8 ULONG* swiPDPRuntimeSettingsResp::pPrDNSIPv4Address

8.912.2.9 WORD* swiPDPRuntimeSettingsResp::pPrDNSIPv6Address

8.912.2.10 ULONG* swiPDPRuntimeSettingsResp::pPrPCSCFIPv4Address

8.912.2.11 WORD* swiPDPRuntimeSettingsResp::pPrPCSCFIPv6Address

8.912.2.12 ULONG* swiPDPRuntimeSettingsResp::pSeDNSIPv4Address

8.912.2.13 WORD* swiPDPRuntimeSettingsResp::pSeDNSIPv6Address

8.912.2.14 ULONG* swiPDPRuntimeSettingsResp::pSePCSCFIPv4Address

8.912.2.15 WORD* swiPDPRuntimeSettingsResp::pSePCSCFIPv6Address

8.913 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

8.913.1 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 • 0x04 – IPv4 • 0x06 – Ipv6
pIPv4SrcAddr	IPv4 filter soruce address See IPv4Addr for more information • Implemented only for unsolicited indication
pIPv4DstAddr	IPv4 filter destination address See IPv4Addr for more information • 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: • 0x01 = ICMP • 0x06 = TCP • 0x11 = UDP • 0x32 = ESP Note: The next header protocol field will be set to 0xFD (TCP & UDP) if a TFT is received specifying a source or destination port number, but IP next header type is not specified.
pTos	IPv4 filter type of service See Tos for more information
pIPv6SrcAddr	IPv6 filter soruce address See IPv6Addr for more information • Implemented only for unsolicited indication
pIPv6DstAddr	IPv6 filter destination address See IPv6Addr for more information • 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) • Implemented only for unsolicited indication
pTCPSrcPort	TCP filter source port filter See Port for more information • Implemented only for unsolicited indication
pTCPDstPort	TCP filter destination port filter See Port for more information • Implemented only for unsolicited indication
pUDPSrcPort	UDP filter source port filter See Port for more information • Implemented only for unsolicited indication

pUDPDstPort	UDP filter destination port filter See Port for more information • 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 • 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 • Implemented only for unsolicited indication
pUDPDstPort	Transport protocol filter destination port See Port for more information • Implemented only for unsolicited indication

8.913.2 Field Documentation

8.913.2.1 BYTE swiQosFilter::index

8.913.2.2 ULONG* swiQosFilter::pEspSpi

8.913.2.3 WORD* swiQosFilter::pId

8.913.2.4 IPv4Addr* swiQosFilter::pIPv4DstAddr

8.913.2.5 IPv4Addr* swiQosFilter::pIPv4SrcAddr

8.913.2.6 IPv6Addr* swiQosFilter::pIPv6DstAddr

8.913.2.7 ULONG* swiQosFilter::pIPv6Label

8.913.2.8 IPv6Addr* swiQosFilter::pIPv6SrcAddr

8.913.2.9 IPv6TrafCls* swiQosFilter::pIPv6TrafCls

8.913.2.10 BYTE* swiQosFilter::pNxtHdrProto

8.913.2.11 WORD* swiQosFilter::pPrecedence

8.913.2.12 Port* swiQosFilter::pTCPDstPort

8.913.2.13 Port* swiQosFilter::pTCPSrcPort

8.913.2.14 Tos* swiQosFilter::pTos

8.913.2.15 Port* swiQosFilter::pTranDstPort

8.913.2.16 Port* swiQosFilter::pTranSrcPort

8.913.2.17 Port* swiQosFilter::pUDPDstPort

8.913.2.18 Port* swiQosFilter::pUDPSrcPort

8.913.2.19 BYTE swiQosFilter::version

8.914 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

8.914.1 Detailed Description

This structure contains the QoS Flow Request

Parameters

index	• IP flow index • Integer that uniquely identifies each flow instance • Unique index must be assigned by the control point to every flow_spec instance
pProfileId3GPP2	• IP flow 3GPP2 profile ID • A profile ID is shorthand for a defined set of QoS flow parameters specified by the network; to be present while requesting QoS for a CDMA device
p3GPP2Pri	• IP flow 3GPP2 flow priority • Flow priority used by the network in case of contention between flows with same QoS; this parameter applies for CDMA devices
pTrafficClass	• IP flow traffic class • Integer that designates the requested traffic class: • 0 – Conversational • 1 – Streaming • 2 – Interactive • 3 – Background

pDataRate	• IP flow data rate min max • See dataRate for more information
pTokenBucket	• IP flow data rate token bucket • See tokenBucket for more information
pLatency	• IP flow latency • Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link
pJitter	• IP flow jitter • Difference between the maximum and minimum latency (in milliseconds) that can be tolerated by an IP packet during the transfer through the wireless link
pPktErrRate	• IP flow packet error rate • See pktErrRate for more information
pMinPolicedPkt-Sz	• IP flow minimum policed packet size • Integer that defines the minimum packet size (in bytes) that will be policed for QoS guarantees; any IP packets that are smaller than the minimum specified policed size may not receive requested QoS
pMaxAllowed-PktSz	• IP flow maximum allowed packet size • Integer that defines the maximum packet size (in bytes) allowed in the IP flow; any IP packets greater in size than the maximum allowed packet size are not queued for transmission
p3GPPRes-ResidualBER	• IP flow 3GPP residual bit error rate • residual_bit_error_rate • 0 = 5∗10-2 residual BER • 1 = 1∗10-2 residual BER • 2 = 5∗10-3 residual BER • 3 = 4∗10-3 residual BER • 4 = 1∗10-3 residual BER • 5 = 1∗10-4 residual BER • 6 = 1∗10-5 residual BER • 7 = 1∗10-6 residual BER • 8 = 6∗10-8 residual BER • Integer that indicates the undetected BER for each IP flow in the delivered packets; Tapplies only to 3GPP networks

p3GPPTraHdlPri	• 3GPP traffic handling priority • 0 – Relative traffic handling priority 1 • 1 – Relative traffic handling priority 2 • 2 – Relative traffic handling priority 3 • Defines the relative priority of the flow; applies only to 3GPP networks
p3GPPImCn	• IP flow 3GPP IM CN flag • IM CN subsystem signaling flag: • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks
p3GPPSigInd	• IP flow 3GPP signaling indication • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks
pLteQci	• LTE QoS Class Identifier • QoS Class Identifier(QCI) is a required parameter to request QoS in LTE • QCI values: – QCI value 0 requests the network to assign the appropriate QCI value – QCI values 1-4 are associated with guaranteed bitrates – QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored

8.914.2 Field Documentation

8.914.2.1 BYTE swiQosFlow::index

8.914.2.2 BYTE* swiQosFlow::p3GPP2Pri

8.914.2.3 BYTE* swiQosFlow::p3GPPImCn

8.914.2.4 WORD* swiQosFlow::p3GPPResResidualBER

8.914.2.5 BYTE* swiQosFlow::p3GPPSigInd

8.914.2.6 BYTE* swiQosFlow::p3GPPTraHdlPri

8.914.2.7 dataRate* swiQosFlow::pDataRate

8.914.2.8 ULONG* swiQosFlow::pJitter

8.914.2.9 ULONG* swiQosFlow::pLatency

8.914.2.10 BYTE* swiQosFlow::pLteQci

8.914.2.11 ULONG* swiQosFlow::pMaxAllowedPktSz

8.914.2.12 ULONG* swiQosFlow::pMinPolicedPktSz

8.914.2.13 pktErrRate* swiQosFlow::pPktErrRate

8.914.2.14 WORD* swiQosFlow::pProfileId3GPP2

8.914.2.15 tokenBucket* swiQosFlow::pTokenBucket

8.914.2.16 BYTE* swiQosFlow::pTrafficClass

8.915 swiQosGranted Struct Reference

Data Fields

- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow

8.915.1 Detailed Description

This structure contains the QoS granted flow

Parameters

pTxFlow	See swiQosFlow for more information
pRxFlow	See swiQosFlow for more information

8.915.2 Field Documentation

8.915.2.1 swiQosFlow* swiQosGranted::pRxFlow

8.915.2.2 swiQosFlow* swiQosGranted::pTxFlow

8.916 swiQosIds Struct Reference

Data Fields

- BYTE sz
- ULONG * pIds

8.916.1 Detailed Description

This structure contains the QoS Response parameters.

Parameters

sz	Number of QoS identifiers
pIds	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

8.916.2 Field Documentation

8.916.2.1 ULONG* swiQosIds::pIds

8.916.2.2 BYTE swiQosIds::sz

8.917 swiQosModifyReq Struct Reference

Data Fields

- ULONG id
- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow
- swiQosFilter * pTxFilter
- swiQosFilter * pRxFilter

8.917.1 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

8.917.2 Field Documentation

8.917.2.1 ULONG swiQosModifyReq::id

8.917.2.2 swiQosFilter* swiQosModifyReq::pRxFilter

8.917.2.3 swiQosFlow* swiQosModifyReq::pRxFlow

8.917.2.4 swiQosFilter* swiQosModifyReq::pTxFilter

8.917.2.5 swiQosFlow* swiQosModifyReq::pTxFlow

8.918 swiQosReq Struct Reference

Data Fields

- BYTE index
- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow
- swiQosFilter * pTxFilter
- swiQosFilter * pRxFilter

8.918.1 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

8.918.2 Field Documentation

8.918.2.1 BYTE swiQosReq::index

8.918.2.2 swiQosFilter* swiQosReq::pRxFilter

8.918.2.3 swiQosFlow* swiQosReq::pRxFlow

8.918.2.4 swiQosFilter* swiQosReq::pTxFilter

8.918.2.5 swiQosFlow* swiQosReq::pTxFlow

8.919 swiRMTrasnferStaticsReq Struct Reference

Data Fields

- BYTE bResetStatistics
- ULONG ulMask

8.919.1 Detailed Description

RM Transfer Satistics Structure

Parameters

bResetStatistics	• Reset Statistics • Values: • 0 - Not Reset • Other - Reset

ulMask	• Enable/Disable RM Transfer Satatistics Indiscation Mask • Bit 0: Tx Packet Ok • Bit 1: Rx Packet Ok • Bit 2: Tx Bytes Ok • Bit 3: Rx Bytes Ok • Bit 4: Tx Packets Dropped • Bit 5: Rx Packets Dropped • Value: -0 - Disable -1 - Enable

8.919.2 Field Documentation

8.919.2.1 BYTE swiRMTrasnferStaticsReq::bResetStatistics

8.919.2.2 ULONG swiRMTrasnferStaticsReq::ulMask

8.920 sysInfoCommon Struct Reference

Data Fields

- BYTE srvDomainValid
- BYTE srvDomain
- BYTE srvCapabilityValid
- BYTE srvCapability
- BYTE roamStatusValid
- BYTE roamStatus
- BYTE isSysForbiddenValid
- BYTE isSysForbidden

8.920.1 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. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
srvDomain	• Service domain registered on the system. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available

srvCapability-Valid	• Indicates whether the service capability is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
srvCapability	• Current system's service capability. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
roamStatusValid	• Indicates whether the roaming status is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
roamStatus	• Current roaming status. – 0x00 - Off – 0x01 - On – 0x02 - Blinking – 0x03 - Out of the neighborhood – 0x04 - Out of the building – 0x05 - Preferred system – 0x06 - Available system – 0x07 - Alliance partner – 0x08 - Premium partner – 0x09 - Full service – 0x0A - Partial service – 0x0B - Banner is on – 0x0C - Banner is off – 0x0D to 0x3F - Reserved for Standard Enhanced Roaming Indicator Numbers – 0x40 to 0x7F - Reserved for Non-Standard Enhanced Roaming Indicator Numbers – 0x40 to 0xFF - Reserved. – 0xFF - Not Available • Values from 0x02 onward are only applicable for 3GPP2
isSysForbidden-Valid	• Indicates whether the forbidden system is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

isSysForbidden	• Whether the system is forbidden. – 0x00 - Not forbidden – 0x01 - Forbidden – 0xFF - Not Available

8.920.2 Field Documentation

8.920.2.1 BYTE sysInfoCommon::isSysForbidden

8.920.2.2 BYTE sysInfoCommon::isSysForbiddenValid

8.920.2.3 BYTE sysInfoCommon::roamStatus

8.920.2.4 BYTE sysInfoCommon::roamStatusValid

8.920.2.5 BYTE sysInfoCommon::srvCapability

8.920.2.6 BYTE sysInfoCommon::srvCapabilityValid

8.920.2.7 BYTE sysInfoCommon::srvDomain

8.920.2.8 BYTE sysInfoCommon::srvDomainValid

8.921 t_gpsTime Struct Reference

Data Fields

- USHORT gpsWeek
- ULONG gpsTimeOfWeekMs

8.921.1 Field Documentation

8.921.1.1 ULONG t_gpsTime::gpsTimeOfWeekMs

8.921.1.2 USHORT t_gpsTime::gpsWeek

8.922 t_sensor Struct Reference

Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

8.922.1 Field Documentation

8.922.1.1 ULONG t_sensor::aidingIndicatorMask

8.922.1.2 ULONG t_sensor::usageMask

8.923 t_Sv Struct Reference

Data Fields

- BYTE len
- USHORT entries [255]

8.923.1 Field Documentation

8.923.1.1 USHORT t_Sv::entries[255]

8.923.1.2 BYTE t_Sv::len

8.924 TDSCDMAECIOThresh Struct Reference

Data Fields

- BYTE TDSCDMAECIOThreshListLen
- ULONG * pTDSCDMAECIOThreshList

8.924.1 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 • Maximum of 32 values.

8.924.2 Field Documentation

8.924.2.1 ULONG* TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.924.2.2 BYTE TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.925 TDSCDMARSCPThresh Struct Reference

Data Fields

- BYTE TDSCDMARSCPThreshListLen
- WORD * pTDSCDMARSCPThreshList

8.925.1 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) • Maximum of 32 values • Range for RSCP values: -120 to -25 (in dBm).

8.925.2 Field Documentation

8.925.2.1 WORD* TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.925.2.2 BYTE TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.926 TDSCDMARSSIThresh Struct Reference

Data Fields

- BYTE TDSCDMARSSIThreshListLen
- ULONG * pTDSCDMARSSIThreshList

8.926.1 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 • Maximum of 32 values.

8.926.2 Field Documentation

8.926.2.1 ULONG* TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.926.2.2 BYTE TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.927 tdscdmaSigInfoExt Struct Reference

Data Fields

- float rssi
- float rscp
- float ecio
- float sinr

8.927.1 Detailed Description

Parameters

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

8.927.2 Field Documentation

8.927.2.1 float tdscdmaSigInfoExt::ecio

8.927.2.2 float tdscdmaSigInfoExt::rscp

8.927.2.3 float tdscdmaSigInfoExt::rssi

8.927.2.4 float tdscdmaSigInfoExt::sinr

8.928 TDSCDMASigInfoExt Struct Reference

Data Fields

- FLOAT rssi
- FLOAT rscp
- FLOAT ecio
- FLOAT sinr

8.928.1 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

8.928.2 Field Documentation

8.928.2.1 FLOAT TDSCDMASigInfoExt::ecio

8.928.2.2 FLOAT TDSCDMASigInfoExt::rscp

8.928.2.3 FLOAT TDSCDMASigInfoExt::rssi

8.928.2.4 FLOAT TDSCDMASigInfoExt::sinr

8.929 TDSCDMASINRCONFThresh Struct Reference

Data Fields

- BYTE TDSCDMASINRCONFThreshListLen
- FLOAT * pTDSCDMASINRCONFThreshList

8.929.1 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 • Maximum of 32 values

8.929.2 Field Documentation

8.929.2.1 FLOAT* TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

8.929.2.2 BYTE TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

8.930 TDSCDMASINRThresh Struct Reference

Data Fields

- BYTE TDSCDMASINRThreshListLen
- ULONG * pTDSCDMASINRThreshList

8.930.1 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 • Maximum of 32 values

8.930.2 Field Documentation

8.930.2.1 ULONG* TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.930.2.2 BYTE TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.931 tempData_t Struct Reference

Data Fields

- uint32_t timeSource
- uint32_t timeOfFirstSample
- uint8_t temperatureDataLen
- uint16_t timeOffset [64]
- uint32_t temperature [64]

8.931.1 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 • Valid values – 0 - Sensor time source is unspecified – 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. • The time stamp is in the time reference scale that is used by the sensor time source. • Units - Milliseconds
temperature-DataLen	• Number of sets of the following elements – timeOffset – temperature
timeOffset	• Sample time offset • Units - Milliseconds
temperature	• Sensor temperature. • Type - Floating point • Units - Degrees Celsius • Range -50 to +100.00

8.931.2 Field Documentation

8.931.2.1 uint32_t tempData_t::temperature[64]

8.931.2.2 uint8_t tempData_t::temperatureDataLen

8.931.2.3 uint32_t tempData_t::timeOfFirstSample

8.931.2.4 uint16_t tempData_t::timeOffset[64]

8.931.2.5 uint32_t tempData_t::timeSource

8.932 tempratureData Struct Reference

Data Fields

- ULONG timeSource
- ULONG timeOfFirstSample
- BYTE temperatureDataLen
- WORD timeOffset [64]
- ULONG temperature [64]

8.932.1 Detailed Description

This structure specifies information regarding the Temperature Data.

Parameters

timeSource	• Time source of the sensor data • Valid values – 0 - Sensor time source is unspecified – 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. • The time stamp is in the time reference scale that is used by the sensor time source. • Units - Milliseconds
temperature-DataLen	• Number of sets of the following elements – timeOffset – temperature
timeOffset	• Sample time offset • Units - Milliseconds
temperature	• Sensor temperature. • Type - Floating point • Units - Degrees Celsius • Range -50 to +100.00

8.932.2 Field Documentation

8.932.2.1 ULONG tempratureData::temperature[64]

8.932.2.2 BYTE tempratureData::temperatureDataLen

8.932.2.3 ULONG tempratureData::timeOfFirstSample

8.932.2.4 WORD tempratureData::timeOffset[64]

8.932.2.5 ULONG tempratureData::timeSource

8.933 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

8.933.1 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 – 4 - IPv4 – 6 - IPv6
sourceIP	• Source IP address – IPv4 - Fill the first 4 bytes – 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

8.933.2 Field Documentation

8.933.2.1 WORD TFTIDParams::destPortRangeEnd

8.933.2.2 WORD TFTIDParams::destPortRangeStart

8.933.2.3 BYTE TFTIDParams::eValid

8.933.2.4 BYTE TFTIDParams::filterId

8.933.2.5 ULONG TFTIDParams::flowLabel

8.933.2.6 ULONG TFTIDParams::IPSECSPI

8.933.2.7 BYTE TFTIDParams::ipVersion

8.933.2.8 BYTE TFTIDParams::nextHeader

8.933.2.9 WORD* TFTIDParams::pSourceIP

8.933.2.10 BYTE TFTIDParams::sourceIPMask

8.933.2.11 WORD TFTIDParams::srcPortRangeEnd

8.933.2.12 WORD TFTIDParams::srcPortRangeStart

8.933.2.13 WORD TFTIDParams::tosMask

8.934 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

8.934.1 Detailed Description

This structure contains the parameters for Network Time.

Parameters

year	• Year
month	• Month • 1 is January and 12 is December
day	• Day • Range - 1 to 31
hour	• Hour • Range - 0 to 59
minute	• Minute • Range - 0 to 59
second	• Second • Range - 0 to 59
dayOfWeek	• Day of the week • 0 is Monday and 6 is Sunday

timeZone	• Offset from Universal time • The difference between local time and Universal time, in increments of 15 min • Signed Value
dayLtSavingAdj	• Daylight saving adjustment in hours • Possible values - 0, 1, and 2. • This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVDO
radioInterface	• Radio interface from which the information comes • Values – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA -TD-SCDMA
TlvPresent	• Tlv Present.

8.934.2 Field Documentation

8.934.2.1 BYTE timeInfo::day

8.934.2.2 BYTE timeInfo::dayLtSavingAdj

8.934.2.3 BYTE timeInfo::dayOfWeek

8.934.2.4 BYTE timeInfo::hour

8.934.2.5 BYTE timeInfo::minute

8.934.2.6 BYTE timeInfo::month

8.934.2.7 BYTE timeInfo::radioInterface

8.934.2.8 BYTE timeInfo::second

8.934.2.9 INT8 timeInfo::timeZone

8.934.2.10 BYTE timeInfo::TlvPresent

8.934.2.11 WORD timeInfo::year

8.935 TmdDeRegNotMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.935.1 Detailed Description

This structure contains mitigation devices Level deregister request parameters

Parameters

mitigationDevID-Len	• Number of sets of the following elements – mitigation_dev_id
mitigationDevID	• Mitigation device ID

8.935.2 Field Documentation

8.935.2.1 CHAR TmdDeRegNotMitigationLvlReq::mitigationDevID[255]

8.935.2.2 BYTE TmdDeRegNotMitigationLvlReq::mitigationDevIDLen

8.936 TmdGetMitigationDevListResp Struct Reference

Data Fields

- BYTE * pMitigationDevListLen
- mitigationDevList * pMitigationDevList

8.936.1 Detailed Description

This structure contains mitigation devices list from the remote endpoint

Parameters

pMitigationDev-ListLen	• Mitigation Device List Length (Optional) • Number of sets of the following elements • pMitigationDevList
pMitigationDev-List	• Mitigation Device List (Optional) • See mitigationDevList for more information.

8.936.2 Field Documentation

8.936.2.1 mitigationDevList* TmdGetMitigationDevListResp::pMitigationDevList

8.936.2.2 BYTE* TmdGetMitigationDevListResp::pMitigationDevListLen

8.937 TmdGetMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.937.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

mitigationDevID-Len	• Number of sets of the following elements – mitigation_dev_id
mitigationDevID	• Mitigation device ID

8.937.2 Field Documentation

8.937.2.1 CHAR TmdGetMitigationLvlReq::mitigationDevID[255]

8.937.2.2 BYTE TmdGetMitigationLvlReq::mitigationDevIDLen

8.938 TmdGetMitigationLvlResp Struct Reference

Data Fields

- BYTE * pCurrentmitigationLvl
- BYTE * pReqMitigationLvl

8.938.1 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) • The requested thermal mitigation level from the client. The default is zero if the client has not previously set the mitigation level.

8.938.2 Field Documentation

8.938.2.1 BYTE* TmdGetMitigationLvlResp::pCurrentmitigationLvl

8.938.2.2 BYTE* TmdGetMitigationLvlResp::pReqMitigationLvl

8.939 TmdMitigationLvlIndReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.939.1 Detailed Description

This structure contains mitigation Level Indication request parameters

Parameters

mitigationDevID-Len	• Number of sets of the following elements – mitigation_dev_id
mitigationDevID	• Mitigation device ID

8.939.2 Field Documentation

8.939.2.1 CHAR TmdMitigationLvlIndReq::mitigationDevID[255]

8.939.2.2 BYTE TmdMitigationLvlIndReq::mitigationDevIDLen

8.940 TmdRegNotMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.940.1 Detailed Description

This structure contains mitigation devices Level register request parameters

Parameters

mitigationDevID-Len	• Number of sets of the following elements – mitigation_dev_id
mitigationDevID	• Mitigation device ID

8.940.2 Field Documentation

8.940.2.1 CHAR TmdRegNotMitigationLvlReq::mitigationDevID[255]

8.940.2.2 BYTE TmdRegNotMitigationLvlReq::mitigationDevIDLen

8.941 tokenBucket Struct Reference

Data Fields

- ULONG peakRate
- ULONG tokenRate
- ULONG bucketSz

8.941.1 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

8.941.2 Field Documentation

8.941.2.1 ULONG tokenBucket::bucketSz

8.941.2.2 ULONG tokenBucket::peakRate

8.941.2.3 ULONG tokenBucket::tokenRate

8.942 Tos Struct Reference

Data Fields

- BYTE val
- BYTE mask

8.942.1 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: • IPv4_filter_tos_val = 00101000 • IPv4_filter_tos_mask = 11111100 The filter will compare only the first 6 bits in the IPv4_filter_-type_of_service with the first 6 bits in the TOS field of the IP packet. The first 6 bits in the TOS field of the IP packet must be 001010 to match the filter. The last 2 bits can be anything since they are ignored by filtering.

8.942.2 Field Documentation

8.942.2.1 BYTE Tos::mask

8.942.2.2 BYTE Tos::val

8.943 transferRouteMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSTransferRouteMTMessageInfo TransferRouteMTMessageInfo

8.943.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
TransferRouteM-TMessageInfo	• Transfer Route MT Message • See sMSTransferRouteMTMessageInfo for more information

8.943.2 Field Documentation

8.943.2.1 uint8_t transferRouteMessageTlv::TlvPresent

8.943.2.2 sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo

8.944 TransferStatInd Struct Reference

Data Fields

- BYTE StatsPeriod
- ULONG StatsMask

8.944.1 Detailed Description

This structure contains Transfer Statistcics Indicator

Parameters

StatsPeriod	• Period between transfer statistics reports – 0 - Do not report – 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. – 0x00000040 - Tx bytes OK – 0x00000080 - Rx bytes OK

8.944.2 Field Documentation

8.944.2.1 ULONG TransferStatInd::StatsMask

8.944.2.2 BYTE TransferStatInd::StatsPeriod

8.945 transferStatInd Struct Reference

Data Fields

- uint8_t StatsPeriod
- uint32_t StatsMask

8.945.1 Detailed Description

Parameters

StatsPeriod	Field Period between transfer statistic reports.
StatsMask	requested statistic bit mask.

8.945.2 Field Documentation

8.945.2.1 uint32_t transferStatInd::StatsMask

8.945.2.2 uint8_t transferStatInd::StatsPeriod

8.946 TransferStatsDataType Struct Reference

Data Fields

- BYTE interval

8.946.1 Field Documentation

8.946.1.1 BYTE TransferStatsDataType::interval

8.947 TrStatInd Struct Reference

Data Fields

- BYTE statsPeriod
- ULONG statsMask

8.947.1 Detailed Description

This structure contains the information about the Transfer Statistics Indicator parameters.

Parameters

statsPeriod	• Period between transfer statistics reports. – 0 - Do not report – Other - Period between reports (seconds)
statsMask	• Requested statistic bit 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 • Each bit set causes the corresponding optional information to be sent in SLQSWdsEventReport-CallBack. • All unlisted bits are reserved for future use and must be set to zero.

8.947.2 Field Documentation

8.947.2.1 ULONG TrStatInd::statsMask

8.947.2.2 BYTE TrStatInd::statsPeriod

8.948 trueIMSI Struct Reference

Data Fields

- BYTE mccT [3]
- WORD imsiT1112
- BYTE imsiTS1 [7]
- BYTE imsiTS2 [3]
- BYTE imsiTaddrNum

8.948.1 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 – 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 – 0xFF - Not Available

8.948.2 Field Documentation

8.948.2.1 WORD trueIMSI::imsiT1112

8.948.2.2 BYTE trueIMSI::imsiTaddrNum

8.948.2.3 BYTE trueIMSI::imsiTS1[7]

8.948.2.4 BYTE trueIMSI::imsiTS2[3]

8.948.2.5 BYTE trueIMSI::mccT[3]

8.949 TXAGCList Struct Reference

Data Fields

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

8.949.1 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

8.949.2 Field Documentation

8.949.2.1 WORD* TXAGCList::pTXAIG

8.949.2.2 WORD* TXAGCList::pTXComprSlope

8.949.2.3 WORD* TXAGCList::pTXComprThres

8.949.2.4 WORD* TXAGCList::pTXExpSlope

8.949.2.5 WORD* TXAGCList::pTXExpThres

8.949.2.6 WORD* TXAGCList::pTXStaticGain

8.950 txInfo Struct Reference

Data Fields

- BYTE isInTraffic
- INT32 txPower

8.950.1 Detailed Description

This structure contains the Tx Information.

Parameters

isInTraffic	• Whether the device is in traffic. – 0x00 - not in traffic – 0x01 - in traffic • The txPower field is only meaningful when in the device is in traffic.
txPower	• Tx power value in 1/10 dbm.

8.950.2 Field Documentation

8.950.2.1 BYTE txInfo::isInTraffic

8.950.2.2 INT32 txInfo::txPower

8.951 TXPCMIIRFltr Struct Reference

Data Fields

- WORD * pFlag
- WORD * pStageCnt
- BYTE * pStage0Val
- BYTE * pStage1Val
- BYTE * pStage2Val
- BYTE * pStage3Val
- BYTE * pStage4Val

8.951.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_TXPCMIIRFLTR.

Parameters

pFlag	• Flag – 0x0000 - IIR filter disable – 0xffff - IIR filter enable
pStageCnt	• Stage Count – 0-4
pStage0Val	• A 20 BYTE sized parameter indicating Stage 0 value – A1 – A2 – B0 – B1 – B2
pStage1Val	• A 20 BYTE sized parameter indicating Stage 1 value – A1 – A2 – B0 – B1 – B2

pStage2Val	• A 20 BYTE sized parameter indicating Stage 2 value – A1 – A2 – B0 – B1 – B2
pStage3Val	• A 20 BYTE sized parameter indicating Stage 3 value – A1 – A2 – B0 – B1 – B2
pStage4Val	• A 20 BYTE sized parameter indicating Stage 4 value – A1 – A2 – B0 – B1 – B2

8.951.2 Field Documentation

8.951.2.1 WORD* TXPCMIIRFItr::pFlag

8.951.2.2 BYTE* TXPCMIIRFItr::pStage0Val

8.951.2.3 BYTE* TXPCMIIRFItr::pStage1Val

8.951.2.4 BYTE* TXPCMIIRFItr::pStage2Val

8.951.2.5 BYTE* TXPCMIIRFItr::pStage3Val

8.951.2.6 BYTE* TXPCMIIRFItr::pStage4Val

8.951.2.7 WORD* TXPCMIIRFItr::pStageCnt

8.952 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

8.952.1 Detailed Description

This structure contains Application Status Information loaded on the card.

Parameters

appType	• Indicates the type of the application. – 0 - Unknown – 1 - SIM card – 2 - USIM application – 3 - RUIM card – 4 - CSIM application – 5 - ISIM application • Other values are reserved for the future and are to be handled as "Unknown".
appState	• Indicates the state of the application. – 0 - Unknown – 1 - Detected – 2 - PIN1 or UPIN is required – 3 - PUK1 or PUK for UPIN is required – 4 - Personalization state must be checked – 5 - PIN1 is blocked – 6 - Illegal – 7 - Ready
persoState	• Indicates the state of the personalization for the application. – 0 - Unknown – 1 - Personalization operation is in progress – 2 - Ready – 3 - Personalization code is required – 4 - PUK for personalization code is required – 5 - Permanently blocked

persoFeature	• Indicates the personalization feature. • This applies only when a personalization code is required to deactivate or unblock personalization. – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization – 11 - Unknown
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 • If zero(0) then no aidVal information exists.
aidVal[MAX_DESCRIPTION_LENGTH]	• Application identifier value.
univPin	• Indicates whether UPIN replaces PIN1. – 0 - PIN1 is used – 1 - UPIN replaces PIN1
pin1State	• Indicates the state of PIN1. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 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. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked
pin2Retries	• Indicates the number of retries remaining to verify PIN2.
puk2Retries	• Indicates the number of retries remaining to unblock PIN2.

8.952.2 Field Documentation

8.952.2.1 uint8_t uim_appStatus::aidLength

8.952.2.2 uint8_t uim_appStatus::aidVal[255]

8.952.2.3 uint8_t uim_appStatus::appState

8.952.2.4 uint8_t uim_appStatus::appType

8.952.2.5 uint8_t uim_appStatus::persoFeature

8.952.2.6 uint8_t uim_appStatus::persoRetries

8.952.2.7 uint8_t uim_appStatus::persoState

8.952.2.8 uint8_t uim_appStatus::persoUnblockRetries

8.952.2.9 uint8_t uim_appStatus::pin1Retries

8.952.2.10 uint8_t uim_appStatus::pin1State

8.952.2.11 uint8_t uim_appStatus::pin2Retries

8.952.2.12 uint8_t uim_appStatus::pin2State

8.952.2.13 uint8_t uim_appStatus::puk1Retries

8.952.2.14 uint8_t uim_appStatus::puk2Retries

8.952.2.15 uint8_t uim_appStatus::univPin

8.953 uim_authenticateResult Struct Reference

Data Fields

- uint16_t contentLen

- uint8_t content [1024]

8.953.1 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. • This value is a sequence of bytes returned from the card.

8.953.2 Field Documentation

8.953.2.1 uint8_t uim_authenticateResult::content[1024]

8.953.2.2 uint16_t uim_authenticateResult::contentLen

8.954 uim_authenticationData Struct Reference

Data Fields

- uint8_t context
- uint16_t dataLen
- uint8_t data [1024]

8.954.1 Detailed Description

This structure contains the Session Information.

Parameters

context	• Authenticate context. – 0 - Runs the GSM algorithm (valid only on a 2GSIM card) – 1 - Runs the CAVE algorithm (valid only on a RUIM card) – 2 - GSM security context (valid only on a USIM application) – 3 - 3G security context (valid only on a USIM application) – 4 - VGCS/VBS security context (valid only on aUSIM application) – 5 - GBA security context, Bootstrapping mode (valid only on a USIM or ISIM application) – 6 - GBA security context, NAF Derivation mode (valid only on a USIM or ISIM application) – 7 - MBMS security context, MSK Update mode (valid only on a USIM application) – 8 - MBMS security context, MTK Generation mode (valid only on a USIM application) – 9 - MBMS security context, MSK Deletion mode (valid only on a USIM application) – 10 - MBMS security context, MUK Deletion mode (valid only on a USIM application) – 11 - IMS AKA security context (valid only on aISIM application) – 12 - HTTP-digest security context (valid only onan ISIM application) – 13 - Compute IP authentication, CHAP (valid onlyon RUIM or CSIM) – 14 - Compute IP authentication, MN-HA authenticator (valid only on RUIM or CSIM) – 15 - Compute IP authentication, MIP-RRQ hash (valid only on RUIM or CSIM) – 16 - Compute IP authentication, MN-AAA authenticator (valid only on RUIM or CSIM) – 17 - Compute IP authentication, HRPD access authenticator (valid only on RUIM or CSIM) • 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.

8.954.2 Field Documentation

8.954.2.1 uint8_t uim_authenticationData::context

8.954.2.2 uint8_t uim_authenticationData::data[1024]

8.954.2.3 uint16_t uim_authenticationData::dataLen

8.955 uim_cardResult Struct Reference

Data Fields

- uint8_t sw1
- uint8_t sw2

8.955.1 Detailed Description

This structure contains the information about the card result.

Parameters

sw1	• SW1 received from the card.
sw2	• SW2 received from the card.

8.955.2 Field Documentation

8.955.2.1 uint8_t uim_cardResult::sw1

8.955.2.2 uint8_t uim_cardResult::sw2

8.956 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]

8.956.1 Detailed Description

This structure contains Card Status Information.

Parameters

indexGwPri	• Index of the primary GW provisioning application. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.
index1xPri	• Index of the primary 1X provisioning application. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.
indexGwSec	• Index of the secondary GW provisioning application. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.

index1xSec	• Index of the secondary GW provisioning application. • The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0). • The value 0xFFFF identifies when the session does not exist.
numSlot	• Indicates the number of slots available on the device. • The following block is repeated for each slot. i.e. cardState • If zero(0) then no cardState information exists.
SlotInfo	• See uim_slotInfo for more information.

8.956.2 Field Documentation

8.956.2.1 uint16_t uim_cardStatus::index1xPri

8.956.2.2 uint16_t uim_cardStatus::index1xSec

8.956.2.3 uint16_t uim_cardStatus::indexGwPri

8.956.2.4 uint16_t uim_cardStatus::indexGwSec

8.956.2.5 uint8_t uim_cardStatus::numSlot

8.956.2.6 uim_slotInfo uim_cardStatus::SlotInfo[5]

8.957 uim_changeUIMPIN Struct Reference

Data Fields

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

8.957.1 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. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN – 4 - Hidden key

oldPINLen	• Length of the following elements i.e. old pin value.
oldPINVal[MAX_DESCRIPTION_LENGTH]	• Old PIN value. • 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. • This value is a sequence of ASCII characters.

8.957.2 Field Documentation

8.957.2.1 uint8_t uim_changeUIMPIN::oldPINLen

8.957.2.2 uint8_t uim_changeUIMPIN::oldPINVal[255]

8.957.2.3 uint8_t uim_changeUIMPIN::pinID

8.957.2.4 uint8_t uim_changeUIMPIN::pinLen

8.957.2.5 uint8_t uim_changeUIMPIN::pinVal[255]

8.958 uim_depersonalizationInformation Struct Reference

Data Fields

- uint8_t feature
- uint8_t operation
- uint8_t ckLen
- uint8_t ckVal [255]

8.958.1 Detailed Description

This structure contains the Depersonalization Information.

Parameters

feature	• Indicates the personalization feature to de-activate or unblock. – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization
operation	• Indicates the operation to perform. – 0 - Deactivate personalization. – 1 - Unblock personalization.
ckLen	• Length of the following elements i.e. control key value.
ckVal[MAX_DESCRIPTION_LENGTH]	• Control key value. • This value is a sequence of ASCII characters.

8.958.2 Field Documentation

8.958.2.1 uint8_t uim_depersonalizationInformation::ckLen

8.958.2.2 uint8_t uim_depersonalizationInformation::ckVal[255]

8.958.2.3 uint8_t uim_depersonalizationInformation::feature

8.958.2.4 uint8_t uim_depersonalizationInformation::operation

8.959 uim_encryptedPIN1 Struct Reference

Data Fields

- uint8_t pin1Len
- uint8_t pin1Val [255]

8.959.1 Detailed Description

This structure contains the encrypted PIN1 Information.

Parameters

pin1Len	• Number of sets of the following elements ie encrypted PIN1 value. • 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.

8.959.2 Field Documentation

8.959.2.1 uint8_t uim_encryptedPIN1::pin1Len

8.959.2.2 uint8_t uim_encryptedPIN1::pin1Val[255]

8.960 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]

8.960.1 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. – 0 - Transparent – 1 - Cyclic – 2 - Linear fixed – 3 - Dedicated file – 4 - Master file
recordSize	• Indicates the size of the records. • Only for cyclic and linear fixed files
recordCount	• Indicates the total no. of the records. • Only for linear fixed files
secRead	• Read security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secReadMask	• Mask with read security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
secWrite	• Write security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secWriteMask	• Mask with write security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM

secIncrease	• Increase security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secIncrease-Mask	• Mask with increase security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
secDeactivate	• Deactivate security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secDeactivate-Mask	• Mask with deactivate security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
secActivate	• Activate security attributes. – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
secActivateMask	• Mask with activate security attributes. • This field is valid only when required by security attributes. – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
rawLen	• Length of the following elements i.e. raw value.

rawValue[MAX_-DESCRIPTION-_LENGTH]	• Raw value of file attributes.

8.960.2 Field Documentation

8.960.2.1 uint16_t uim_fileAttributes::fileID

8.960.2.2 uint16_t uim_fileAttributes::fileSize

8.960.2.3 uint8_t uim_fileAttributes::fileType

8.960.2.4 uint16_t uim_fileAttributes::rawLen

8.960.2.5 uint8_t uim_fileAttributes::rawValue[255]

8.960.2.6 uint16_t uim_fileAttributes::recordCount

8.960.2.7 uint16_t uim_fileAttributes::recordSize

8.960.2.8 uint8_t uim_fileAttributes::secActivate

8.960.2.9 uint16_t uim_fileAttributes::secActivateMask

8.960.2.10 uint8_t uim_fileAttributes::secDeactivate

8.960.2.11 uint16_t uim_fileAttributes::secDeactivateMask

8.960.2.12 uint8_t uim_fileAttributes::secIncrease

8.960.2.13 uint16_t uim_fileAttributes::secIncreaseMask

8.960.2.14 uint8_t uim_fileAttributes::secRead

8.960.2.15 uint16_t uim_fileAttributes::secReadMask

8.960.2.16 uint8_t uim_fileAttributes::secWrite

8.960.2.17 uint16_t uim_fileAttributes::secWriteMask

8.961 uim_fileInfo Struct Reference

Data Fields

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

8.961.1 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).

8.961.2 Field Documentation

8.961.2.1 uint16_t uim_fileInfo::fileID

8.961.2.2 uint16_t uim_fileInfo::path[255]

8.961.2.3 uint8_t uim_fileInfo::pathLen

8.962 uim_hotSwapStatus Struct Reference

Data Fields

- uint8_t hotSwapLength
- uint8_t hotSwap [255]

8.962.1 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. – 0 - Hot-swap is not supported – 1 - Hot-swap is supported, but the status of the switch is not supported – 2 - Switch indicates that the card is present – 3 - Switch indicates that the card is not present

8.962.2 Field Documentation

8.962.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

8.962.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

8.963 uim_personalizationStatus Struct Reference

Data Fields

- uint8_t numFeatures
- uint8_t feature [12]
- uint8_t verifyLeft [12]
- uint8_t unblockLeft [12]

8.963.1 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: – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization
verifyLeft	• Number of the remaining attempts to verify the personalization feature.
unblockLeft	• Number of the remaining attempts to unblock the personalization feature.

8.963.2 Field Documentation

8.963.2.1 uint8_t uim_personalizationStatus::feature[12]

8.963.2.2 uint8_t uim_personalizationStatus::numFeatures

8.963.2.3 uint8_t uim_personalizationStatus::unblockLeft[12]

8.963.2.4 uint8_t uim_personalizationStatus::verifyLeft[12]

8.964 uim_readResult Struct Reference

Data Fields

- uint16_t contentLen
- uint8_t content [255]

8.964.1 Detailed Description

This structure contains the information for write operation.

Parameters

contentLen	• Number of sets of content.
content[255]	• Read content. • The content is the sequence of bytes as read from the card.

8.964.2 Field Documentation

8.964.2.1 uint8_t uim_readResult::content[255]

8.964.2.2 uint16_t uim_readResult::contentLen

8.965 uim_readTransparentInfo Struct Reference

Data Fields

- uint16_t offset
- uint16_t length

8.965.1 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. • The value 0 is used to read the complete file.

8.965.2 Field Documentation

8.965.2.1 uint16_t uim_readTransparentInfo::length

8.965.2.2 uint16_t uim_readTransparentInfo::offset

8.966 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]

8.966.1 Detailed Description

This structure contains information of parameters associated with the Refresh Event.

Parameters

stage	• Indicates the stage of the Refresh procedure. – 0 - Waiting for OK to refresh – 1 - Refresh started – 2 - Refresh ended successfully – 3 - Refresh failed
mode	• Indicates the Refresh mode. – 0 - Reset – 1 - Init – 2 - Init and FCN – 3 - FCN – 4 - Init and Full FCN – 5 - Application reset – 6 - 3G session reset
sessionType	• Indicates the session type. – 0 - Primary GW provisioning – 1 - Primary 1X provisioning – 2 - Secondary GW provisioning – 3 - Secondary 1X provisioning – 4 - Nonprovisioning on slot 1 – 5 - Nonprovisioning on slot 2 – 6 - Card on slot 1 – 7 - Card on slot 2 – 8 - Logical channel on slot 1 – 9 - Logical channel on slot 2
aidLength	• Number of sets of the following elements – 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: – file_id – path_len – path
arrfileInfo	• Array of file Information struct

8.966.2 Field Documentation

8.966.2.1 uint8_t uim_refreshevent::aid[255]

8.966.2.2 uint8_t uim_refreshevent::aidLength

8.966.2.3 uim_fileInfo uim_refreshevent::arrfileInfo[255]

8.966.2.4 uint8_t uim_refreshevent::mode

8.966.2.5 uint16_t uim_refreshevent::numOfFiles

8.966.2.6 uint8_t uim_refreshevent::sessionType

8.966.2.7 uint8_t uim_refreshevent::stage

8.967 uim_registerRefresh Struct Reference

Data Fields

- uint8_t registerFlag
- uint8_t voteForInit
- uint16_t numFiles
- uim_fileInfo arrfileInfo [255]

8.967.1 Detailed Description

This structure contains parameters of refresh Information

Parameters

registerFlag	• Flag that indicates whether to register or deregister for refresh indications. Valid values: – 0 - Deregister – 1 - Register
voteForInit	• Flag that indicates whether to vote for the init when there is a refresh. Valid values: – 0 - Client does not vote for initialization – 1 - Client votes for initialization

numFiles	• Number of sets of the following elements: – file_id – path_len – path
arrfileInfo	• Array of file Information structure. • See uim_fileInfo for more information

8.967.2 Field Documentation

8.967.2.1 uim_fileInfo uim_registerRefresh::arrfileInfo[255]

8.967.2.2 uint16_t uim_registerRefresh::numFiles

8.967.2.3 uint8_t uim_registerRefresh::registerFlag

8.967.2.4 uint8_t uim_registerRefresh::voteForInit

8.968 uim_remainingRetries Struct Reference

Data Fields

- uint8_t verifyLeft
- uint8_t unblockLeft

8.968.1 Detailed Description

This structure contains the information about the retries remaining.

Parameters

verifyLeft	• Number of remaining attempts to verify the PIN. • 0xFF, if unavailable.
unblockLeft	• Number of remaining attempts to unblock the PIN. • 0xFF, if unavailable.

Note

This value is returned only when the enable/disable operation has failed. This information is not sent for a hidden key PIN type.

8.968.2 Field Documentation

8.968.2.1 uint8_t uim_remainingRetries::unblockLeft

8.968.2.2 uint8_t uim_remainingRetries::verifyLeft

8.969 uim_sessionInformation Struct Reference

Data Fields

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

8.969.1 Detailed Description

This structure contains the Session Information.

Parameters

sessionType	• Indicates the session type. – 0 - Primary GW provisioning – 1 - Primary 1X provisioning – 2 - Secondary GW provisioning – 3 - Secondary 1X provisioning – 4 - Non-provisioning on slot 1 – 5 - Non-provisioning on slot 2 – 6 - Card on slot 1 – 7 - Card on slot 2 – 8 - Logical channel on slot 1 – 9 - Logical channel on slot 2
aidLength	• Length of the following elements i.e. 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.

8.969.2 Field Documentation

8.969.2.1 uint8_t uim_sessionInformation::aid[255]

8.969.2.2 uint8_t uim_sessionInformation::aidLength

8.969.2.3 uint8_t uim_sessionInformation::sessionType

8.970 uim_setPINProtection Struct Reference

Data Fields

- uint8_t pinID
- uint8_t pinOperation

- uint8_t pinLength
- uint8_t pinValue [255]

8.970.1 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. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN – 4 - Hidden key
pinOperation	• Indicates whether the PIN is enabled or disabled. – 0 - Disable the PIN – 1 - Enable the PIN
pinLength	• Length of the following elements i.e. pin value.
pinValue[MAX_DESCRIPTION_LENGTH]	• PIN value. • This value is a sequence of ASCII characters.

8.970.2 Field Documentation

8.970.2.1 uint8_t uim_setPINProtection::pinID

8.970.2.2 uint8_t uim_setPINProtection::pinLength

8.970.2.3 uint8_t uim_setPINProtection::pinOperation

8.970.2.4 uint8_t uim_setPINProtection::pinValue[255]

8.971 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]

8.971.1 Detailed Description

This structure contains information about the SLOTS present.

Parameters

cardState	• Indicates the state of the card for each slot. – 0 - Absent – 1 - Present – 2 - Error
upinState	• Indicates the state of UPIN. – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked – 0xFF - Not Available
upinRetries	• Indicates the number of retries remaining to verify the UPIN. • If 0xFF, information not available.
upukRetries	• Indicates the number of retries remaining to unblock the UPIN. • If 0xFF, information not available.
errorState	• Indicates the reason for the card error, and is valid only when the card state is Error – 0 - Unknown – 1 - Power down – 2 - Poll error – 3 - No ATR received – 4 - Volt mismatch – 5 - Parity error – 6 - Unknown; possibly removed – 7 - Card returned technical problems – 0xFF - Not Available • Other values are possible and reserved for future use. • When an unknown value is received, it is to be handled as "Unknown".
numApp	• Indicates the number of applications available on the card. • The following block is repeated for each application. i.e. AppStatus. • If zero(0) then no AppStatus information exists.
AppStatus	• See uim_appStatus for more information.

8.971.2 Field Documentation

8.971.2.1 uim_appStatus uim_slotInfo::AppStatus[10]

8.971.2.2 uint8_t uim_slotInfo::cardState

8.971.2.3 uint8_t uim_slotInfo::errorState

8.971.2.4 uint8_t uim_slotInfo::numApp

8.971.2.5 uint8_t uim_slotInfo::upinRetries

8.971.2.6 uint8_t uim_slotInfo::upinState

8.971.2.7 uint8_t uim_slotInfo::upukRetries

8.972 uim_UIMSessionInformation Struct Reference

Data Fields

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

8.972.1 Detailed Description

This structure contains the Session Information.

Parameters

sessionType	• Indicates the session type. – 0 - Primary GW provisioning – 1 - Primary 1X provisioning – 2 - Secondary GW provisioning – 3 - Secondary 1X provisioning – 4 - Non-provisioning on slot 1 – 5 - Non-provisioning on slot 2 – 6 - Card on slot 1 – 7 - Card on slot 2 – 8 - Logical channel on slot 1 – 9 - Logical channel on slot 2
aidLength	• Length of the following elements i.e. 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.

8.972.2 Field Documentation

8.972.2.1 uint8_t uim_UIMSessionInformation::aid[255]

8.972.2.2 uint8_t uim_UIMSessionInformation::aidLength

8.972.2.3 uint8_t uim_UIMSessionInformation::sessionType

8.973 uim_unblockUIMPIN Struct Reference

Data Fields

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

8.973.1 Detailed Description

This structure contains the information about the unblock pin parameters.

Parameters

pinID	• Indicates the PIN ID to be changed. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN
pukLen	• Length of the following elements i.e. puk value.
pukVal[UIM_MAX_DESCRIPTION_LENGTH]	• PIN Unlock Key value. • 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. • This value is a sequence of ASCII characters.

8.973.2 Field Documentation

8.973.2.1 uint8_t uim_unblockUIMPIN::newPINLen

8.973.2.2 uint8_t uim_unblockUIMPIN::newPINVal[255]

8.973.2.3 uint8_t uim_unblockUIMPIN::pinID

8.973.2.4 uint8_t uim_unblockUIMPIN::pukLen

8.973.2.5 uint8_t uim_unblockUIMPIN::pukVal[255]

8.974 uim_verifyUIMPIN Struct Reference

Data Fields

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

8.974.1 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. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN – 4 - Hidden key
pinLen	• Length of the following elements i.e. pin value.
pinVal[MAX_DESCRIPTION_LENGTH]	• PIN value. • This value is a sequence of ASCII characters.

8.974.2 Field Documentation

8.974.2.1 uint8_t uim_verifyUIMPIN::pinID

8.974.2.2 uint8_t uim_verifyUIMPIN::pinLen

8.974.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

8.975 UIMAuthenticateReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- authenticationData authData
- ULONG * pIndicationToken

8.975.1 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. • 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.

8.975.2 Field Documentation

8.975.2.1 authenticationData UIMAuthenticateReq::authData

8.975.2.2 ULONG* UIMAuthenticateReq::pIndicationToken

8.975.2.3 UIMSessionInformation UIMAuthenticateReq::sessionInfo

8.976 UIMAuthenticateResp Struct Reference

Data Fields

- cardResult * pCardResult
- authenticateResult * pAuthenticateResult
- ULONG * pIndicationToken

8.976.1 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. • 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.

8.976.2 Field Documentation

8.976.2.1 authenticateResult* UIMAuthenticateResp::pAuthenticateResult

8.976.2.2 cardResult* UIMAuthenticateResp::pCardResult

8.976.2.3 ULONG* UIMAuthenticateResp::pIndicationToken

8.977 UIMChangePinReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- changeUIMPIN changePIN
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.977.1 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. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
pIndicationToken(optional)	• Response in Indication. • 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.

8.977.2 Field Documentation

8.977.2.1 changeUIMPIN UIMChangePinReq::changePIN

8.977.2.2 ULONG* UIMChangePinReq::pIndicationToken

8.977.2.3 BYTE* UIMChangePinReq::pKeyReferenceID

8.977.2.4 UIMSessionInformation UIMChangePinReq::sessionInfo

8.978 UIMDepersonalizationReq Struct Reference

Data Fields

- depersonalizationInformation depersonilisationInfo

8.978.1 Detailed Description

This structure contains information of the request parameters associated with a Depersonalization API.

Parameters

depersonilisation-Info	• See depersonalizationInformation for more information.

8.978.2 Field Documentation

8.978.2.1 depersonalizationInformation UIMDepersonalizationReq::depersonilisationInfo

8.979 UIMDepersonalizationResp Struct Reference

Data Fields

- remainingRetries * pRemainingRetries

8.979.1 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.

8.979.2 Field Documentation

8.979.2.1 remainingRetries* UIMDepersonalizationResp::pRemainingRetries

8.980 UIMEventRegisterReqResp Struct Reference

Data Fields

- ULONG eventMask

8.980.1 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. – Bit 0 - Card status – Bit 1 - SAP connection – Bit 4 - Physical Slot Status

8.980.2 Field Documentation

8.980.2.1 ULONG UIMEventRegisterReqResp::eventMask

8.981 UIMGetCardStatusResp Struct Reference

Data Fields

- cardStatus * pCardStatus
- hotSwapStatus * pHotSwapStatus

8.981.1 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.

8.981.2 Field Documentation

8.981.2.1 cardStatus* UIMGetCardStatusResp::pCardStatus

8.981.2.2 hotSwapStatus* UIMGetCardStatusResp::pHotSwapStatus

8.982 UIMGetConfigurationReq Struct Reference

Data Fields

- ULONG * pConfigurationMask

8.982.1 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 – Bit 0 - Automatic selection – Bit 1 - Personalization status – Bit 2 - Halt subscription – All other bits are reserved for future use

Note

- if the TLV is missing, the service returns all configuration items in the response.

8.982.2 Field Documentation

8.982.2.1 ULONG* UIMGetConfigurationReq::pConfigurationMask

8.983 UIMGetConfigurationResp Struct Reference

Data Fields

- BYTE * pAutoSelection
- personalizationStatus * pPersonalizationStatus
- BYTE * pHaltSubscription

8.983.1 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. • Valid values – 0 - Automatic provisioning is off – 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. • Valid values – 0 - Modem proceeds with publishing the subscription – 1 - Modem does not publish the subscription

8.983.2 Field Documentation

8.983.2.1 BYTE* UIMGetConfigurationResp::pAutoSelection

8.983.2.2 BYTE* UIMGetConfigurationResp::pHaltSubscription

8.983.2.3 personalizationStatus* UIMGetConfigurationResp::pPersonalizationStatus

8.984 UIMGetFileAttributesReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- ULONG * pIndicationToken

8.984.1 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. • 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.

8.984.2 Field Documentation

8.984.2.1 fileInfo UIMGetFileAttributesReq::fileIndex

8.984.2.2 ULONG* UIMGetFileAttributesReq::pIndicationToken

8.984.2.3 UIMSessionInformation UIMGetFileAttributesReq::sessionInfo

8.985 UIMGetFileAttributesResp Struct Reference

Data Fields

- cardResult * pCardResult
- fileAttributes * pFileAttributes
- ULONG * pIndicationToken

8.985.1 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. • 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.

8.985.2 Field Documentation

8.985.2.1 cardResult∗ UIMGetFileAttributesResp::pCardResult

8.985.2.2 fileAttributes∗ UIMGetFileAttributesResp::pFileAttributes

8.985.2.3 ULONG∗ UIMGetFileAttributesResp::pIndicationToken

8.986 UIMGetSlotsStatusResp Struct Reference

Data Fields

- BYTE ∗ pNumberOfPhySlot
- UIMSlotsStatus ∗ pUimSlotsStatus

8.986.1 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..

8.986.2 Field Documentation

8.986.2.1 BYTE∗ UIMGetSlotsStatusResp::pNumberOfPhySlot

8.986.2.2 UIMSlotsStatus* UIMGetSlotsStatusResp::pUimSlotsStatus

8.987 UIMPinResp Struct Reference

Data Fields

- remainingRetries * pRemainingRetries
- encryptedPIN1 * pEncryptedPIN1
- ULONG * pIndicationToken

8.987.1 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. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • 0xFFFFFFFF, if unavailable

Note

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

8.987.2 Field Documentation

8.987.2.1 encryptedPIN1* UIMPinResp::pEncryptedPIN1

8.987.2.2 ULONG* UIMPinResp::pIndicationToken

8.987.2.3 remainingRetries* UIMPinResp::pRemainingRetries

8.988 UIMPowerDownReq Struct Reference

Data Fields

- BYTE slot

8.988.1 Detailed Description

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

Parameters

slot	• Indicates the slot to be used. – 1 - Slot 1 – 2 - Slot 2

8.988.2 Field Documentation

8.988.2.1 BYTE UIMPowerDownReq::slot

8.989 UIMPowerUpReq Struct Reference

Data Fields

- BYTE slot
- BYTE * pIgnoreHotSwapSwitch

8.989.1 Detailed Description

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

Parameters

slot	• Indicates the slot to be used. – 1 - Slot 1 – 2 - Slot 2
pIgnoreHotSwapSwitch(optional)	• Hot-swap switch status. – 0 - Checks the hot-swap switch status – 1 - Ignores the hot-swap switch status

8.989.2 Field Documentation

8.989.2.1 BYTE* UIMPowerUpReq::pIgnoreHotSwapSwitch

8.989.2.2 BYTE UIMPowerUpReq::slot

8.990 UIMReadTransparentReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- readTransparentInfo readTransparent
- ULONG * pIndicationToken
- BYTE * pEncryptData

8.990.1 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. • When this TLV is present, it indicates that the result must be provided in a subsequent indication.
pEncrypt-Data(optional)	• Encrypt Data. • Indicates whether the data read from the card is to be encrypted.

Note

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

8.990.2 Field Documentation

8.990.2.1 fileInfo UIMReadTransparentReq::fileIndex

8.990.2.2 BYTE* UIMReadTransparentReq::pEncryptData

8.990.2.3 ULONG* UIMReadTransparentReq::pIndicationToken

8.990.2.4 readTransparentInfo UIMReadTransparentReq::readTransparent

8.990.2.5 UIMSessionInformation UIMReadTransparentReq::sessionInfo

8.991 UIMReadTransparentResp Struct Reference

Data Fields

- cardResult * pCardResult
- readResult * pReadResult
- ULONG * pIndicationToken
- BYTE * pEncryptedData

8.991.1 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. • When this TLV is present, it indicates that the result must be provided in a subsequent indication.
pEncrypted-Data(optional)	• Encrypted Data. • Indicates whether the data from the card passed in read_result is encrypted.

Note

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

8.991.2 Field Documentation

8.991.2.1 cardResult∗ UIMReadTransparentResp::pCardResult

8.991.2.2 BYTE∗ UIMReadTransparentResp::pEncryptedData

8.991.2.3 ULONG∗ UIMReadTransparentResp::pIndicationToken

8.991.2.4 readResult∗ UIMReadTransparentResp::pReadResult

8.992 UIMRefreshCompleteReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- BYTE refreshComplete

8.992.1 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: – 0 - Refresh was not completed successfully – 1 - Refresh was completed successfully

8.992.2 Field Documentation

8.992.2.1 BYTE UIMRefreshCompleteReq::refreshComplete

8.992.2.2 UIMSessionInformation UIMRefreshCompleteReq::sessionInfo

8.993 UIMRefreshEvent Struct Reference

Data Fields

- BYTE stage
- BYTE mode
- BYTE sessionType
- BYTE aidLength
- BYTE aid [255]
- WORD numOfFiles
- fileInfo arrfileInfo [255]

8.993.1 Detailed Description

This structure contains information of parameters associated with the Refresh Event.

Parameters

stage	• Indicates the stage of the Refresh procedure. – 0 - Waiting for OK to refresh – 1 - Refresh started – 2 - Refresh ended successfully – 3 - Refresh failed
mode	• Indicates the Refresh mode. – 0 - Reset – 1 - Init – 2 - Init and FCN – 3 - FCN – 4 - Init and Full FCN – 5 - Application reset – 6 - 3G session reset

sessionType	• Indicates the session type. – 0 - Primary GW provisioning – 1 - Primary 1X provisioning – 2 - Secondary GW provisioning – 3 - Secondary 1X provisioning – 4 - Nonprovisioning on slot 1 – 5 - Nonprovisioning on slot 2 – 6 - Card on slot 1 – 7 - Card on slot 2 – 8 - Logical channel on slot 1 – 9 - Logical channel on slot 2
aidLength	• Number of sets of the following elements – 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: – file_id – path_len – path
arrfileInfo	• Array of file Information struct

8.993.2 Field Documentation

8.993.2.1 BYTE UIMRefreshEvent::aid[255]

8.993.2.2 BYTE UIMRefreshEvent::aidLength

8.993.2.3 fileInfo UIMRefreshEvent::arrfileInfo[255]

8.993.2.4 BYTE UIMRefreshEvent::mode

8.993.2.5 WORD UIMRefreshEvent::numOfFiles

8.993.2.6 BYTE UIMRefreshEvent::sessionType

8.993.2.7 BYTE UIMRefreshEvent::stage

8.994 UIMRefreshGetLastEventReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo

8.994.1 Detailed Description

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

Parameters

sessionInfo(-Mandatory)	• See UIMSessionInformation for more information.

8.994.2 Field Documentation

8.994.2.1 UIMSessionInformation UIMRefreshGetLastEventReq::sessionInfo

8.995 UIMRefreshGetLastEventResp Struct Reference

Data Fields

- UIMRefreshEvent * pRefreshEvent

8.995.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMRefreshGetLastEvent.

Parameters

refreshEvent(-Optional)	• See UIMRefreshEvent for more information.

8.995.2 Field Documentation

8.995.2.1 UIMRefreshEvent* UIMRefreshGetLastEventResp::pRefreshEvent

8.996 UIMRefreshOKReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- BYTE OKtoRefresh

8.996.1 Detailed Description

This structure contains Parameters of the Session Information

Parameters

sessionInfo	• Session Information • See UIMSessionInformation for more information

OKtoRefresh	• Indicates whether a refresh is OK. Valid values: – 0 - Not OK to refresh – 1 - OK to refresh

8.996.2 Field Documentation

8.996.2.1 BYTE UIMRefreshOKReq::OKtoRefresh

8.996.2.2 UIMSessionInformation UIMRefreshOKReq::sessionInfo

8.997 UIMRefreshRegisterReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- registerRefresh regRefresh

8.997.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Register.

Parameters

sessionInfo	• Session Information params • See UIMSessionInformation for more information
regRefresh	• Register Refresh parameters • See registerRefresh for more information

8.997.2 Field Documentation

8.997.2.1 registerRefresh UIMRefreshRegisterReq::regRefresh

8.997.2.2 UIMSessionInformation UIMRefreshRegisterReq::sessionInfo

8.998 UIMSessionInformation Struct Reference

Data Fields

- BYTE sessionType
- BYTE aidLength
- BYTE aid [255]

8.998.1 Detailed Description

This structure contains the Session Information.

Parameters

sessionType	• Indicates the session type. – 0 - Primary GW provisioning – 1 - Primary 1X provisioning – 2 - Secondary GW provisioning – 3 - Secondary 1X provisioning – 4 - Non-provisioning on slot 1 – 5 - Non-provisioning on slot 2 – 6 - Card on slot 1 – 7 - Card on slot 2 – 8 - Logical channel on slot 1 – 9 - Logical channel on slot 2
aidLength	• Length of the following elements i.e. 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.

8.998.2 Field Documentation

8.998.2.1 BYTE UIMSessionInformation::aid[255]

8.998.2.2 BYTE UIMSessionInformation::aidLength

8.998.2.3 BYTE UIMSessionInformation::sessionType

8.999 UIMSetPinProtectionReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- setPINProtection pinProtection
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.999.1 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. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
pIndication-Token(optional)	• Response in Indication. • 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.

8.999.2 Field Documentation

8.999.2.1 ULONG* UIMSetPinProtectionReq::pIndicationToken

8.999.2.2 setPINProtection UIMSetPinProtectionReq::pinProtection

8.999.2.3 BYTE* UIMSetPinProtectionReq::pKeyReferenceID

8.999.2.4 UIMSessionInformation UIMSetPinProtectionReq::sessionInfo

8.1000 UIMSlotsStatus Struct Reference

Data Fields

- UIMSlotStatus uimSlotStatus [255]

8.1000.1 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.

8.1000.2 Field Documentation

8.1000.2.1 UIMSlotStatus UIMSlotsStatus::uimSlotStatus[255]

8.1001 UIMSlotStatus Struct Reference

Data Fields

- ULONG uPhyCardStatus
- ULONG uPhySlotStatus
- BYTE bLogicalSlot
- BYTE bICCIDLength
- BYTE bICCID [255]

8.1001.1 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. – 0x00 - Unknown. – 0x01 - Absent. – 0x02 - Present.
uPhySlotStatus	• State of the Physical Slot status. – 0x00 Inactive. – 0x01 Activate.
bLogicalSlot	• Logical Slot associated with this physical slot. THis is valid if the physical slot is active. – 1 - Slot 1. – 2 - Slot 2. – 3 - Slot 3. – 4 - Slot 4. – 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.

8.1001.2 Field Documentation

8.1001.2.1 BYTE UIMSlotStatus::bICCID[255]

8.1001.2.2 BYTE UIMSlotStatus::bICCIDLength

8.1001.2.3 BYTE UIMSlotStatus::bLogicalSlot

8.1001.2.4 ULONG UIMSlotStatus::uPhyCardStatus

8.1001.2.5 ULONG UIMSlotStatus::uPhySlotStatus

8.1002 UIMSlotStatusChangeInfo Struct Reference

Data Fields

- UIMSlotsStatus slotsstatusChange
- BYTE bNumberOfPhySlots

8.1002.1 Detailed Description

Structure consist of cardstatus params

Parameters

slotstatus-Change	• See UIMSlotStatus for more information
bNumberOfPhy-Slots	• Number of Physical Slot(s)

8.1002.2 Field Documentation

8.1002.2.1 BYTE UIMSlotStatusChangeInfo::bNumberOfPhySlots

8.1002.2.2 UIMSlotsStatus UIMSlotStatusChangeInfo::slotsstatusChange

8.1003 UIMStatusChangeInfo Struct Reference

Data Fields

- cardStatus statusChange

8.1003.1 Detailed Description

Structure consist of cardstatus params

Parameters

statusChange	• See cardStatus for more information

8.1003.2 Field Documentation

8.1003.2.1 cardStatus UIMStatusChangeInfo::statusChange

8.1004 UIMSwitchSlotReq Struct Reference

Data Fields

- BYTE bLogicalSlot
- ULONG ulPhysicalSlot

8.1004.1 Detailed Description

This structure contains information of the request parameters associated with a Switch Slot.

Parameters

bLogicalSlot	• Indicates the slot to be used. – 1 - Slot 1 – 2 - Slot 2 – 3 - Slot 3 – 4 - Slot 4 – 5 - Slot 5
bPhysicalSlot	• 1 - Slot 1 • 2 - Slot 2 • 3 - Slot 3 • 4 - Slot 4 • 5 - Slot 5

8.1004.2 Field Documentation

8.1004.2.1 BYTE UIMSwitchSlotReq::bLogicalSlot

8.1004.2.2 ULONG UIMSwitchSlotReq::ulPhysicalSlot

8.1005 UIMUnblockPinReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- unblockUIMPIN unblockPIN
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.1005.1 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. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
pIndicationToken(optional)	• Response in Indication. • 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.

8.1005.2 Field Documentation

8.1005.2.1 ULONG* UIMUnblockPinReq::pIndicationToken

8.1005.2.2 BYTE* UIMUnblockPinReq::pKeyReferenceID

8.1005.2.3 UIMSessionInformation UIMUnblockPinReq::sessionInfo

8.1005.2.4 unblockUIMPIN UIMUnblockPinReq::unblockPIN

8.1006 UIMVerifyPinReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- verifyUIMPIN verifyPIN
- encryptedPIN1 * pEncryptedPIN1
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.1006.1 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. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
pIndication-Token(optional)	• Response in Indication. • 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.

8.1006.2 Field Documentation

8.1006.2.1 encryptedPIN1∗ UIMVerifyPinReq::pEncryptedPIN1

8.1006.2.2 ULONG∗ UIMVerifyPinReq::pIndicationToken

8.1006.2.3 BYTE∗ UIMVerifyPinReq::pKeyReferenceID

8.1006.2.4 UIMSessionInformation UIMVerifyPinReq::sessionInfo

8.1006.2.5 verifyUIMPIN UIMVerifyPinReq::verifyPIN

8.1007 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]

8.1007.1 Detailed Description

This structure contains information about the UMTS Network.

Parameters

cellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.

plmn[PLMN_LENGTH]	• MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when nmrCellID is not present.
lac	• Location area code. • This field is ignored when nmrCellID is not present. – 0xFFFF - Not Available
uarfcn	• UTRA absolute RF channel number. – 0xFFFF - Not Available
psc	• Primary scrambling code. – 0xFFFF - Not Available
rscp	• Received signal code power. – 0xFFFF - Not Available
ecio	• ECIO(Signal-to-Interference-ratio). – 0xFFFF - Not Available
umtsInst	• Provides the number of set of UMTS info instances. • 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. • If 0(zero), then no information follows it.
GeranInstInfo[MAX_DESCRIPTION_LENGTH]	• See geranInstInfo for more information.

8.1007.2 Field Documentation

8.1007.2.1 WORD UMTSInfo::cellID

8.1007.2.2 SHORT UMTSInfo::ecio

8.1007.2.3 BYTE UMTSInfo::geranInst

8.1007.2.4 geranInstInfo UMTSInfo::GeranInstInfo[255]

8.1007.2.5 WORD UMTSInfo::lac

8.1007.2.6 BYTE UMTSInfo::plmn[3]

8.1007.2.7 WORD UMTSInfo::psc

8.1007.2.8 SHORT UMTSInfo::rscp

8.1007.2.9 WORD UMTSInfo::uarfcn

8.1007.2.10 BYTE UMTSInfo::umtsInst

8.1007.2.11 UMTSinstInfo UMTSInfo::UMTSInstInfo[255]

8.1008 UMTSinstInfo Struct Reference

Data Fields

- WORD umtsUarfcn
- WORD umtsPsc
- SHORT umtsRscp
- SHORT umtsEcio

8.1008.1 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).

8.1008.2 Field Documentation

8.1008.2.1 SHORT UMTSinstInfo::umtsEcio

8.1008.2.2 WORD UMTSinstInfo::umtsPsc

8.1008.2.3 SHORT UMTSinstInfo::umtsRscp

8.1008.2.4 WORD UMTSinstInfo::umtsUarfcn

8.1009 umtsLTENbrCell Struct Reference

Data Fields

- WORD earfcn
- WORD pci
- ULONG rsrp
- ULONG rsrq
- SHORT srxlev
- BYTE cellIsTDD

8.1009.1 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. • 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. • 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.

8.1009.2 Field Documentation

8.1009.2.1 BYTE umtsLTENbrCell::cellIsTDD

8.1009.2.2 WORD umtsLTENbrCell::earfcn

8.1009.2.3 WORD umtsLTENbrCell::pci

8.1009.2.4 ULONG umtsLTENbrCell::rsrp

8.1009.2.5 ULONG umtsLTENbrCell::rsrq

8.1009.2.6 SHORT umtsLTENbrCell::srxlev

8.1010 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

8.1010.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

Parameters

trafficClass	• 0x00 - Subscribed • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 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 • 0x00 - Subscribe • 0x01 - Delivery order on • 0x02 - Delivery order off
maxSDUSize	• Maximum SDU size

sduErrorRatio	- SDU error ratio • Target value for fraction of SDUs lost or detected as erroneous. • 0x00 - Subscribe • 0x01 - 1∗10^(-2) • 0x02 - 7∗10^(-3) • 0x03 - 1∗10^(-3) • 0x04 - 1∗10^(-4) • 0x05 - 1∗10^(-5) • 0x06 - 1∗10^(-6) • 0x07 - 1∗10^(-1)
resBerRatio	- Residual bit error ratio • Target value for undetected bit error ratio in in the delivered SDUs. • 0x00 - Subscribe • 0x01 - 5∗10^(-2) • 0x02 - 1∗10^(-2) • 0x03 - 5∗10^(-3) • 0x04 - 4∗10^(-3) • 0x05 - 1∗10^(-3) • 0x06 - 1∗10^(-4) • 0x07 - 1∗10^(-5) • 0x08 - 1∗10^(-6) • 0x09 - 1∗10^(-8)
deliveryErrSDU	- delivery of erroneous SDUs • Indicates whether SDUs detected as erroneous shall be delivered or not. • 0x00 - Subscribe • 0x01 - 5∗10^(-2) • 0x02 - 1∗10^(-2) • 0x03 - 5∗10^(-3) • 0x04 - 4∗10^(-3) • 0x05 - 1∗10^(-3) • 0x06 - 1∗10^(-4) • 0x07 - 1∗10^(-5) • 0x08 - 1∗10^(-6) • 0x09 - 1∗10^(-8)
transferDelay	- Transfer delay (ms) • Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds.
trafficPriority	- Transfer handling priority • Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.

8.1010.2 Field Documentation

8.1010.2.1 BYTE UMTSMinQoS::deliveryErrSDU

8.1010.2.2 ULONG UMTSMinQoS::grntDownlinkBitrate

8.1010.2.3 ULONG UMTSMinQoS::grntUplinkBitrate

8.1010.2.4 ULONG UMTSMinQoS::maxDownlinkBitrate

8.1010.2.5 ULONG UMTSMinQoS::maxSDUSize

8.1010.2.6 ULONG UMTSMinQoS::maxUplinkBitrate

8.1010.2.7 BYTE UMTSMinQoS::qosDeliveryOrder

8.1010.2.8 BYTE UMTSMinQoS::resBerRatio

8.1010.2.9 BYTE UMTSMinQoS::sduErrorRatio

8.1010.2.10 BYTE UMTSMinQoS::trafficClass

8.1010.2.11 ULONG UMTSMinQoS::trafficPriority

8.1010.2.12 ULONG UMTSMinQoS::transferDelay

8.1011 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

8.1011.1 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 • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 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 • 0x00 - Subscribe • 0x01 - delivery order on • 0x02 - delivery order off
maxSDUSize	• Maximum SDU size
sduErrorRatio	- SDU error ratio • Target value for fraction of SDUs lost or detected as erroneous. • 0x00 - Subscribe • 0x01 - 1∗10^(-2) • 0x02 - 7∗10^(-3) • 0x03 - 1∗10^(-3) • 0x04 - 1∗10^(-4) • 0x05 - 1∗10^(-5) • 0x06 - 1∗10^(-6) • 0x07 - 1∗10^(-1)
resBerRatio	- Residual bit error ratio • Target value for undetected bit error ratio in in the delivered SDUs. • 0x00 - Subscribe • 0x01 - 5∗10^(-2) • 0x02 - 1∗10^(-2) • 0x03 - 5∗10^(-3) • 0x04 - 4∗10^(-3) • 0x05 - 1∗10^(-3) • 0x06 - 1∗10^(-4) • 0x07 - 1∗10^(-5) • 0x08 - 1∗10^(-6) • 0x09 - 1∗10^(-8)

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

8.1011.2 Field Documentation

8.1011.2.1 BYTE UMTSQoS::deliveryErrSDU

8.1011.2.2 ULONG UMTSQoS::grntDownlinkBitrate

8.1011.2.3 ULONG UMTSQoS::grntUplinkBitrate

8.1011.2.4 ULONG UMTSQoS::maxDownlinkBitrate

8.1011.2.5 ULONG UMTSQoS::maxSDUSize

8.1011.2.6 ULONG UMTSQoS::maxUplinkBitrate

8.1011.2.7 BYTE UMTSQoS::qosDeliveryOrder

8.1011.2.8 BYTE UMTSQoS::resBerRatio

8.1011.2.9 BYTE UMTSQoS::sduErrorRatio

8.1011.2.10 BYTE UMTSQoS::trafficClass

8.1011.2.11 ULONG UMTSQoS::trafficPriority

8.1011.2.12 ULONG UMTSQoS::transferDelay

8.1012 UMTSReqQoSSigInd Struct Reference

Data Fields

- struct UMTSQoS UMTSReqQoS
- BYTE SigInd

8.1012.1 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 • TRUE - Signaling indication ON • FALSE - Signaling indication OFF

8.1012.2 Field Documentation

8.1012.2.1 BYTE UMTSReqQoSSigInd::SigInd

8.1012.2.2 struct UMTSQoS UMTSReqQoSSigInd::UMTSReqQoS

8.1013 unblockUIMPIN Struct Reference

Data Fields

- BYTE pinID
- BYTE pukLen
- BYTE pukVal [255]
- BYTE newPINLen
- BYTE newPINVal [255]

8.1013.1 Detailed Description

This structure contains the information about the unblock pin parameters.

Parameters

pinID	• Indicates the PIN ID to be changed. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN
pukLen	• Length of the following elements i.e. puk value.

pukVal[MAX_PUK_LENGTH]	• PIN Unlock Key value. • 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. • This value is a sequence of ASCII characters.

8.1013.2 Field Documentation

8.1013.2.1 BYTE unblockUIMPIN::newPINLen

8.1013.2.2 BYTE unblockUIMPIN::newPINVal[255]

8.1013.2.3 BYTE unblockUIMPIN::pinID

8.1013.2.4 BYTE unblockUIMPIN::pukLen

8.1013.2.5 BYTE unblockUIMPIN::pukVal[255]

8.1014 UniversalTime Struct Reference

Data Fields

- WORD year
- BYTE month
- BYTE day
- BYTE hour
- BYTE minute
- BYTE second
- BYTE dayOfWeek

8.1014.1 Detailed Description

This structure contains the parameters for Universal Time Information.

Parameters

year	• Year.
month	• Month. – 1 is January and 12 is December.
day	• Day. – Range 1 to 31.

hour	• Hour. – Range 0 to 59.
minute	• Minute. – Range 0 to 59.
second	• Second. – Range 0 to 59.
dayOfWeek	• Day of the Week. – 0 is Monday and 6 is Sunday.

8.1014.2 Field Documentation

8.1014.2.1 BYTE UniversalTime::day

8.1014.2.2 BYTE UniversalTime::dayOfWeek

8.1014.2.3 BYTE UniversalTime::hour

8.1014.2.4 BYTE UniversalTime::minute

8.1014.2.5 BYTE UniversalTime::month

8.1014.2.6 BYTE UniversalTime::second

8.1014.2.7 WORD UniversalTime::year

8.1015 unpack_dms_GetActivationState_t Struct Reference

Data Fields

- uint8_t state

8.1015.1 Detailed Description

Parameters

pActivation-State[OUT]	• Service Activation Code 0 - Service not activated 1 - Service activated 2 - Activation connecting 3 - Activation connected 4 - OTASP security authenticated 5 - OTASP NAM downloaded 6 - OTASP MDN downloaded 7 - OTASP IMSI downloaded 8 - OTASP PRL downloaded 9 - OTASP SPC downloaded 10 - OTASP settings committed

8.1015.2 Field Documentation

8.1015.2.1 uint8_t unpack_dms_GetActivationState_t::state

8.1016 unpack_dms_GetBandCapability_t Struct Reference

Data Fields

- uint32_t BandCapability
- uint16_t Tlvresult

8.1016.1 Field Documentation

8.1016.1.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

8.1016.1.2 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

8.1017 unpack_dms_GetCrashAction_t Struct Reference

Data Fields

- uint8_t DevCrashState
- uint16_t Tlvresult

8.1017.1 Field Documentation

8.1017.1.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

8.1017.1.2 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

8.1018 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

8.1018.1 Field Documentation

8.1018.1.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

8.1018.1.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

8.1018.1.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

8.1018.1.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

8.1018.1.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

8.1018.1.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

8.1018.1.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

8.1018.1.8 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

8.1018.1.9 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

8.1018.1.10 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

8.1019 unpack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

- DMSgetCustomFeatureV2 GetCustomFeatureV2
- uint16_t Tlvresult

8.1019.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

Tlvresult	• Unpack Result

8.1019.2 Field Documentation

8.1019.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

8.1019.2.2 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

8.1020 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

8.1020.1 Field Documentation

8.1020.1.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

8.1020.1.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

8.1020.1.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

8.1020.1.4 uint8_t unpack_dms_GetDeviceCap_t::RadioIfaces[64]

8.1020.1.5 uint32_t unpack_dms_GetDeviceCap_t::RadioIfacesSize

8.1020.1.6 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

8.1020.1.7 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

8.1021 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]

8.1021.1 Detailed Description

Parameters

maxTxChannel-Rate	Maximum Tx transmission rate in bits per second.
maxRxChannel-Rate	Maximum Rx transmission rate in bits per second
dataServiceCa-Capability	data service capability
simCapability	SIM Capability
radioIfacesSize	radio interface length
RadioIfaces	radio interfaces

8.1021.2 Field Documentation

8.1021.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability

8.1021.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate

8.1021.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate

8.1021.2.4 uint8_t unpack_dms_GetDeviceCapabilities_t::RadioIfaces[255]

8.1021.2.5 uint32_t unpack_dms_GetDeviceCapabilities_t::radioIfacesSize

8.1021.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

8.1022 unpack_dms_GetDeviceHardwareRev_t Struct Reference

Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult

8.1022.1 Field Documentation

8.1022.1.1 char unpack_dms_GetDeviceHardwareRev_t::String[255]

8.1022.1.2 uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize

8.1022.1.3 uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult

8.1023 unpack_dms_GetDeviceMfr_t Struct Reference

Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult

8.1023.1 Field Documentation

8.1023.1.1 char unpack_dms_GetDeviceMfr_t::String[255]

8.1023.1.2 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

8.1023.1.3 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

8.1024 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

8.1024.1 Field Documentation

8.1024.1.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

8.1024.1.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

8.1024.1.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

8.1024.1.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

8.1024.1.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

8.1024.1.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

8.1024.1.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

8.1024.1.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

8.1024.1.9 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

8.1025 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

8.1025.1 Detailed Description

Parameters

modelid_str	Mode ID String.
bootversion_str	Boot Version.
appversion_str	Application Version String.
sku_str	SKU String.
packageid_str	• Package ID String. • deprecated on EM/MC74xx(9x30) devices

carrier_str	Carrier String.
priversion_str	PRI Version String.
priversion_str	PRI Version String.
cur_carr_name	Current Carrier Name String.
cur_carr_rev	Current Carrier Revision String.
Tlvresult	Tlv Result.

8.1025.2 Field Documentation

8.1025.2.1 char unpack_dms_GetFirmwareInfo_t::appversion_str[85]

8.1025.2.2 char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]

8.1025.2.3 char unpack_dms_GetFirmwareInfo_t::carrier_str[20]

8.1025.2.4 char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]

8.1025.2.5 char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]

8.1025.2.6 char unpack_dms_GetFirmwareInfo_t::modelid_str[20]

8.1025.2.7 char unpack_dms_GetFirmwareInfo_t::packageid_str[85]

8.1025.2.8 char unpack_dms_GetFirmwareInfo_t::priversion_str[16]

8.1025.2.9 char unpack_dms_GetFirmwareInfo_t::sku_str[15]

8.1025.2.10 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

8.1026 unpack_dms_GetFirmwareRevision_t Struct Reference

Data Fields

- uint8_t amssSize
- char AMSSString [255]
- char PRIString [255]
- uint16_t Tlvresult

8.1026.1 Field Documentation

8.1026.1.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

8.1026.1.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

8.1026.1.3 char unpack_dms_GetFirmwareRevision_t::PRIString[255]

8.1026.1.4 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

8.1027 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

8.1027.1 Detailed Description

Parameters

amssstring	AMSS revision string
bootstring	boot code revision string
pristring	PRI revision string

8.1027.2 Field Documentation

8.1027.2.1 uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize

8.1027.2.2 char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]

8.1027.2.3 uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize

8.1027.2.4 char unpack_dms_GetFirmwareRevisions_t::BootString[255]

8.1027.2.5 uint8_t unpack_dms_GetFirmwareRevisions_t::priSize

8.1027.2.6 char unpack_dms_GetFirmwareRevisions_t::PRIString[255]

8.1027.2.7 uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult

8.1028 unpack_dms_GetFSN_t Struct Reference

Data Fields

- char String [255]
- uint16_t Tlvresult

8.1028.1 Field Documentation

8.1028.1.1 char unpack_dms_GetFSN_t::String[255]

8.1028.1.2 uint16_t unpack_dms_GetFSN_t::Tlvresult

8.1029 unpack_dms_GetHardwareRevision_t Struct Reference

Data Fields

- char hwVer [255]

8.1029.1 Detailed Description

Parameters

hwVer	hardware vesion

8.1029.2 Field Documentation

8.1029.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

8.1030 unpack_dms_GetIMSI_t Struct Reference

Data Fields

- char imsi [255]
- uint16_t Tlvresult

8.1030.1 Field Documentation

8.1030.1.1 char unpack_dms_GetIMSI_t::imsi[255]

8.1030.1.2 uint16_t unpack_dms_GetIMSI_t::Tlvresult

8.1031 unpack_dms_GetManufacturer_t Struct Reference

Data Fields

- char manufacturer [255]
- uint16_t Tlvresult

8.1031.1 Detailed Description

This structure is used to store device manufacturer information.

Parameters

manufacturer[OUT]	• NULL terminated string
Tlvresult	• Unpack Result

8.1031.2 Field Documentation

8.1031.2.1 char unpack_dms_GetManufacturer_t::manufacturer[255]

8.1031.2.2 uint16_t unpack_dms_GetManufacturer_t::Tlvresult

8.1032 unpack_dms_GetModelID_t Struct Reference

Data Fields

- char modelid [255]

- uint16_t Tlvresult

8.1032.1 Detailed Description

Parameters

modelid	device model id

8.1032.2 Field Documentation

8.1032.2.1 char unpack_dms_GetModelID_t::modelid[255]

8.1032.2.2 uint16_t unpack_dms_GetModelID_t::Tlvresult

8.1033 unpack_dms_GetNetworkTime_t Struct Reference

Data Fields

- uint16_t source
- uint64_t timestamp
- uint16_t Tlvresult

8.1033.1 Detailed Description

Parameters

source	• Source of timestamp 0 - 32 kHz device clock 1 - CDMA network 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

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.

8.1033.2 Field Documentation

8.1033.2.1 uint16_t unpack_dms_GetNetworkTime_t::source

8.1033.2.2 uint64_t unpack_dms_GetNetworkTime_t::timestamp

8.1033.2.3 uint16_t unpack_dms_GetNetworkTime_t::Tlvresult

8.1034 unpack_dms_GetOfflineReason_t Struct Reference

Data Fields

- uint32_t * pReasonMask
- uint32_t * pbPlatform
- uint16_t Tlvresult

8.1034.1 Detailed Description

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

Parameters

pReasonMask[-OUT]	• Optional parameter • Bitmask of offline reasons – 0x00000001 - Host image configuration issue – 0x00000002 - PRI image configuration issue – 0x00000004 - PRI version incompatible – 0x00000008 - PRI copy issue – All others - Reserved
pbPlatform[OUT]	• Optional parameter • Is the device offline due to a platform restriction? – 0 - No – 1 - Yes
Tlvresult	• Unpack Result

8.1034.2 Field Documentation

8.1034.2.1 uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform

8.1034.2.2 uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask

8.1034.2.3 uint16_t unpack_dms_GetOfflineReason_t::Tlvresult

8.1035 unpack_dms_GetPower_t Struct Reference

Data Fields

- uint32_t OperationMode
- uint32_t OfflineReason
- uint32_t HardwareControlledMode
- uint16_t Tlvresult

8.1035.1 Detailed Description

Parameters

OperationMode	operating mode
OfflineReason	offline reason
Hardware-ControlledMode	hardware restricted mode

8.1035.2 Field Documentation

8.1035.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

8.1035.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

8.1035.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

8.1035.2.4 uint16_t unpack_dms_GetPower_t::Tlvresult

8.1036 unpack_dms_GetPRLVersion_t Struct Reference

Data Fields

- uint8_t u8PRLPreference
- uint16_t u16PRLVersion
- uint16_t Tlvresult

8.1036.1 Field Documentation

8.1036.1.1 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

8.1036.1.2 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

8.1036.1.3 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

8.1037 unpack_dms_GetSerialNumbers_t Struct Reference

Data Fields

- char esn [255]
- char imei_no [255]
- char meid [255]
- char imeisv_svn [255]

8.1037.1 Detailed Description

Parameters

esn	Electronic Serial Number of the device
imei_no	International Mobile Equipment Identity of the device.
meid	Mobile Equipment Identifier of the device.
imeisv_svn	imei software version revision

8.1037.2 Field Documentation

8.1037.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

8.1037.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

8.1037.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

8.1037.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

8.1038 unpack_dms_GetUSBComp_t Struct Reference

Data Fields

- uint8_t USBComp [255]
- uint8_t NumSupUSBComps
- uint8_t SupUSBComps
- uint16_t Tlvresult

8.1038.1 Field Documentation

8.1038.1.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

8.1038.1.2 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

8.1038.1.3 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

8.1038.1.4 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

8.1039 unpack_dms_GetVoiceNumber_t Struct Reference

Data Fields

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

8.1039.1 Field Documentation

8.1039.1.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

8.1039.1.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

8.1039.1.3 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

8.1039.1.4 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

8.1039.1.5 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

8.1040 unpack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1040.1 Detailed Description

This structure contains reset to factory default unpack

Parameters

Tlvresult	• Unpack Result

8.1040.2 Field Documentation

8.1040.2.1 uint16_t unpack_dms_ResetToFactoryDefaults_t::Tlvresult

8.1041 unpack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1041.1 Detailed Description

unpack structure for Activation status callback

Parameters

Tlvresult	• Unpack Result

8.1041.2 Field Documentation

8.1041.2.1 uint16_t unpack_dms_SetActivationStatusCallback_t::Tlvresult

8.1042 unpack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t notused

8.1042.1 Detailed Description

Modem response. Not used

8.1042.2 Field Documentation

8.1042.2.1 uint8_t unpack_dms_SetCrashAction_t::notused

8.1043 unpack_dms_SetCustFeature_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1043.1 Field Documentation

8.1043.1.1 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

8.1044 unpack_dms_SetCustFeaturesV2_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1044.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

Tlvresult	• Unpack Result

8.1044.2 Field Documentation

8.1044.2.1 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

8.1045 unpack_dms_SetEventReport_ind_t Struct Reference

Data Fields

- dms_ActivationStatusTlv ActivationStatusTlv
- dms_OperatingModeTlv OperatingModeTlv
- uint16_t Tlvresult

8.1045.1 Detailed Description

DMS Event Report indication structure

Parameters

ActivationStatus-Tlv	• See dms_ActivationStatusTlv
OperatingMode-Tlv	• See dms_OperatingModeTlv

Tlvresult	• Unpack Result

8.1045.2 Field Documentation

8.1045.2.1 dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv

8.1045.2.2 dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv

8.1045.2.3 uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult

8.1046 unpack_dms_SetEventReport_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1046.1 Field Documentation

8.1046.1.1 uint16_t unpack_dms_SetEventReport_t::Tlvresult

8.1047 unpack_dms_SetFirmwarePreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1047.1 Field Documentation

8.1047.1.1 uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult

8.1048 unpack_dms_SetPower_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1048.1 Field Documentation

8.1048.1.1 uint16_t unpack_dms_SetPower_t::Tlvresult

8.1049 unpack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1049.1 Field Documentation

8.1049.1.1 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

8.1050 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult

8.1050.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

OUT]	type[OUT] • type of reset or power down, possible values listed below: – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down
OUT]	source[OUT] • entity which initiated the reset or power down, possible values listed below: – 0 - unknown – 1 - user requested (AT!RESET, AT!BOOTHOLD, FW/PRI download – including host-initiated image switching) – 2 - hardware switch (W_DISABLE) – 3 - temperature critical – 4 - voltage critical – 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset) – 6 - LWM2M (Light Weight M2M client (internal process for LWM2M)) – 7 - OMA-DM – 8 - FOTA
Tlvresult	• Unpack Result

8.1050.2 Field Documentation

8.1050.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

8.1050.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

8.1050.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

8.1051 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult

8.1051.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

OUT]	type[OUT] • type of reset or power down, possible values listed below: – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down
OUT]	source[OUT] • entity which initiated the reset or power down, possible values listed below: – 0 - unknown – 1 - user requested (AT!RESET, AT!BOOTHOLD, FW/PRI download – including host-initiated image switching) – 2 - hardware switch (W_DISABLE) – 3 - temperature critical – 4 - voltage critical – 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset) – 6 - LWM2M (Light Weight M2M client (internal process for LWM2M)) – 7 - OMA-DM – 8 - FOTA
Tlvresult	• Unpack Result

8.1051.2 Field Documentation

8.1051.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

8.1051.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

8.1051.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

8.1052 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1052.1 Detailed Description

This structure contains set registration state for different indication unpack

Parameters

Tlvresult	• Unpack Result

8.1052.2 Field Documentation

8.1052.2.1 uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::Tlvresult

8.1053 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

8.1053.1 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 • Bit 0 - Band class 0, A-system • Bit 1 - Band class 0, B-system • Bit 2 - Band class 1, all blocks • Bit 3 - Band class 2 • Bit 4 - Band class 3, A-system • Bit 5 - Band class 4, all blocks • Bit 6 - Band class 5, all blocks • Bit 7 - GSM DCS band (1800) • Bit 8 - GSM Extended GSM (E-GSM) band (900) • Bit 9 - GSM Primary GSM (P-GSM) band (900) • Bit 10 - Band class 6 • Bit 11 - Band class 7 • Bit 12 - Band class 8 • Bit 13 - Band class 9 • Bit 14 - Band class 10 • Bit 15 - Band class 11 • Bit 16 - GSM 450 band • Bit 17 - GSM 480 band • Bit 18 - GSM 750 band • Bit 19 - GSM 850 band • Bit 20 - GSM railways GSM band (900) • Bit 21 - GSM PCS band (1900) • Bit 22 - WCDMA (Europe, Japan, and China) 2100 band • Bit 23 - WCDMA US PCS 1900 band • Bit 24 - WCDMA (Europe and China) DCS 1800 band • Bit 25 - WCDMA US 1700 band • Bit 26 - WCDMA US 850 band • Bit 27 - WCDMA Japan 800 band • Bit 28 - Band class 12 • Bit 29 - Band class 14 • Bit 30 - Reserved • Bit 31 - Band class 15 • Bits 32 through 47 - Reserved • Bit 48 - WCDMA Europe 2600 band • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved • Bit 56 - Band class 16 • Bit 57 - Band class 17 • Bit 58 - Band class 18 • Bit 59 - Band class 19

LteBand-Capability[OUT]	Bitmask of LTE bands supported by the device • Bit 0 - LTE EUTRAN Band 1 UL:1920-1980; DL: 2110-2170 • Bit 1 - LTE EUTRAN Band 2 UL:1850-1910; DL: 1930-1990 • Bit 2 - LTE EUTRAN Band 3 UL:1710-1785; DL: 1805-1880 • Bit 3 - LTE EUTRAN Band 4 UL:1710-1755; DL: 2110-2155 • Bit 4 - LTE EUTRAN Band 5 UL: 824-849; DL: 869-894 • Bit 5 - LTE EUTRAN Band 6 UL: 830-840; DL: 875-885 • Bit 6 - LTE EUTRAN Band 7 UL:2500-2570; DL: 2620-2690 • Bit 7 - LTE EUTRAN Band 8 UL: 880-915; DL: 925-960 • Bit 8 - LTE EUTRAN Band 9 UL:1749.9-1784.9; DL: 1844.9-1879.9 • Bit 9 - LTE EUTRAN Band 10 UL:1710-1770; DL: 2110-2170 • Bit 10 - LTE EUTRAN Band 11 UL:1427.9-1452.9; DL: 1475.9-1500.9 • Bit 11 - LTE EUTRAN Band 12 UL:698-716; DL: 728-746 • Bit 12 - LTE EUTRAN Band 13 UL: 777-787; DL: 746-756 • Bit 13 - LTE EUTRAN Band 14 UL: 788-798; DL: 758-768 • Bits 14 and 15 - Reserved • Bit 16 - LTE EUTRAN Band 17 UL: 704-716; DL: 734-746 • Bit 17 - LTE EUTRAN Band 18 UL: 815-830; DL: 860-875 • Bit 18 - LTE EUTRAN Band 19 UL: 830-845; DL: 875-890 • Bit 19 - LTE EUTRAN Band 20 UL: 832-862; DL: 791-821 • Bit 20 - LTE EUTRAN Band 21 UL: 1447.9-1462.9; DL: 1495.9-1510.9 • Bit 21 - Reserved • Bit 22 - LTE EUTRAN Band 23 UL: 2000-2020; DL: 2180-2200 • Bit 23 - LTE EUTRAN Band 24 UL: 1626.5-1660.5; DL: 1525-1559 • Bit 24 - LTE EUTRAN Band 25 UL: 1850-1915; DL: 1930-1995 • Bit 25 - LTE EUTRAN Band 26 UL: 814-849; DL: 859-894 • Bit 26 - Reserved • Bit 27 - LTE EUTRAN Band 28 UL: 703-748; DL: 758-803 • Bit 28 - LTE EUTRAN Band 29 UL: 1850-1910 or 1710-1755; DL: 716-728 • Bits 29 through 31 - Reserved • Bit 32 - LTE EUTRAN Band 33 UL: 1900-1920; DL: 1900-1920 • Bit 33 - LTE EUTRAN Band 34 UL: 2010-2025; DL: 2010-2025 • Bit 34 - LTE EUTRAN Band 35 UL: 1850-1910; DL: 1850-1910 • Bit 35 - LTE EUTRAN Band 36 UL: 1930-1990; DL: 1930-1990 • Bit 36 - LTE EUTRAN Band 37 UL: 1910-1930; DL: 1910-1930 • Bit 37 - LTE EUTRAN Band 38 UL: 2570-2620; DL: 2570-2620 • Bit 38 - LTE EUTRAN Band 39 UL: 1880-1920; DL: 1880-1920 • Bit 39 - LTE EUTRAN Band 40 UL: 2300-2400; DL: 2300-2400 • Bit 40 - LTE EUTRAN Band 41 UL: 2496-2690; DL: 2496-2690 • Bit 41 - LTE EUTRAN Band 42 UL: 3400-3600; DL: 3400-3600 • Bit 42 - LTE EUTRAN Band 43 UL: 3600-3800; DL: 3600-3800 • Bits 43 through 64 - Reserved

TdsBand-Capability[OUT]	Bitmask of TDS bands supported by the device. • Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz • Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz • Bit 2 - TDS Band C 1910 to 1930 MHz • Bit 3 - TDS Band D 2570 to 2620 MHz • Bit 4 - TDS Band E 2300 to 2400 MHz • Bit 5 - TDS Band F 1880 to 1920 MHz

8.1053.2 Field Documentation

8.1053.2.1 uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability

8.1053.2.2 int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available

8.1053.2.3 int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available

8.1053.2.4 uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability

8.1053.2.5 uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability

8.1054 unpack_dms_SLQSGetERIFile_t Struct Reference

Data Fields

- uint16_t Tlvresult
- eriDataparams eriFile

8.1054.1 Detailed Description

This structure contains Get ERI file unpack

Parameters

Tlvresult	• Unpack Result

8.1054.2 Field Documentation

8.1054.2.1 eriDataparams unpack_dms_SLQSGetERIFile_t::eriFile

8.1054.2.2 uint16_t unpack_dms_SLQSGetERIFile_t::Tlvresult

8.1055 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1055.1 Detailed Description

This structure contains Clear Dying GASP unpack

Parameters

TlvResult	• Unpack Result

8.1055.2 Field Documentation

8.1055.2.1 uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::TlvResult

8.1056 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference

Data Fields

- uint16_t TlvResult
- crashInfoParams crashInfoParam

8.1056.1 Detailed Description

This structure contains SWI get crash information unpack information

Parameters

TlvResult	• Unpack Result
crashInfoParam[-OUT]	• See crashInfoParams

8.1056.2 Field Documentation

8.1056.2.1 crashInfoParams unpack_dms_SLQSSwiGetCrashInfo_t::crashInfoParam

8.1056.2.2 uint16_t unpack_dms_SLQSSwiGetCrashInfo_t::TlvResult

8.1057 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

Data Fields

- packgetDyingGaspCfg * pGetDyingGaspCfg
- uint16_t TlvResult

8.1057.1 Detailed Description

This structure contains Get Dying GASP Config unpack

Parameters

Tlvresult	• Unpack Result

8.1057.2 Field Documentation

8.1057.2.1 packgetDyingGaspCfg∗ unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg

8.1057.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult

8.1058 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference

Data Fields

- packgetDyingGaspStatistics ∗ pGetDyingGaspStatistics
- uint16_t Tlvresult

8.1058.1 Detailed Description

This structure contains Get Dying GASP Statistics.

Parameters

Tlvresult	• Unpack Result

8.1058.2 Field Documentation

8.1058.2.1 packgetDyingGaspStatistics∗ unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics

8.1058.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult

8.1059 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]

8.1059.1 Detailed Description

Parameters

numEntries[IN/-OUT]	• Number of entries in the image list to follow • 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

8.1059.2 Field Documentation

8.1059.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

8.1059.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

8.1059.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

8.1059.2.4 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

8.1059.2.5 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

8.1059.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

8.1060 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

8.1060.1 Detailed Description

This structure is used to store Firmware Update Status

Parameters

ResCode	• FW Update Result Code • Values: – 0x00000001 - Successful – 0xFFFFFFFF - Unknown (due to power off reset after firmware update) – 0x100000nn - File update errors while nn will be the exact error number: * 00 - General error – 0x200000nn - NVUP update errors while nn will be the exact error number: * 00 - General error – 0x40000nnn - FOTA update agent errors while nnn will be the exact error number: * 000 $\sim$ 0FF - Insignia defined error code * 100 $\sim$ 1FF - Sierra defined error code * See qaGobiApiTableFwDldErrorCodes.h for more detailed information – 0x800000nn - FDT/SSDP reported errors while nn will be the exact error number * See qaGobiApiTableFwDldErrorCodes.h for more detailed information
imgType	• Optional parameter • Firmware image type that failed the update
refData	• Optional parameter • Failed image reference data • This is normally the offset of the image that caused the failure
refString	• Optional parameter • Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.
logString	• Optional parameter • Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.
Tlvresult	• Unpack Result

8.1060.2 Field Documentation

8.1060.2.1 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType

8.1060.2.2 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]

8.1060.2.3 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData

8.1060.2.4 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]

8.1060.2.5 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode

8.1060.2.6 uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::TlvResult

8.1061 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference

Data Fields

- uint16_t TlvResult
- char manString [255]
- char modelString [255]
- char swVerString [255]
- char plasmaIDString [255]

8.1061.1 Detailed Description

This structure contains SWI get host device info unpack information

Parameters

TlvResult	• unpack Result
manString[OUT]	• optional parameter, host device manufacture
modelString[OUT]	• optional parameter, host device model
swVerString[OUT]	• optional parameter, host device software version
plasmaIDString[OUT]	• optional parameter, host device plasma ID

8.1061.2 Field Documentation

8.1061.2.1 char unpack_dms_SLQSSwiGetHostDevInfo_t::manString[255]

8.1061.2.2 char unpack_dms_SLQSSwiGetHostDevInfo_t::modelString[255]

8.1061.2.3 char unpack_dms_SLQSSwiGetHostDevInfo_t::plasmaIDString[255]

8.1061.2.4 char unpack_dms_SLQSSwiGetHostDevInfo_t::swVerString[255]

8.1061.2.5 uint16_t unpack_dms_SLQSSwiGetHostDevInfo_t::TlvResult

8.1062 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference

Data Fields

- uint16_t TlvResult
- char nameString [255]
- char versionString [255]

8.1062.1 Detailed Description

This structure contains SWI get host os info unpack information

Parameters

TlvResult	• pack Result
nameString[OUT]	• optional parameter, host operating system name
VersionString[OUT]	• optional parameter, host operating system version

8.1062.2 Field Documentation

8.1062.2.1 char unpack_dms_SLQSSwiGetOSInfo_t::nameString[255]

8.1062.2.2 uint16_t unpack_dms_SLQSSwiGetOSInfo_t::TlvResult

8.1062.2.3 char unpack_dms_SLQSSwiGetOSInfo_t::versionString[255]

8.1063 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference

Data Fields

- uint16_t TlvResult
- char meidString [8]

8.1063.1 Detailed Description

This structure contains SWI get device serial number extension unpack information

Parameters

TlvResult	• pack Result
meidString[OUT]	• optional parameter, mobile equipment identifier

8.1063.2 Field Documentation

8.1063.2.1 char unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]

8.1063.2.2 uint16_t unpack_dms_SLQSSwiGetSerialNoExt_t::TlvResult

8.1064 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1064.1 Detailed Description

This structure contains set Dying GASP Config unpack

Parameters

Tlvresult	• Unpack Result

8.1064.2 Field Documentation

8.1064.2.1 uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

8.1065 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1065.1 Detailed Description

This structure contains SWI set host dev info unpack

Parameters

Tlvresult	• Unpack Result

8.1065.2 Field Documentation

8.1065.2.1 uint16_t unpack_dms_SLQSSwiSetHostDevInfo_t::Tlvresult

8.1066 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1066.1 Detailed Description

This structure contains SWI set host OS info unpack

Parameters

Tlvresult	• Unpack Result

8.1066.2 Field Documentation

8.1066.2.1 uint16_t unpack_dms_SLQSSwiSetOSInfo_t::TlvResult

8.1067 unpack_dms_SLQSUIMGetState_t Struct Reference

Data Fields

- uint8_t state
- uint16_t TlvResult

8.1067.1 Detailed Description

This structure contains UIM get state unpack information

Parameters

state[OUT]	• UIM state
TlvResult	• Unpack Result

8.1067.2 Field Documentation

8.1067.2.1 uint8_t unpack_dms_SLQSUIMGetState_t::state

8.1067.2.2 uint16_t unpack_dms_SLQSUIMGetState_t::TlvResult

8.1068 unpack_dms_UIMGetControlKeyStatus_t Struct Reference

Data Fields

- uint8_t facilityState
- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t TlvResult

8.1068.1 Detailed Description

This structure contains PIN retries status

Parameters

facilityState[OUT]	• Control key status – 0 - Deactivated – 1 - Activated – 2 - Blocked

verifyRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
unblockRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. – 0xFFFFFFFF - Unknown
TlvResult	• Unpack Result

8.1068.2 Field Documentation

8.1068.2.1 uint8_t unpack_dms_UIMGetControlKeyStatus_t::facilityState

8.1068.2.2 uint16_t unpack_dms_UIMGetControlKeyStatus_t::TlvResult

8.1068.2.3 uint8_t unpack_dms_UIMGetControlKeyStatus_t::unblockRetriesLeft

8.1068.2.4 uint8_t unpack_dms_UIMGetControlKeyStatus_t::verifyRetriesLeft

8.1069 unpack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint8_t stringSize
- uint8_t String [255]
- uint16_t TlvResult

8.1069.1 Detailed Description

This structure contains Get ICCID pack

Parameters

stringSize	• Size of String.
String	• ICCID String.
TlvResult	• Pack result.

8.1069.2 Field Documentation

8.1069.2.1 uint8_t unpack_dms_UIMGetICCID_t::String[255]

8.1069.2.2 uint8_t unpack_dms_UIMGetICCID_t::stringSize

8.1069.2.3 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

8.1070 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

8.1070.1 Detailed Description

This structure contains Get PIN Status unpack information

Parameters

p1Status[OUT]	• PIN1 status(0xFFFFFFFF - Unknown) – 0 - PIN not initialized – 1 - PIN enabled, not verified – 2 - PIN enabled, verified – 3 - PIN disabled – 4 - PIN blocked – 5 - PIN permanently blocked
p1VerifyRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left of PIN1, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
p1Unblock-RetriesLeft[OUT]	• Optional parameter • 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. – 0xFFFFFFFF - Unknown
p2Status[OUT]	• PIN2 status(0xFFFFFFFF - Unknown) – 0 - PIN not initialized – 1 - PIN enabled, not verified – 2 - PIN enabled, verified – 3 - PIN disabled – 4 - PIN blocked – 5 - PIN permanently blocked

p2VerifyRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left of PIN2, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
p2Unblock-RetriesLeft[OUT]	• Optional parameter • 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. – 0xFFFFFFFF - Unknown
Tlvresult	• Unpack Result

8.1070.2 Field Documentation

8.1070.2.1 uint8_t unpack_dms_UIMGetPINStatus_t::p1Status

8.1070.2.2 uint8_t unpack_dms_UIMGetPINStatus_t::p1UnblockRetriesLeft

8.1070.2.3 uint8_t unpack_dms_UIMGetPINStatus_t::p1VerifyRetriesLeft

8.1070.2.4 uint8_t unpack_dms_UIMGetPINStatus_t::p2Status

8.1070.2.5 uint8_t unpack_dms_UIMGetPINStatus_t::p2UnblockRetriesLeft

8.1070.2.6 uint8_t unpack_dms_UIMGetPINStatus_t::p2VerifyRetriesLeft

8.1070.2.7 uint16_t unpack_dms_UIMGetPINStatus_t::Tlvresult

8.1071 unpack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

- uint8_t verifyRetriesLeft
- uint16_t Tlvresult

8.1071.1 Detailed Description

This structure contains UIM Set control key protection unpack information

Parameters

verifyRetries-Left[OUT]	• Optional parameter • number of retries left after which the control key is blocked
Tlvresult	• Unpack Result

8.1071.2 Field Documentation

8.1071.2.1 uint16_t unpack_dms_UIMSetControlKeyProtection_t::Tlvresult

8.1071.2.2 uint8_t unpack_dms_UIMSetControlKeyProtection_t::verifyRetriesLeft

8.1072 unpack_dms_UIMSetPINProtection_t Struct Reference

Data Fields

- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

8.1072.1 Detailed Description

This structure contains PIN retries status

Parameters

verifyRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
unblockRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. – 0xFFFFFFFF - Unknown
Tlvresult	• Unpack Result

8.1072.2 Field Documentation

8.1072.2.1 uint16_t unpack_dms_UIMSetPINProtection_t::Tlvresult

8.1072.2.2 uint8_t unpack_dms_UIMSetPINProtection_t::unblockRetriesLeft

8.1072.2.3 uint8_t unpack_dms_UIMSetPINProtection_t::verifyRetriesLeft

8.1073 unpack_dms_UIMUnblockControlKey_t Struct Reference

Data Fields

- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

8.1073.1 Detailed Description

This structure contains UIM Set control key protection unpack information

Parameters

unblockRetries-Left[OUT]	• Optional parameter • number of unblock retries left after which the control key is permanently blocked
Tlvresult	• Unpack Result

8.1073.2 Field Documentation

8.1073.2.1 uint16_t unpack_dms_UIMUnblockControlKey_t::Tlvresult

8.1073.2.2 uint8_t unpack_dms_UIMUnblockControlKey_t::unblockRetriesLeft

8.1074 unpack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint32_t ImageListSize
- FMSPrefImageList * pImageList
- uint16_t Tlvresult

8.1074.1 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 • See FMSImageElement
Tlvresult	• Unpack result

8.1074.2 Field Documentation

8.1074.2.1 uint32_t unpack_fms_GetImagesPreference_t::ImageListSize

8.1074.2.2 FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList

8.1074.2.3 uint16_t unpack_fms_GetImagesPreference_t::Tlvresult

8.1075 unpack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint32_t imagelistSize
- FMSImageList imageList
- uint16_t Tlvresult

8.1075.1 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 • See FMSImageElement
Tlvresult	• Unpack result

8.1075.2 Field Documentation

8.1075.2.1 FMSImageList unpack_fms_GetStoredImages_t::imageList

8.1075.2.2 uint32_t unpack_fms_GetStoredImages_t::imagelistSize

8.1075.2.3 uint16_t unpack_fms_GetStoredImages_t::Tlvresult

8.1076 unpack_fms_SetImagesPreference_t Struct Reference

Data Fields

- uint32_t ImageTypesSize
- uint8_t ImageTypes [255]
- uint16_t Tlvresult

8.1076.1 Detailed Description

This structure contains the Set Images Preference unpack

Parameters

ImageTypesSize	• Image Type Size
ImageTypes	• Image Type

Tlvresult	• Unpack result

8.1076.2 Field Documentation

8.1076.2.1 uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]

8.1076.2.2 uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize

8.1076.2.3 uint16_t unpack_fms_SetImagesPreference_t::Tlvresult

8.1077 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

8.1077.1 Detailed Description

This structure contains Best Available Position

Parameters

status	• Valid values: – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure
xid	Transaction ID that was specified in the Get Best Available Position request.
pLatitude	• Latitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude
pLongitude	• Longitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude
pHorUncCircular	• Horizontal position uncertainty. • Units - Meters
pAltitudeWrtEllipsoid	• Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883

pVertUnc	• Vertical uncertainty. • Units - Meters
pTimestampUtc	• UTC timestamp • Units - Milliseconds since Jan. 1, 1970
pTimeUnc	• Time uncertainty. • Units - Milliseconds
pHorUncEllipse-SemiMinor	• Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters
pHorUncEllipse-SemiMajor	• Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
pHorUncEllipse-OrientAzimuth	• Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180
pHorCirConf	• Horizontal circular uncertainty confidence • Units: Precent • Range: 0 to 99
pHorEllpConf	• Horizontal elliptical uncertainty confidence • Units: Precent • Range: 0 to 99
pHorReliability	• 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
pSpeed-Horizontal	• Horizontal speed. • Units - Meters/second
pSpeedUnc	• 3-D Speed uncertainty. • Units - Meters/second.

pAltitudeWrtMeanSeaLevel	• Altitude With Respect to Sea Level. • Units - Meters
pVertConfidence	• Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99
pVertReliability	• 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
pSpeedVertical	• Vertical speed. • Units - Meters/second
pSpeedVerticalUnc	• Veritical speed • Units: Meters/second
pHeading	• Heading. • Units - Degree • Range 0 to 359.999
pHeadingUnc	• Heading uncertainty. • Units - Degree • Range 0 to 359.999
pMagneticDeviation	• 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.
pTechnologyMask	• Values – 0x00000001 - Satellites were used to generate the fix – 0x00000002 - Cell towers were used to generate the fix – 0x00000004 - Wi-Fi access points were used to generate the fix – 0x00000008 - Sensors were used to generate the fix – 0x00000010 - Reference Location was used to generate the fix – 0x00000020 - Coarse position injected into the location engine was used to generate the fix – 0x00000040 - AFLT was used to generate the fix – 0x00000080 - GNSS and network-provided measurements were used to generate the fix

-pPrecision-Dilution	• See loc_precisionDilution for more information
-pGpsTime	• See loc_gpsTime for more information
pTimeSrc	• Values – 0 - Invalid time. – 1 - Time is set by the 1X system. – 2 - Time is set by WCDMA/GSM time tagging. – 3 - Time is set by an external injection. – 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite. – 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites. – 6 - Both time of the week and the GPS week number are known. – 7 - Time is set by the position engine after the fix is obtained – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites.
-pSensorData-Usage	• See loc_sensorDataUsage for more information
-pSvUsedforFix	• See loc_svUsedforFix for more information

8.1077.2 Field Documentation

8.1077.2.1 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid

8.1077.2.2 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel

8.1077.2.3 loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime

8.1077.2.4 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading

8.1077.2.5 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc

8.1077.2.6 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf

8.1077.2.7 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf

8.1077.2.8 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability

8.1077.2.9 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular

8.1077.2.10 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth

8.1077.2.11 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor

8.1077.2.12 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor

8.1077.2.13 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude

8.1077.2.14 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude

8.1077.2.15 uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation

8.1077.2.16 loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution

8.1077.2.17 loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage

8.1077.2.18 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal

8.1077.2.19 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc

8.1077.2.20 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical

8.1077.2.21 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc

8.1077.2.22 loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix

8.1077.2.23 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask

8.1077.2.24 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc

8.1077.2.25 uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc

8.1077.2.26 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc

8.1077.2.27 uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence

8.1077.2.28 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability

8.1077.2.29 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc

8.1077.2.30 uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid

8.1077.2.31 uint32_t unpack_loc_BestAvailPos_Ind_t::status

8.1077.2.32 uint16_t unpack_loc_BestAvailPos_Ind_t::TlvResult

8.1078 unpack_loc_Delete_Assist_Data_t Struct Reference

Data Fields

- uint16_t TlvResult

8.1078.1 Detailed Description

This structure contains LOC delete assist data unpack

Parameters

Tlvresult	• Unpack result.

8.1078.2 Field Documentation

8.1078.2.1 uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult

8.1079 unpack_loc_DeleteAssistData_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1079.1 Detailed Description

Contain the parameters passed for SetLocDeleteAssistDataCallback by the device.

Parameters

status	• Status of the Delete Assist Data request • Valid values: – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure
Tlvresult	• unpack result

8.1079.2 Field Documentation

8.1079.2.1 uint32_t unpack_loc_DeleteAssistData_Ind_t::status

8.1079.2.2 uint16_t unpack_loc_DeleteAssistData_Ind_t::Tlvresult

8.1080 unpack_loc_EngineState_Ind_t Struct Reference

Data Fields

- uint32_t engineState
- uint16_t Tlvresult

8.1080.1 Detailed Description

This structure contains LOC Engine State field.

Parameters

engineState	• Location engine state. • Valid values – 1 - Location engine is on – 2 - Loction engine is off
Tlvresult	• unpack result

8.1080.2 Field Documentation

8.1080.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

8.1080.2.2 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

8.1081 unpack_loc_EventRegister_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1081.1 Detailed Description

This structure contains Event Register unpack

Parameters

Tlvresult	• Unpack result.

8.1081.2 Field Documentation

8.1081.2.1 uint16_t unpack_loc_EventRegister_t::Tlvresult

8.1082 unpack_loc_GnssSvInfo_Ind_t Struct Reference

Data Fields

- uint8_t altitudeAssumed
- loc_satelliteInfo * pSatelliteInfo
- uint16_t Tlvresult

8.1082.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

Parameters

altitudeAssumed	• Indicates whether altitude is assumed or calculated – 0x00 (FALSE) - Valid altitude is calculated – 0x01 (TRUE) - Valid altitude is assumed; there may not be enough satellites to determine precise altitude
pSatelliteInfo	• See loc_satelliteInfo for more information.
Tlvresult	• unpack result

8.1082.2 Field Documentation

8.1082.2.1 uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed

8.1082.2.2 loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo

8.1082.2.3 uint16_t unpack_loc_GnssSvInfo_Ind_t::Tlvresult

8.1083 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

8.1083.1 Detailed Description

This structure contains Event Position Report Indication unpack

Parameters

sessionStatus	• Values – 0 - Session was successful – 1 - Session is still in progress; further position reports will be generated until either the fix criteria specified by the client are met or the client response timeout occurs. – 2 - Session failed.. – 3 - Fix request failed because the session timed out. – 4 - Fix request failed because the session was ended by the user. – 5 - Fix request failed due to bad parameters in the request. – 6 - Fix request failed because the phone is offline. – 7 - Fix request failed because the engine is locked
sessionId	• ID of the session that was specified in the Start request • Range - 0 to 255
pLatitude	• Latitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude

pLongitude	• Longitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude
pHorUncCircular	• Horizontal position uncertainty. • Units - Meters
pHorUncEllipse-SemiMinor	• Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters
pHorUncEllipse-SemiMajor	• Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
pHorUncEllipse-OrientAzimuth	• Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180
pHorConfidence	• Horizontal uncertainty confidence. • If both elliptical and horizontal uncertainties are specified in this message, the confidence corresponds to the elliptical uncertainty. • Units - Percentage • Range 0-99
pHorReliability	• 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
pSpeed-Horizontal	• Horizontal speed. • Units - Meters/second
pSpeedUnc	• 3-D Speed uncertainty. • Units - Meters/second.

pAltitudeWrt-Ellipsoid	• Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883
pAltitudeWrt-MeanSeaLevel	• Altitude With Respect to Sea Level. • Units - Meters
pVertUnc	• Vertical uncertainty. • Units - Meters
pVertConfidence	• Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99
pVertReliability	• 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
pSpeedVertical	• Vertical speed. • Units - Meters/second
pHeading	• Heading. • Units - Degree • Range 0 to 359.999
pHeadingUnc	• Heading uncertainty. • Units - Degree • 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 – 0x00000001 - Satellites were used to generate the fix – 0x00000002 - Cell towers were used to generate the fix – 0x00000004 - Wi-Fi access points were used to generate the fix – 0x00000008 - Sensors were used to generate the fix – 0x00000010 - Reference Location was used to generate the fix – 0x00000020 - Coarse position injected into the location engine was used to generate the fix – 0x00000040 - AFLT was used to generate the fix – 0x00000080 - GNSS and network-provided measurements were used to generate the fix
-pPrecision-Dilution	• See loc_precisionDilution for more information
pTimestampUtc	• UTC timestamp • 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. • Units - Seconds
-pGpsTime	• See loc_gpsTime for more information
pTimeUnc	• Time uncertainty. • Units - Milliseconds

pTimeSrc	• Values – 0 - Invalid time. – 1 - Time is set by the 1X system. – 2 - Time is set by WCDMA/GSM time tagging. – 3 - Time is set by an external injection. – 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite. – 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites. – 6 - Both time of the week and the GPS week number are known. – 7 - Time is set by the position engine after the fix is obtained – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites.
-pSensorData-Usage	• See loc_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 loc_svUsedforFix for more information
pAltitude-Assumed	• Indicates whether altitude is assumed or calculated.

- Value
  - 0x00 - Altitude is calculated
  - 0x01 - Altitude is assumed

8.1083.2 Field Documentation

8.1083.2.1 uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed

8.1083.2.2 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid

8.1083.2.3 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel

8.1083.2.4 uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId

8.1083.2.5 loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime

8.1083.2.6 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading

8.1083.2.7 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc

8.1083.2.8 uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence

8.1083.2.9 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability

8.1083.2.10 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular

8.1083.2.11 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth

8.1083.2.12 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor

8.1083.2.13 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor

8.1083.2.14 uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude

8.1083.2.15 uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds

8.1083.2.16 uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude

8.1083.2.17 uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation

8.1083.2.18 loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution

8.1083.2.19 loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage

8.1083.2.20 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal

8.1083.2.21 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc

8.1083.2.22 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical

8.1083.2.23 loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix

8.1083.2.24 uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask

8.1083.2.25 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc

8.1083.2.26 uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc

8.1083.2.27 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc

8.1083.2.28 uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence

8.1083.2.29 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability

8.1083.2.30 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc

8.1083.2.31 uint8_t unpack_loc_PositionRpt_Ind_t::sessionId

8.1083.2.32 uint32_t unpack_loc_PositionRpt_Ind_t::sessionStatus

8.1083.2.33 uint16_t unpack_loc_PositionRpt_Ind_t::TlvResult

8.1084 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1084.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

Parameters

status	• Valid values – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure
Tlvresult	• unpack result

8.1084.2 Field Documentation

8.1084.2.1 uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status

8.1084.2.2 uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult

8.1085 unpack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1085.1 Detailed Description

This structure contains Set Ext Power State unpack

Parameters

Tlvresult	• Unpack result.

8.1085.2 Field Documentation

8.1085.2.1 uint16_t unpack_loc_SetExtPowerState_t::Tlvresult

8.1086 unpack_loc_SetOperationMode_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1086.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

Parameters

status	• Status of the Set Operation Mode request. • Valid values: – 0 - Request was completed successfully – 1 - Request failed because of a general failure – 2 - Request failed because it is not supported – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - Request failed because the engine could not allocate sufficient memory for the request – 9 - Request failed because the maximum number of Geofences are already programmed – 10 - Location service failed because of an XTRA version-based file format check failure
Tlvresult	• unpack result

8.1086.2 Field Documentation

8.1086.2.1 uint32_t unpack_loc_SetOperationMode_Ind_t::status

8.1086.2.2 uint16_t unpack_loc_SetOperationMode_Ind_t::Tlvresult

8.1087 unpack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1087.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

Tlvresult	• Unpack result.

8.1087.2 Field Documentation

8.1087.2.1 uint16_t unpack_loc_SetOperationMode_t::Tlvresult

8.1088 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1088.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

Tlvresult	• Unpack result.

8.1088.2 Field Documentation

8.1088.2.1 uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.1089 unpack_loc_Start_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1089.1 Detailed Description

This structure contains Start LOC unpack

Parameters

Tlvresult	• Unpack result.

8.1089.2 Field Documentation

8.1089.2.1 uint16_t unpack_loc_Start_t::Tlvresult

8.1090 unpack_loc_Stop_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1090.1 Detailed Description

This structure contains Stop LOC unpack

Parameters

Tlvresult	• Unpack result.

8.1090.2 Field Documentation

8.1090.2.1 uint16_t unpack_loc_Stop_t::Tlvresult

8.1091 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

8.1091.1 Detailed Description

Parameters

SCI	slot cycle index
SCM	station class mark
RegHomeSID	register on home sid
RegForeignSID	register on foreign sid
RegForeignNID	register on foreign nid
ForceRev0	force header revision
CustomSCP	custom SCP
Protocol	protocol
Booadcast	broadcast
Application	application
Roaming	roaming

8.1091.2 Field Documentation

8.1091.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

8.1091.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

8.1091.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

8.1091.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

8.1091.2.5 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

8.1091.2.6 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

8.1091.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

8.1091.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

8.1091.2.9 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

8.1091.2.10 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

8.1091.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

8.1092 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

8.1092.1 Detailed Description

Structure for HomeNetwork3GPP2 unpack.

Parameters

pMCC	• Mobile country code (UMTS only).
pMNC	• Mobile network code (UMTS only).
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).
pSID	• Home network system ID – 0xFFFF - Unknown. – Only applies to cdma2000
pNID	• Home network ID – 0xFFFF - Unknown. – Only applies to cdma2000
pNw2MCC	• Mobile country code (3GPP2 only). • Range : 0 to 999
pNw2MNC	• Mobile network code (3GPP2 only). • Range : 0 to 999
pNw2DescDisp	• Network Name Display (3GPP2 only). -Valid Value – 0x00 - Do not display – 0x01 - Display – 0xFF - Unknown
pNw2DescEnc	• Encoding of the network description (3GPP2 only). • Valid Value – 0x00 - Octet, unspecified – 0x02 - 7-bit ASCII – 0x04 - Unicode – 0x09 - GSM 7-bit default
nw2DescLen	• Network Description Length (3GPP2 only).
pNw2Name	• Network Name (3GPP2 only).

8.1092.2 Field Documentation

8.1092.2.1 uint8_t unpack_nas_GetHomeNetwork3GPP2_t::nameSize

8.1092.2.2 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMCC

8.1092.2.3 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMNC

8.1092.2.4 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pName

8.1092.2.5 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNID

8.1092.2.6 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescDisp

8.1092.2.7 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescEnc

8.1092.2.8 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescLen

8.1092.2.9 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MCC

8.1092.2.10 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MNC

8.1092.2.11 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2Name

8.1092.2.12 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pSID

8.1093 unpack_nas_GetHomeNetwork_t Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- char name [255]
- uint16_t sid
- uint16_t nid

8.1093.1 Detailed Description

Parameters

mcc	mobile country code
mnc	mobile network code
name	network name or description
sid	home network system id only applies to cdma2000
nid	home network id Only applies to cdma2000

8.1093.2 Field Documentation

8.1093.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

8.1093.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

8.1093.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

8.1093.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

8.1093.2.5 uint16_t unpack_nas_GetHomeNetwork_t::sid

8.1094 unpack_nas_GetNetworkPreference_t Struct Reference

Data Fields

- uint32_t ActiveTechPref
- uint32_t Duration
- uint32_t PersistentTechPref
- uint16_t Tlvresult

8.1094.1 Detailed Description

Parameters

TechnologyPref[-OUT]	• Bitmask representing the radio technology preference set. • No bits set indicates to the device to automatically determine the technology to use • Values: – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved
Duration[OUT]	• Duration of active preference – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call
PersistentTechnologyPref[-OUT]	• Bit field representing persistent radio technology preference – Same representation as the pTechnologyPref parameter
Tlvresult	• unpack result

8.1094.2 Field Documentation

8.1094.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

8.1094.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

8.1094.2.3 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

8.1094.2.4 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

8.1095 unpack_nas_GetRFInfo_t Struct Reference

Data Fields

- uint8_t instancesSize
- RFBandInfoElements RFBandInfoElements [255]

8.1095.1 Detailed Description

Parameters

instancesSize	number of elements in RF info instances array.
RFBandInfo-Elements	RF info instances array

8.1095.2 Field Documentation

8.1095.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

8.1095.2.2 RFBandInfoElements unpack_nas_GetRFInfo_t::RFBandInfoElements[255]

8.1096 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]

8.1096.1 Detailed Description

Parameters

Registration-State	registration state
CSDomain	CS domain
PSDomain	PS domain
RAN	radio access network
RadioIfacesSize	radio interface size
RadioIfaces	radio interface list
Roaming	romaing indicator
MCC	Mobile country code
MNC	Mobile network code
nameSize	network name size
Name	network name
DataCapsLen	data capabilities len
DataCap	data capabilities

8.1096.2 Field Documentation

8.1096.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

8.1096.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

8.1096.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

8.1096.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

8.1096.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

8.1096.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

8.1096.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

8.1096.2.8 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

8.1096.2.9 uint8_t unpack_nas_GetServingNetwork_t::RadioIfaces[255]

8.1096.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadioIfacesSize

8.1096.2.11 uint32_t unpack_nas_GetServingNetwork_t::RAN

8.1096.2.12 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

8.1096.2.13 uint32_t unpack_nas_GetServingNetwork_t::Roaming

8.1097 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

Data Fields

- uint8_t DataCapsLen
- uint8_t DataCaps [255]

8.1097.1 Detailed Description

Parameters

DataCapsLen	data capabilities len
DataCap	data capabilities

8.1097.2 Field Documentation

8.1097.2.1 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]

8.1097.2.2 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen

8.1098 unpack_nas_GetSignalStrengths_t Struct Reference

Data Fields

- uint32_t len
- signed char rssi [8]

- uint32_t radio [8]

8.1098.1 Detailed Description

Parameters

len	number of rssi & radio items following
rssi	signal strength array
radio	radio interface array

8.1098.2 Field Documentation

8.1098.2.1 uint32_t unpack_nas_GetSignalStrengths_t::len

8.1098.2.2 uint32_t unpack_nas_GetSignalStrengths_t::radio[8]

8.1098.2.3 signed char unpack_nas_GetSignalStrengths_t::rssi[8]

8.1099 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

8.1099.1 Detailed Description

Parameters

InstanceSize	total instances
Instances	info for instances

8.1099.2 Field Documentation

8.1099.2.1 nas_QmiNas3GppNetworkInfo* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances

8.1099.2.2 uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize

8.1099.2.3 nas_QmisNasPcsDigit* unpack_nas_PerformNetworkScan_t::pPCSInstance

8.1099.2.4 uint8_t* unpack_nas_PerformNetworkScan_t::pPCSInstanceSize

8.1099.2.5 nas_QmiNas3GppNetworkRAT* unpack_nas_PerformNetworkScan_t::pRATINstance

8.1099.2.6 uint8_t* unpack_nas_PerformNetworkScan_t::pRATInstanceSize

8.1099.2.7 uint32_t* unpack_nas_PerformNetworkScan_t::pScanResult

8.1100 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference

Data Fields

- uint8_t dataCapsSize
- uint8_t dataCaps [255]

8.1100.1 Detailed Description

Parameters

dataCapsSize	Number of Data Capabilities
dataCaps	Data Capabilities

8.1100.2 Field Documentation

8.1100.2.1 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]

8.1100.2.2 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize

8.1101 unpack_nas_SetEventReportInd_t Struct Reference

Data Fields

- nas_SignalStrengthTlv SSTlv
- nas_RFInfoTlv RFTlv
- nas_RejectReasonTlv RRTlv
- nas_SLQSSignalStrengthsTlv SLQSSSTlv

8.1101.1 Detailed Description

Parameters

SSTlv	signal strength tlv
RFTlv	RF tlv
RRTlv	RR tlv
SLQSSSTlv	signal strength complete info tlv

8.1101.2 Field Documentation

8.1101.2.1 nas_RFInfoTlv unpack_nas_SetEventReportInd_t::RFTlv

8.1101.2.2 nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv

8.1101.2.3 nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv

8.1101.2.4 nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv

8.1102 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference

Data Fields

- nas_PhyCaAggScellIndType sPhyCaAggScellIndType

- nas_PhyCaAggScellDlBw sPhyCaAggScellDlBw
- nas_PhyCaAggScellInfo sPhyCaAggScellInfo
- nas_PhyCaAggPcellInfo sPhyCaAggPcellInfo
- nas_PhyCaAggScellIndex sPhyCaAggScellIndex

8.1102.1 Detailed Description

Structure for storing the LTEC 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.

8.1102.2 Field Documentation

8.1102.2.1 nas_PhyCaAggPcellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggPcellInfo

8.1102.2.2 nas_PhyCaAggScellDlBw unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellDlBw

8.1102.2.3 nas_PhyCaAggScellIndex unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellIndex

8.1102.2.4 nas_PhyCaAggScellIndType unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellIndType

8.1102.2.5 nas_PhyCaAggScellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellInfo

8.1103 unpack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1103.1 Detailed Description

Parameters

TechnologyPref[-OUT]	• Bitmask representing the radio technology preference set. • No bits set indicates to the device to automatically determine the technology to use • Values: – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved
Duration[OUT]	• Duration of active preference – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call
Persistent-TechnologyPref[-OUT]	• Bit field representing persistent radio technology preference – Same representation as the pTechnologyPref parameter
Tlvresult	• unpack result

8.1103.2 Field Documentation

8.1103.2.1 uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult

8.1104 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

Data Fields

- uint8_t roaming

8.1104.1 Detailed Description

Parameters

roaming	• Roaming Indication – 0 - Roaming – 1 - Home – 2 - Roaming partner – >2 - Operator defined values

8.1104.2 Field Documentation

8.1104.2.1 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

8.1105 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

Data Fields

- NASServingSystemInfo SSInfo
- uint16_t Tlvresult

8.1105.1 Detailed Description

Parameters

SSInfo	• Serving system parameters information – See NASServingSystemInfo for more details
Tlvresult	• unpack result

8.1105.2 Field Documentation

8.1105.2.1 NASServingSystemInfo unpack_nas_SetServingSystemCallback_ind_t::SSInfo

8.1105.2.2 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

8.1106 unpack_nas_SLQSGetErrorRate_t Struct Reference

Data Fields

- uint16_t * pCDMAFrameErrRate
- uint16_t * pHDRPackErrRate
- uint8_t * pGSMBER
- uint8_t * pWCDMABER

8.1106.1 Detailed Description

Structure for Error Rate unpack parameters.

Parameters

pCDMAFrame-ErrRate	• CDMA Frame Error Rate • 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%. • A value of 0xFFFF indicates that the error rate is unknown/unavailable.

pHDRPackErr-Rate	• HDR Packet Error Rate • 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%. • A value of 0xFFFF indicates that the error rate is unknown/unavailable.
pGSMBER	• GSM Bit Error Rate • Valid error rate values between 1 and 100 are returned to indicate the percentage value. • A 0% block error rate (BLER) indicates No Data.
pWCDMABER	• WCDMA Block Error Rate • Valid error rate values between 1 and 100 are returned to indicate the percentage value. • A value of 0xFF indicates that the error rate is unknown/unavailable.

8.1106.2 Field Documentation

8.1106.2.1 uint16_t* unpack_nas_SLQSGetErrorRate_t::pCDMAFrameErrRate

8.1106.2.2 uint8_t* unpack_nas_SLQSGetErrorRate_t::pGSMBER

8.1106.2.3 uint16_t* unpack_nas_SLQSGetErrorRate_t::pHDRPackErrRate

8.1106.2.4 uint8_t* unpack_nas_SLQSGetErrorRate_t::pWCDMABER

8.1107 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

Data Fields

- NasGetLTECphyCaInfo LTECphyCaInfo
- uint16_t Tlvresult

8.1107.1 Detailed Description

Parameters

LTECphyCa	• Carrier aggregation event information – See NasGetLTECphyCaInfo for more details
Tlvresult	• unpack result

8.1107.2 Field Documentation

8.1107.2.1 NasGetLTECphyCaInfo unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCaInfo

8.1107.2.2 uint16_t unpack_nas_SlqsGetLTECphyCAInfo_t::Tlvresult

8.1108 unpack_nas_SLQSGetNetworkTime_t Struct Reference

Data Fields

- nas_timeInfo * p3GPP2TimeInfo
- nas_timeInfo * p3GPPTimeInfo

8.1108.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

Parameters

p3GPP2TimeInfo	[Optional] • See nas_timeInfo for more information
p3GPPTimeInfo	[Optional] • See nas_timeInfo for more information

8.1108.2 Field Documentation

8.1108.2.1 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPP2TimeInfo

8.1108.2.2 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPPTimeInfo

8.1109 unpack_nas_SLQSGetOperatorNameData_t Struct Reference

Data Fields

- nas_serviceProviderName * pSrvcProviderName
- nas_operatorPLMNList * pOperatorPLMNList
- nas_PLMNNetworkName * pPLMNNetworkName
- nas_operatorNameString * pOperatorNameString
- nas_PLMNNetworkNameData * pNITZInformation

8.1109.1 Detailed Description

Structure for Operator Name unpack.

Parameters

pSrvcProviderName	• Refer nas_serviceProviderName for details (Optional). • Can provide NULL if this parameter is not required.
pOperatorPLMNList	• Refer nas_operatorPLMNList for details (Optional). • Can provide NULL if this parameter is not required.
pPLMNNetworkName	• Refer nas_PLMNNetworkName for details (Optional). • Can provide NULL if this parameter is not required.

pOperatorName-String	• Refer nas_operatorNameString for details (Optional). • Can provide NULL if this parameter is not required.
pNITZ-Information	• Refer nas_PLMNNetworkNameData for details (Optional). • Can provide NULL if this parameter is not required.

8.1109.2 Field Documentation

8.1109.2.1 nas_PLMNNetworkNameData* unpack_nas_SLQSGetOperatorNameData_t::pNITZInformation

8.1109.2.2 nas_operatorNameString* unpack_nas_SLQSGetOperatorNameData_t::pOperatorNameString

8.1109.2.3 nas_operatorPLMNList* unpack_nas_SLQSGetOperatorNameData_t::pOperatorPLMNList

8.1109.2.4 nas_PLMNNetworkName* unpack_nas_SLQSGetOperatorNameData_t::pPLMNNetworkName

8.1109.2.5 nas_serviceProviderName* unpack_nas_SLQSGetOperatorNameData_t::pSrvcProviderName

8.1110 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]

8.1110.1 Field Documentation

8.1110.1.1 char unpack_nas_SLQSGetPLMNName_t::longName[255]

8.1110.1.2 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameCI

8.1110.1.3 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameEn

8.1110.1.4 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameLen

8.1110.1.5 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameSB

8.1110.1.6 uint8_t unpack_nas_SLQSGetPLMNName_t::shortName[255]

8.1110.1.7 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameCI

8.1110.1.8 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameEn

8.1110.1.9 char unpack_nas_SLQSGetPLMNName_t::shortNameLen

8.1110.1.10 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameSB

8.1110.1.11 char unpack_nas_SLQSGetPLMNName_t::spn[255]

8.1110.1.12 uint8_t unpack_nas_SLQSGetPLMNName_t::spnEncoding

8.1110.1.13 uint8_t unpack_nas_SLQSGetPLMNName_t::spnLength

8.1111 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

8.1111.1 Detailed Description

Parameters

ServingSystem	serving system info
RoamIndicator-Val	roaming indicator value
DataSrv-Capabilities	data servcie capabilities
CurrentPLMN	current PLMN info
SystemID	system id

NetworkID	network id
BasestationID	base station id
Basestation-Latitude	base station latitude
Basestation-Longitude	base station longitude
Roaming-IndicatorList	roaming indicator list
DefaultRoamInd	default roaming indicator
3Gpp2TimeZone	3Gpp2 time zone
pCDMA_P_Rev	cdma P_Rev in use
3GppTimeZone	3Gpp time zone
GppNetworkDS-TAdjustment	3GPP network daylight saving adjustment
Lac	location area code
CellID	3GPP cell id
ConcSvcInfo	3GPP2 concurrent servcie info
PRLInd	3GPP2 PRL indicator
DTMInd	DTM indicator(GSM)
DetailedSvcInfo	detail servcie info
CDMASystem-InfoExt	extra cdma system info
HdrPersonality	hdr personality
TrackAreaCode	track area code
CallBarStatus	call barring status

8.1111.2 Field Documentation

8.1111.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID

8.1111.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude

8.1111.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude

8.1111.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus

8.1111.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev

8.1111.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt

8.1111.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID

8.1111.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo

8.1111.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN

8.1111.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities

8.1111.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd

8.1111.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo

8.1111.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd

8.1111.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone

8.1111.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment

8.1111.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone

8.1111.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality

8.1111.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac

8.1111.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID

8.1111.2.20 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd

8.1111.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal

8.1111.2.22 nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList

8.1111.2.23 nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem

8.1111.2.24 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID

8.1111.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

8.1112 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

8.1112.1 Detailed Description

Parameters

rxSignal-StrengthListLen	number of elements in Receive Signal Strength List
rxSignal-StrengthList	signal strength list
ecioListLen	number of elements in ECIO List
ecioList	ecio list
Io	received Io in dBm; IO is only applicable for 1xEV-DO
sinr	SINR level; SINR is only applicable for 1xEV-DO
errorRateListLen	number of elements in Error Rate List
errorRateList	error rate list

rsrqInfo	rsrq info
ltesnr	lte snr info
ltersrp	lte srp info

8.1112.2 Field Documentation

8.1112.2.1 nas_ecioListElement unpack_nas_SLQSGetSignalStrength_t::ecioList[18]

8.1112.2.2 uint16_t unpack_nas_SLQSGetSignalStrength_t::ecioListLen

8.1112.2.3 nas_errorRateListElement unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]

8.1112.2.4 uint16_t unpack_nas_SLQSGetSignalStrength_t::errorRateListLen

8.1112.2.5 int32_t unpack_nas_SLQSGetSignalStrength_t::lo

8.1112.2.6 int16_t unpack_nas_SLQSGetSignalStrength_t::ltersrp

8.1112.2.7 int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr

8.1112.2.8 nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo

8.1112.2.9 nas_rxSignalStrengthListElement unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]

8.1112.2.10 uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen

8.1112.2.11 uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask

8.1112.2.12 uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr

8.1113 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

8.1113.1 Detailed Description

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. • 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. • 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. – 0x00 - Voice is not supported – 0x01 - Voice is supported
pGSMCipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pWCDMACipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched

8.1113.2 Field Documentation

8.1113.2.1 nas_AddCDMASysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo

8.1113.2.2 nas_AddSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo

8.1113.2.3 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo

8.1113.2.4 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo

8.1113.2.5 nas_AddSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo

8.1113.2.6 nas_SrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo

8.1113.2.7 nas_CDMASysInfo∗ unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo

8.1113.2.8 nas_CallBarringSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo

8.1113.2.9 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo

8.1113.2.10 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMSrvStatusInfo

8.1113.2.11 nas_GSMSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pGSMSysInfo

8.1113.2.12 nas_SrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo

8.1113.2.13 nas_HDRSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo

8.1113.2.14 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo

8.1113.2.15 **nas_LTESysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pLTESysInfo

8.1113.2.16 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo

8.1113.2.17 **nas_CallBarringSysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo

8.1113.2.18 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo

8.1113.2.19 **nas_GSMSrvStatusInfo**∗ unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo

8.1113.2.20 **nas_WCDMASysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo

8.1114 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

8.1114.1 Detailed Description

Parameters

pEmerMode	• Optional parameter specifying the emergency Mode • Values: – 0 - OFF (normal) – 1 - ON (Emergency)
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE

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

pPRLPref	• Optional parameter indicating the CDMA PRL Preference • Values: – 0x0001 - Acquire available system only on the A side – 0x0002 - Acquire available system only on the B side – 0x3FFF - Acquire any available systems
pRoamPref	• Optional parameter indicating the roaming Preference • Values: – 0x01 - Acquire only systems for which the roaming indicator is off – 0x02 - Acquire a system as long as its roaming indicator is not off – 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only – 0xFF - Acquire systems, regardless of their roaming indicator
pLTEBandPref	• Optional parameter • Bit mask representing the LTE band preference • Bit Values – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – All other bits are reserved

pNetSelPref	• Optional parameter indicating network selection preference • Values: – 0x00 - Automatic network selection – 0x01 - Manual network selection.
pSrvDomainPref	• Optional parameter indicating Service domain preference • Values: – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach
pGWAcqOrder-Pref	• Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM

8.1114.2 Field Documentation

8.1114.2.1 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

8.1114.2.2 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

8.1114.2.3 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

8.1114.2.4 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

8.1114.2.5 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref

8.1114.2.6 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

8.1114.2.7 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

8.1114.2.8 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

8.1114.2.9 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

8.1115 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

8.1115.1 Detailed Description

Structure for 3GPP2Subscription unpack.

Parameters

pNAMNameInfo	[Optional] • See nas_namName for more information
pDirNum	[Optional] • See nas_dirNum for more information
pHomeSIDNID	[Optional] • See nas_homeSIDNID for more information
pMinBasedIMSI	[Optional] • See naS_minBasedIMSI for more information
pTrueIMSI	[Optional] • See nas_trueIMSI for more information
pCDMAChannel	[Optional] • See nas_CDMAChannel for more information

8.1115.2 Field Documentation

8.1115.2.1 nas_CDMAChannel* unpack_nas_SLQSNasGet3GPP2Subscription_t::pCDMAChannel

8.1115.2.2 nas_dirNum* unpack_nas_SLQSNasGet3GPP2Subscription_t::pDirNum

8.1115.2.3 nas_homeSIDNID* unpack_nas_SLQSNasGet3GPP2Subscription_t::pHomeSIDNID

8.1115.2.4 nas_minBasedIMSI* unpack_nas_SLQSNasGet3GPP2Subscription_t::pMinBasedIMSI

8.1115.2.5 nas_namName* unpack_nas_SLQSNasGet3GPP2Subscription_t::pNAMNameInfo

8.1115.2.6 nas_trueIMSI* unpack_nas_SLQSNasGet3GPP2Subscription_t::pTrueIMSI

8.1116 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

8.1116.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.

Parameters

pGERANInfo	• See nas_GERANInfo for more information.
pUMTSInfo	• See nas_UMTSInfo for more information.
pCDMAInfo	• See nas_CDMAInfo for more information.
pLTEInfo-Intrafreq	• See nas_LTEInfoIntrafreq for more information.
pLTEInfo-Interfreq	• See nas_LTEInfoInterfreq for more information.
pLTEInfo-NeighboringGS-M	• See nas_LTEInfoNeighboringGSM for more information.
pLTEInfo-NeighboringWC-DMA	• See nas_LTEInfoNeighboringWCDMA for more information.
pUMTSCellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
pWCDMAInfoLT-ENeighborCell	• See nas_WCDMAInfoLTENeighborCell for more information.

8.1116.2 Field Documentation

8.1116.2.1 nas_CDMAInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo

8.1116.2.2 nas_GERANInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo

8.1116.2.3 nas_LTEInfoInterfreq∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq

8.1116.2.4 nas_LTEInfoIntrafreq∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq

8.1116.2.5 nas_LTEInfoNeighboringGSM∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM

8.1116.2.6 nas_LTEInfoNeighboringWCDMA∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA

8.1116.2.7 uint32_t∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID

8.1116.2.8 nas_UMTSInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo

8.1116.2.9 nas_WCDMAInfoLTENeighborCell∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

8.1117 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference

Data Fields

- uint8_t ∗ pColorCode

8.1117.1 Detailed Description

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

Parameters

pColorCode	[Optional] • Color code value • Color code corresponding to the sector to which the AT is sending the access probe • See 3GPP2 C.S0024-B V3.0, Section 7.11.6.2.1 for more information. – 0xFF - Not Available

8.1117.2 Field Documentation

8.1117.2.1 uint8_t∗ unpack_nas_SLQSNasGetHDRColorCode_t::pColorCode

8.1118 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

Data Fields

- cdmaSSInfo CDMASSInfo
- hdrSSInfo HDRSSInfo
- int8_t GSMSSInfo
- cdmaSSInfo WCDMASSInfo
- lteSSInfo LTESSInfo

8.1118.1 Detailed Description

Parameters

CDMASSInfo	CDMA Signal Strength Information
HDRSSInfo	HDR Signal Strength Information
GSMSSInfo	GSM signal strength is the RSSI in dBm.
WCDMASSInfo	WCDMA Signal Strength Information
LTESSInfo	LTE Signal Strength Information

8.1118.2 Field Documentation

8.1118.2.1 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo

8.1118.2.2 int8_t unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo

8.1118.2.3 hdrSSInfo unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo

8.1118.2.4 lteSSInfo unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo

8.1118.2.5 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo

8.1119 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- nas_rxInfo * pRXChain0Info
- nas_rxInfo * pRXChain1Info
- nas_txInfo * pTXInfo

8.1119.1 Detailed Description

Structure for TxRx Info unpack

Parameters

pRXChain0Info	[Optional] • See nas_rxInfo for more information.
pRXChain1Info	[Optional] • See nas_rxInfo for more information.
pTXInfo	[Optional] • See nas_txInfo for more information.

8.1119.2 Field Documentation

8.1119.2.1 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain0Info

8.1119.2.2 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain1Info

8.1119.2.3 nas_txInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pTXInfo

8.1120 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference

Data Fields

- nas_UniversalTime universalTime
- uint8_t * pTimeZone
- uint8_t * pDayltSavAdj
- uint8_t * pRadioInterface

8.1120.1 Detailed Description

Structure for storing the NAS Network Time indication parameters.

Parameters

universalTime	• See nas_UniversalTime for more information.
pTimeZone	• Time Zone. • Offset from Universal time, i.e., the difference between local time and Universal time, in increments of 15 min (signed value).
pDayltSavAdj	• Daylight Saving Adjustment. • Daylight saving adjustment in hr. – Possible values: 0, 1, and 2.
pRadioInterface	• Radio interface from which the information comes • Values – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA -TD-SCDMA

8.1120.2 Field Documentation

8.1120.2.1 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

8.1120.2.2 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

8.1120.2.3 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

8.1120.2.4 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

8.1121 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

8.1121.1 Detailed Description

Parameters

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

8.1121.2 Field Documentation

8.1121.2.1 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo

8.1121.2.2 int8_t∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo

8.1121.2.3 hdrSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo

8.1121.2.4 lteSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo

8.1121.2.5 int8_t∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp

8.1121.2.6 tdscdmaSigInfoExt∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt

8.1121.2.7 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo

8.1122 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference

Data Fields

- nas_wcdmaUARFCN ∗ pWcdmaUARFCN
- nas_lteEARFCN ∗ pLteEARFCN
- nas_ltePCI ∗ pLtePCI

8.1122.1 Detailed Description

Structure for Get Channel Lock unpack.

Parameters

pWcdmaUARF-CN	[Optional] • See nas_wcdmaUARFCN for more information
pLteEARFCN	[Optional] • See nas_lteEARFCN for more information
pLtePCI	[Optional] • See nas_ltePCI for more information

8.1122.2 Field Documentation

8.1122.2.1 nas_lteEARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLteEARFCN

8.1122.2.2 nas_ltePCI∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLtePCI

8.1122.2.3 nas_wcdmaUARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pWcdmaUARFCN

8.1123 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

Data Fields

- nas_CommInfo commonInfo
- nas_LTEInfo ∗ pLTEInfo

8.1123.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

Parameters

commonInfo	(mandatory) • See CommInfo for more information
pLTEInfo	(optional) • See LTEInfo for more information

8.1123.2 Field Documentation

8.1123.2.1 nas_CommInfo unpack_nas_SLQSNasSwiModemStatus_t::commonInfo

8.1123.2.2 nas_LTEInfo∗ unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo

8.1124 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference

Data Fields

- NASQmiCbkNasSwiOTAMessageInd Info
- uint16_t Tlvresult

8.1124.1 Detailed Description

Parameters

Info	• Structure used to store all QMI Notification Info. – See NASQmiCbkNasSwiOTAMessageInd for more details
Tlvresult	• unpack result

8.1124.2 Field Documentation

8.1124.2.1 NASQmiCbkNasSwiOTAMessageInd unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info

8.1124.2.2 uint16_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::TlvResult

8.1125 unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference

Data Fields

- char t3396_apn [101]
- uint8_t t3396_plmn_id [3]
- uint32_t t3396_val

8.1125.1 Detailed Description

Structure for Network Timer indication parameters.

Parameters

t3396_apn	• apn
t3396_plmn_id	• plmn id
t3396_val	• timer value

8.1125.2 Field Documentation

8.1125.2.1 char unpack_nas_SLQSNasTimerCallback_ind_t::t3396_apn[101]

8.1125.2.2 uint8_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_plmn_id[3]

8.1125.2.3 uint32_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_val

8.1126 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference

Data Fields

- NASQmiCbkNasSystemSelPrefInd Info
- uint16_t TlvResult

8.1126.1 Detailed Description

Parameters

Info	• Structure used to store all QMI Notification Info. – See NASQmiCbkNasSystemSelPrefInd for more details
TlvResult	• unpack result

8.1126.2 Field Documentation

8.1126.2.1 NASQmiCbkNasSystemSelPrefInd unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

8.1126.2.2 uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::TlvResult

8.1127 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference

Data Fields

- uint16_t * pCurrentPersonality
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtocolSubtypeElement

8.1127.1 Detailed Description

Structure for HDRPersonality unpack.

Parameters

pCurrentPersonality[Out]	• Current active personality index.
pPersonalityListLength[In/Out]	• Number of Personality Protocol Subtype contains in this response. • maximum input value is 3
pProtocolSubtypeElement[Out]	• See nas_protocolSubtypeElement for more information.

8.1127.2 Field Documentation

8.1127.2.1 uint16_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pCurrentPersonality

8.1127.2.2 uint8_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pPersonalityListLength

8.1127.2.3 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRPersonality_t::pProtocolSubtypeElement

8.1128 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference

Data Fields

- uint16_t * pCurrentPrsnlty
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtoSubTypElmnt
- uint64_t * pAppSubType

8.1128.1 Detailed Description

Structure for HDRProtSubtype unpack.

Parameters

pCurrent-Personality	• Current active personality index.
pPersonalityList-Length	• Number of Personality Protocol Subtype contains in this response. • maximum input value is 4
pProtocol-SubtypeElement	• See nas_protocolSubtypeElement for more information.
pAppSubType	• Stream application subtype • Application subtype for each stream,

8.1128.2 Field Documentation

8.1128.2.1 uint64_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pAppSubType

8.1128.2.2 uint16_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pCurrentPrsnlty

8.1128.2.3 uint8_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pPersonalityListLength

8.1128.2.4 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pProtoSubTypElmnt

8.1129 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference

Data Fields

- nas_DRCParams * pDRCParams
- uint8_t * pUATI
- nas_PilotSetData * pPilotSetData

8.1129.1 Detailed Description

Structure for HRPD Stats unpack.

Parameters

pDRCParams	• See nas_DRCParams for more information.
pUATI	• A 128-bit address that includes the access terminal identifier and subnet ID • Size must be 16 bytes
pPilotSetData	• See nas_PilotSetData for more information.

8.1129.2 Field Documentation

8.1129.2.1 nas_DRCParams* unpack_nas_SLQSSwiGetHRPDStats_t::pDRCParams

8.1129.2.2 nas_PilotSetData* unpack_nas_SLQSSwiGetHRPDStats_t::pPilotSetData

8.1129.2.3 uint8_t* unpack_nas_SLQSSwiGetHRPDStats_t::pUATI

8.1130 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

Data Fields

- uint8_t ValidityCW0
- uint8_t CQIValueCW0
- uint8_t ValidityCW1
- uint8_t CQIValueCW1

8.1130.1 Detailed Description

Parameters

ValidityCW0[OUT]	• Values – 0- Invalid. – 1- Valid.
CQIValueCW0[OUT]	• Values – Range 0∼15
ValidityCW1[OUT]	• Values – 0- Invalid. – 1- Valid.
CQIValueCW1[OUT]	• Values – Range 0∼15

8.1130.2 Field Documentation

8.1130.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

8.1130.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

8.1130.2.3 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

8.1130.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

8.1131 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference

Data Fields

- nas_SccRxInfo * pSccRxInfo

8.1131.1 Detailed Description

Parameters

pSccRxInfo	Secondary carrier Rx signal level info

8.1131.2 Field Documentation

8.1131.2.1 nas_SccRxInfo∗ unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo

8.1132 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference

Data Fields

- uint8_t ∗ pObjectVer
- nas_NetworkStat1x ∗ pNetworkStat1x
- nas_NetworkStatEVDO ∗ pNetworkStatEVDO
- nas_DeviceConfigDetail ∗ pDeviceConfigDetail
- nas_DataStatusDetail ∗ pDataStatusDetail

8.1132.1 Detailed Description

Structure for SwiNetworkDebug unpack.

Parameters

pObjectVer	• Object's version number for the host to handle – 0xFF - NA – Others - shows in decimal
pNetworkStat1x	• See nas_NetworkStat1x for more information
pNetworkStatEVDO	• See nas_NetworkStatEVDO for more information.
pDeviceConfigDetail	• See nas_DeviceConfigDetail for more information.
pDataStatusDetail	• See nas_DataStatusDetail for more information.

8.1132.2 Field Documentation

8.1132.2.1 nas_DataStatusDetail∗ unpack_nas_SLQSSwiNetworkDebug_t::pDataStatusDetail

8.1132.2.2 nas_DeviceConfigDetail∗ unpack_nas_SLQSSwiNetworkDebug_t::pDeviceConfigDetail

8.1132.2.3 nas_NetworkStat1x∗ unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStat1x

8.1132.2.4 nas_NetworkStatEVDO∗ unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStatEVDO

8.1132.2.5 uint8_t* unpack_nas_SLQSSwiNetworkDebug_t::pObjectVer

8.1133 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

8.1133.1 Detailed Description

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. • 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. • 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. – 0x00 - Voice is not supported – 0x01 - Voice is supported
pGSMCipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
pWCDMACipherDomainSysInfo	• Ciphering on the service domain. – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched

pSysInfoNo-Change	• System Info No Change. • Flag used to notify clients that a request to select a network ended with no change in the PLMN. – 0x01 - No change in system information

8.1133.2 Field Documentation

8.1133.2.1 nas_AddCDMASysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo

8.1133.2.2 nas_AddSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSMSysInfo

8.1133.2.3 uint16_t* unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDRSysInfo

8.1133.2.4 uint16_t* unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTESysInfo

8.1133.2.5 nas_AddSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo

8.1133.2.6 nas_SrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo

8.1133.2.7 nas_CDMASysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo

8.1133.2.8 nas_CallBarringSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo

8.1133.2.9 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo

8.1133.2.10 nas_GSMSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo

8.1133.2.11 nas_GSMSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo

8.1133.2.12 nas_SrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSrvStatusInfo

8.1133.2.13 nas_HDRSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSysInfo

8.1133.2.14 nas_GSMSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo

8.1133.2.15 nas_LTESysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo

8.1133.2.16 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo

8.1133.2.17 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange

8.1133.2.18 nas_CallBarringSysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo

8.1133.2.19 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo

8.1133.2.20 nas_GSMSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo

8.1133.2.21 nas_WCDMASysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo

8.1134 unpack_omaDmConfigTlv_t Struct Reference

Data Fields

- uint8_t state
- uint8_t userInputReq
- uint16_t userInputTimeout
- uint16_t alertmsglength
- uint8_t alertmsg [256]

8.1134.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

Parameters

state	• 0x01 - OMA-DM Read Request • 0x02 - OMA-DM Change Request • 0x03 - OMA-DM Config Complete
user_input_req	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject
user_input_-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)

8.1134.2 Field Documentation

8.1134.2.1 uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]

8.1134.2.2 uint16_t unpack_omaDmConfigTlv_t::alertmsglength

8.1134.2.3 uint8_t unpack_omaDmConfigTlv_t::state

8.1134.2.4 uint8_t unpack_omaDmConfigTlv_t::userInputReq

8.1134.2.5 uint16_t unpack_omaDmConfigTlv_t::userInputTimeout

8.1135 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

8.1135.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

Parameters

state	• 0x01 - No Firmware available • 0x02 - Query Firmware Download • 0x03 - Firmware Downloading • 0x04 - Firmware downloaded • 0x05 - Query Firmware Update • 0x06 - Firmware updating • 0x07 - Firmware updated
user_input_req	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject
user_input_timeout	• Timeout for user input in minutes. A value of 0 means no time-out
fw_dload_size	• The size (in bytes) of the firmware update package
fw_dload_complete	• The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment.
update_complete_status	• See table below.
severity	• 0x01 - Mandatory • 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 • 0x01 - Network initiated

8.1135.2 Field Documentation

8.1135.2.1 uint8_t unpack_omaDmFotaTlv_t::description[256]

8.1135.2.2 uint16_t unpack_omaDmFotaTlv_t::descriptionlength

8.1135.2.3 uint32_t unpack_omaDmFotaTlv_t::fwdloadsize

8.1135.2.4 uint32_t unpack_omaDmFotaTlv_t::fwloadComplete

8.1135.2.5 uint16_t unpack_omaDmFotaTlv_t::namelength

8.1135.2.6 uint8_t unpack_omaDmFotaTlv_t::package_name[256]

8.1135.2.7 uint8_t unpack_omaDmFotaTlv_t::sessionType

8.1135.2.8 uint8_t unpack_omaDmFotaTlv_t::severity

8.1135.2.9 uint8_t unpack_omaDmFotaTlv_t::state

8.1135.2.10 uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus

8.1135.2.11 uint8_t unpack_omaDmFotaTlv_t::userInputReq

8.1135.2.12 uint16_t unpack_omaDmFotaTlv_t::userInputTimeout

8.1135.2.13 uint8_t unpack_omaDmFotaTlv_t::version[256]

8.1135.2.14 uint16_t unpack_omaDmFotaTlv_t::versionlength

8.1136 unpack_omaDmNotificationsTlv_t Struct Reference

Data Fields

- uint8_t notification
- uint16_t sessionStatus

8.1136.1 Field Documentation

8.1136.1.1 uint8_t unpack_omaDmNotificationsTlv_t::notification

8.1136.1.2 uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus

8.1137 unpack_pds_GetPDSDefaults_t Struct Reference

Data Fields

- uint32_t * pOperation
- uint8_t * pTimeout
- uint32_t * pInterval
- uint32_t * pAccuracy

8.1137.1 Detailed Description

Structure contain session configuration parameters.

Parameters

pOperation	• Current session operating mode – 0 - Standalone – 1 - MS based – 2 - MS assisted
pTimeout	• Maximum amount of time (seconds) to work on each fix, maximum is 255
pInterval	• Interval (seconds) between fix requests
pAccuracy	• Preferred accuracy threshold (meters)

8.1137.2 Field Documentation

8.1137.2.1 uint32_t* unpack_pds_GetPDSDefaults_t::pAccuracy

8.1137.2.2 uint32_t* unpack_pds_GetPDSDefaults_t::pInterval

8.1137.2.3 uint32_t* unpack_pds_GetPDSDefaults_t::pOperation

8.1137.2.4 uint8_t* unpack_pds_GetPDSDefaults_t::pTimeout

8.1138 unpack_pds_GetPDSState_t Struct Reference

Data Fields

- uint32_t * pEnabledStatus
- uint32_t * pTrackingStatus

8.1138.1 Detailed Description

Structure contain parameters of current PDS state.

Parameters

pEnabledStatus	• Current PDS state – 0 - disable – 1 - enable
pTrackingStatus	• Current PDS tracking session state • Values: – 0x00 - Unknown – 0x01 - Inactive – 0x02 - Active

8.1138.2 Field Documentation

8.1138.2.1 uint32_t* unpack_pds_GetPDSState_t::pEnabledStatus

8.1138.2.2 uint32_t* unpack_pds_GetPDSState_t::pTrackingStatus

8.1139 unpack_pds_GetPortAutomaticTracking_t Struct Reference

Data Fields

- uint32_t * pbAuto

8.1139.1 Detailed Description

Structure contain the parameter of automatic tracking configuration for the NMEA COM port.

Parameters

pbAuto	• Automatic tracking enabled for NMEA COM port – 0x00 - Disabled – 0x01 - Enabled

8.1139.2 Field Documentation

8.1139.2.1 uint32_t* unpack_pds_GetPortAutomaticTracking_t::pbAuto

8.1140 unpack_qmi_t Struct Reference

Data Fields

- enum msgtype type

- uint16_t msgid
- uint16_t xid

8.1140.1 Detailed Description

qmi response context

Parameters

out	*type*	message type
out	*msgid*	message id
out	*xid*	transaction id

8.1140.2 Field Documentation

8.1140.2.1 uint16_t unpack_qmi_t::msgid

8.1140.2.2 enum msgtype unpack_qmi_t::type

8.1140.2.3 uint16_t unpack_qmi_t::xid

8.1141 unpack_qos_dataRate_t Struct Reference

Data Fields

- uint32_t dataRateMax
- uint32_t guaranteedRate

8.1141.1 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)

8.1141.2 Field Documentation

8.1141.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

8.1141.2.2 uint32_t unpack_qos_dataRate_t::guaranteedRate

8.1142 unpack_qos_IPv4Addr_t Struct Reference

Data Fields

- uint32_t addr
- uint32_t subnetMask

8.1142.1 Detailed Description

This structure contains the IPv4 filter address

Parameters

addr	IPv4 address
subnetMask	A packet matches if: • (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

8.1142.2 Field Documentation

8.1142.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

8.1142.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

8.1143 unpack_qos_IPv6Addr_t Struct Reference

Data Fields

- uint8_t addr [16]
- uint8_t prefixLen

8.1143.1 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.

8.1143.2 Field Documentation

8.1143.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

8.1143.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

8.1144 unpack_qos_IPv6TrafCls_t Struct Reference

Data Fields

- uint8_t val
- uint8_t mask

8.1144.1 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: • IPv6_filter_tc_val = 00101000 • IPv6_filter_tc_mask = 11111100 Filter will compare only the first 6 bits in IPv6_filter_traffic_class with the first 6 bits in the traffic class field of the IP packet; first 6 bits in the traffic class field of the IP packet must be 001010 to match filter; last 2 bits can be anything, since they are ignored by filtering

8.1144.2 Field Documentation

8.1144.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

8.1144.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

8.1145 unpack_qos_pktErrRate_t Struct Reference

Data Fields

- uint16_t multiplier
- uint16_t exponent

8.1145.1 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)

8.1145.2 Field Documentation

8.1145.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

8.1145.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

8.1146 unpack_qos_Port_t Struct Reference

Data Fields

- uint16_t port
- uint16_t range

8.1146.1 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

8.1146.2 Field Documentation

8.1146.2.1 uint16_t unpack_qos_Port_t::port

8.1146.2.2 uint16_t unpack_qos_Port_t::range

8.1147 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

8.1147.1 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 • 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 • See LITEQMI_MAX_QOS_FILTERS for more information
BearerID	• The bearer ID • Bearer ID or Radio Link Protocol (RLP) ID of the activated flow. • Valid Values - 0 to 16 • 0xFF - Invalid value.

8.1147.2 Field Documentation

8.1147.2.1 uint8_t unpack_qos_QosFlowInfo_t::BearerID

8.1147.2.2 uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available

8.1147.2.3 uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available

8.1147.2.4 uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters

8.1147.2.5 uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters

8.1147.2.6 unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState

8.1147.2.7 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]

8.1147.2.8 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted

8.1147.2.9 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]

8.1147.2.10 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted

8.1148 unpack_qos_QosFlowInfoState_t Struct Reference

Data Fields

- uint32_t id
- uint8_t isNewFlow
- uint8_t state

8.1148.1 Detailed Description

This structure contains QoS flow state

Parameters

id	QoS identifier
isNewFlow	• 1 – Newly added flow • 0 – Existing flow
state	This indicates that the flow that was added/modified/deleted: • 0x01 – Flow activated • 0x02 – Flow modified • 0x03 – Flow deleted • 0x04 – Flow suspended • 0x05 – Flow enabled • 0x06 – Flow disabled

8.1148.2 Field Documentation

8.1148.2.1 uint32_t unpack_qos_QosFlowInfoState_t::id

8.1148.2.2 uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow

8.1148.2.3 uint8_t unpack_qos_QosFlowInfoState_t::state

8.1149 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference

Data Fields

- uint8_t NWQoSStatus

8.1149.1 Detailed Description

Structure that contains the response to get NW QoS status command

Parameters

NWQoSStatus	Network QoS support status • 0 – No QoS support in network • 1 – Network supports QoS

8.1149.2 Field Documentation

8.1149.2.1 uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus

8.1150 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

8.1150.1 Detailed Description

Structure that contains extra APN parameters

Parameters

apnId	• APN id • ID identifing the APN that the client would like to query the AMBR params
ambr_ul	• APN AMBR uplink • APN AMBR uplink values from 1 kbps to 8640 kbps
ambr_dl	• APN AMBR downlink • APN AMBR downlink values from 1 kbps to 8640 kbps
ambr_ul_ext	• Extended APN AMBR uplink • APN AMBR uplink values from 8700 kbps to 256 Mbps
ambr_dl_ext	• Extended APN AMBR downlink • APN AMBR downlink values from 8700 kbps to 256 Mbps
ambr_ul_ext2	• Second extended APN AMBR uplink • APN AMBR uplink values from 256 Mbps to 65280 Mbps
ambr_dl_ext2	• Second extended APN AMBR downlink • APN AMBR downlink values from 256 Mbps to 65280 Mbps

8.1150.2 Field Documentation

8.1150.2.1 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl

8.1150.2.2 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext

8.1150.2.3 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext2

8.1150.2.4 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul

8.1150.2.5 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext

8.1150.2.6 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext2

8.1150.2.7 uint32_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.1151 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]

8.1151.1 Detailed Description

Structure that contains APN data statistics

Parameters

apnId	• APN id • 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 • See unpack_QosFlowStat_t for more information • See LITEQMI_MAX_QOS_FLOW_PER_APN_STATS for more information

8.1151.2 Field Documentation

8.1151.2.1 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId

8.1151.2.2 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow

8.1151.2.3 unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]

8.1151.2.4 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes

8.1151.2.5 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt

8.1151.2.6 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes

8.1151.2.7 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp

8.1151.2.8 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt

8.1151.2.9 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp

8.1152 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

Data Fields

- uint8_t NumFlows
- unpack_qos_QosFlowInfo_t QosFlowInfo [8]

8.1152.1 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 • See LITEQMI_MAX_QOS_FLOWS for more information

8.1152.2 Field Documentation

8.1152.2.1 uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows

8.1152.2.2 unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]

8.1153 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference

Data Fields

- uint8_t status

8.1153.1 Detailed Description

Structure with network's QoS status

Parameters

status	Network QoS support status • 0x00 – Current network does not support QoS • 0x01 – Current network supports QoS

Note

- Technology Supported: CDMA

8.1153.2 Field Documentation

8.1153.2.1 uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status

8.1154 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference

Data Fields

- uint16_t event

8.1154.1 Detailed Description

Structure with QoS primary flow events

Parameters

event	Event which causes this indication: • 0x0001 – Primary flow QoS modify operation success • 0x0002 – Primary flow QoS modify operation failure

8.1154.2 Field Documentation

8.1154.2.1 uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event

8.1155 unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference

Data Fields

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

8.1155.1 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: • 0x01 – QMI_QOS_STATUS_ACTIVATED • 0x02 – QMI_QOS_STATUS_SUSPENDED • 0x03 – QMI_QOS_STATUS_GONE
event	• 0x01 – QMI_QOS_ACTIVATED_EV • 0x02 – QMI_QOS_SUSPENDED_EV • 0x03 – QMI_QOS_GONE_EV • 0x04 – QMI_QOS_MODIFY_ACCEPTED_EV • 0x05 – QMI_QOS_MODIFY_REJECTED_EV • 0x06 – QMI_QOS_INFO_CODE_UPDATED_EV
reason	• 0x01 - QMI_QOS_INVALID_PARAMS • 0x02 - QMI_QOS_INTERNAL_CALL_ENDED • 0x03 - QMI_QOS_INTERNAL_ERROR • 0x04 - QMI_QOS_INSUFFICIENT_LOCAL_Resources • 0x05 - QMI_QOS_TIMED_OUT_OPERATION • 0x06 - QMI_QOS_INTERNAL_UNKNOWN_CAUSE_CODE • 0x07 - QMI_QOS_INTERNAL_MODIFY_IN_PROGRESS • 0x08 - QMI_QOS_NOT_SUPPORTED • 0x09 - QMI_QOS_NOT_AVAILABLE • 0x0A - QMI_QOS_NOT_GUARANTEED • 0x0B - QMI_QOS_INSUFFICIENT_NETWORK_RESOURCES • 0x0C - QMI_QOS_AWARE_SYSTEM • 0x0D - QMI_QOS_UNAWARE_SYSTEM • 0x0E - QOS_REJECTED_OPERATION • 0x0F - QMI_QOS_WILL_GRANT_WHEN_QOS_RESUMED • 0x10 - QMI_QOS_NETWORK_CALL_ENDED • 0x11 - QMI_QOS_NETWORK_SERVICE_NOT_AVAILABLE • 0x12 - QMI_QOS_NETWORK_L2_LINK_RELEASED • 0x13 - QMI_QOS_NETWORK_L2_LINK_REESTAB_REJ • 0x14 - QMI_QOS_NETWORK_L2_LINK_REESTAB_IND • 0x15 - QMI_QOS_NETWORK_UNKNOWN_CAUSE_CODE • 0x16 - QMI_NETWORK_BUSY

8.1155.2 Field Documentation

8.1155.2.1 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event

8.1155.2.2 uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id

8.1155.2.3 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason

8.1155.2.4 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status

8.1156 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

8.1156.1 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 • 0x04 – IPv4 • 0x06 – Ipv6
IPv4SrcAddr	IPv4 filter soruce address See unpack_qos_IPv4Addr_t for more information • Implemented only for unsolicited indication
IPv4DstAddr	IPv4 filter destination address See unpack_qos_IPv4Addr_t for more information • 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: • 0x01 = ICMP • 0x06 = TCP • 0x11 = UDP • 0x32 = ESP Note: The next header protocol field will be set to 0xFD (TCP & UDP) if a TFT is received specifying a source or destination port number, but IP next header type is not specified.
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 • Implemented only for unsolicited indication
IPv6DstAddr	IPv6 filter destination address See unpack_qos_IPv6Addr_t for more information • 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) • Implemented only for unsolicited indication
TCPSrcPort	TCP filter source port filter See unpack_qos_Port_t for more information • Implemented only for unsolicited indication
TCPDstPort	TCP filter destination port filter See unpack_qos_Port_t for more information • Implemented only for unsolicited indication
UDPSrcPort	UDP filter source port filter See unpack_qos_Port_t for more information • Implemented only for unsolicited indication
UDPDstPort	UDP filter destination port filter See unpack_qos_Port_t for more information • 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 • 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 • Implemented only for unsolicited indication
UDPDstPort	Transport protocol filter destination port See unpack_qos_Port_t for more information • Implemented only for unsolicited indication

8.1156.2 Field Documentation

8.1156.2.1 uint32_t unpack_qos_swiQosFilter_t::EspSpi

8.1156.2.2 uint16_t unpack_qos_swiQosFilter_t::Id

8.1156.2.3 uint8_t unpack_qos_swiQosFilter_t::index

8.1156.2.4 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr

8.1156.2.5 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr

8.1156.2.6 unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos

8.1156.2.7 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr

8.1156.2.8 uint32_t unpack_qos_swiQosFilter_t::IPv6Label

8.1156.2.9 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr

8.1156.2.10 unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls

8.1156.2.11 uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available

8.1156.2.12 uint8_t unpack_qos_swiQosFilter_t::is_Id_Available

8.1156.2.13 uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available

8.1156.2.14 uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available

8.1156.2.15 uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available

8.1156.2.16 uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available

8.1156.2.17 uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available

8.1156.2.18 uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available

8.1156.2.19 uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available

8.1156.2.20 uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available

8.1156.2.21 uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available

8.1156.2.22 uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available

8.1156.2.23 uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available

8.1156.2.24 uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available

8.1156.2.25 uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available

8.1156.2.26 uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available

8.1156.2.27 uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available

8.1156.2.28 uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto

8.1156.2.29 uint16_t unpack_qos_swiQosFilter_t::Precedence

8.1156.2.30 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort

8.1156.2.31 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort

8.1156.2.32 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort

8.1156.2.33 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort

8.1156.2.34 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort

8.1156.2.35 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort

8.1156.2.36 uint8_t unpack_qos_swiQosFilter_t::version

8.1157 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

8.1157.1 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 • IP flow index • Integer that uniquely identifies each flow instance • Unique index must be assigned by the control point to every flow_spec instance
ProfileId3GPP2	• IP flow 3GPP2 profile ID • A profile ID is shorthand for a defined set of QoS flow parameters specified by the network; to be present while requesting QoS for a CDMA device
val_3GPP2Pri	• IP flow 3GPP2 flow priority • Flow priority used by the network in case of contention between flows with same QoS; this parameter applies for CDMA devices
TrafficClass	• IP flow traffic class • Integer that designates the requested traffic class: • 0 – Conversational • 1 – Streaming • 2 – Interactive • 3 – Background
DataRate	• IP flow data rate min max • See unpack_qos_dataRate_t for more information
TokenBucket	• IP flow data rate token bucket • See unpack_qos_tokenBucket_t for more information
Latency	• IP flow latency • Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link

Jitter	• IP flow jitter • Difference between the maximum and minimum latency (in milliseconds) that can be tolerated by an IP packet during the transfer through the wireless link
PktErrRate	• IP flow packet error rate • See unpack_qos_pktErrRate_t for more information
MinPolicedPktSz	• IP flow minimum policed packet size • Integer that defines the minimum packet size (in bytes) that will be policed for QoS guarantees; any IP packets that are smaller than the minimum specified policed size may not receive requested QoS
MaxAllowedPktSz	• IP flow maximum allowed packet size • Integer that defines the maximum packet size (in bytes) allowed in the IP flow; any IP packets greater in size than the maximum allowed packet size are not queued for transmission
val_3GPPResResidualBER	• IP flow 3GPP residual bit error rate • residual_bit_error_rate • 0 = 5∗10-2 residual BER • 1 = 1∗10-2 residual BER • 2 = 5∗10-3 residual BER • 3 = 4∗10-3 residual BER • 4 = 1∗10-3 residual BER • 5 = 1∗10-4 residual BER • 6 = 1∗10-5 residual BER • 7 = 1∗10-6 residual BER • 8 = 6∗10-8 residual BER • Integer that indicates the undetected BER for each IP flow in the delivered packets; Tapplies only to 3GPP networks
val_3GPPTraHdlPri	• 3GPP traffic handling priority • 0 – Relative traffic handling priority 1 • 1 – Relative traffic handling priority 2 • 2 – Relative traffic handling priority 3 • Defines the relative priority of the flow; applies only to 3GPP networks
val_3GPPImCn	• IP flow 3GPP IM CN flag • IM CN subsystem signaling flag: • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks

val_3GPPSigInd	• IP flow 3GPP signaling indication • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks
LteQci	• LTE QoS Class Identifier • QoS Class Identifier(QCI) is a required parameter to request QoS in LTE • QCI values: – QCI value 0 requests the network to assign the appropriate QCI value – QCI values 1-4 are associated with guaranteed bitrates – QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored

8.1157.2 Field Documentation

8.1157.2.1 unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate

8.1157.2.2 uint8_t unpack_qos_swiQosFlow_t::index

8.1157.2.3 uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available

8.1157.2.4 uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available

8.1157.2.5 uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available

8.1157.2.6 uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available

8.1157.2.7 uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available

8.1157.2.8 uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available

8.1157.2.9 uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available

8.1157.2.10 uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available

8.1157.2.11 uint8_t unpack_qos_swiQosFlow_t::is_TokenBucket_Available

8.1157.2.12 uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available

8.1157.2.13 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available

8.1157.2.14 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available

8.1157.2.15 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available

8.1157.2.16 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available

8.1157.2.17 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available

8.1157.2.18 uint32_t unpack_qos_swiQosFlow_t::Jitter

8.1157.2.19 uint32_t unpack_qos_swiQosFlow_t::Latency

8.1157.2.20 uint8_t unpack_qos_swiQosFlow_t::LteQci

8.1157.2.21 uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz

8.1157.2.22 uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz

8.1157.2.23 unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate

8.1157.2.24 uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2

8.1157.2.25 unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket

8.1157.2.26 uint8_t unpack_qos_swiQosFlow_t::TrafficClass

8.1157.2.27 uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri

8.1157.2.28 uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn

8.1157.2.29 uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER

8.1157.2.30 uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd

8.1157.2.31 uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri

8.1158 unpack_qos_tokenBucket_t Struct Reference

Data Fields

- uint32_t peakRate
- uint32_t tokenRate
- uint32_t bucketSz

8.1158.1 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

8.1158.2 Field Documentation

8.1158.2.1 uint32_t unpack_qos_tokenBucket_t::bucketSz

8.1158.2.2 uint32_t unpack_qos_tokenBucket_t::peakRate

8.1158.2.3 uint32_t unpack_qos_tokenBucket_t::tokenRate

8.1159 unpack_qos_Tos_t Struct Reference

Data Fields

- uint8_t val
- uint8_t mask

8.1159.1 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: • IPv4_filter_tos_val = 00101000 • IPv4_filter_tos_mask = 11111100 The filter will compare only the first 6 bits in the IPv4_filter_type_of_service with the first 6 bits in the TOS field of the IP packet. The first 6 bits in the TOS field of the IP packet must be 001010 to match the filter. The last 2 bits can be anything since they are ignored by filtering.

8.1159.2 Field Documentation

8.1159.2.1 uint8_t unpack_qos_Tos_t::mask

8.1159.2.2 uint8_t unpack_qos_Tos_t::val

8.1160 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

8.1160.1 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

8.1160.2 Field Documentation

8.1160.2.1 uint32_t unpack_QosFlowStat_t::bearerId

8.1160.2.2 uint64_t unpack_QosFlowStat_t::tx_bytes

8.1160.2.3 uint64_t unpack_QosFlowStat_t::tx_bytes_drp

8.1160.2.4 uint32_t unpack_QosFlowStat_t::tx_pkt

8.1160.2.5 uint32_t unpack_QosFlowStat_t::tx_pkt_drp

8.1161 unpack_rms_GetSMSWake_t Struct Reference

Data Fields

- uint32_t enabled
- uint32_t wake_mask

8.1161.1 Detailed Description

Parameters

enabled	• SMS wake functionality enabled – 0 - Disabled – 1 - Enabled
wake_mask	• SMS wake mask to search for incoming messages (only relevant when enabled)

8.1161.2 Field Documentation

8.1161.2.1 uint32_t unpack_rms_GetSMSWake_t::enabled

8.1161.2.2 uint32_t unpack_rms_GetSMSWake_t::wake_mask

8.1162 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

8.1162.1 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.

8.1162.2 Field Documentation

8.1162.2.1 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv

8.1162.2.2 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::RxOkByteCountTlv

8.1162.2.3 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxOkConutTlv

8.1162.2.4 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxDropConutTlv

8.1162.2.5 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::TxOkByteCountTlv

8.1162.2.6 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxOkConutTlv

8.1163 unpack_sar_SLQSGetRfSarState_t Struct Reference

Data Fields

- uint32_t * pSarRFState

8.1163.1 Detailed Description

Parameters

pSarRFState	• SAR RF State – 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

8.1163.2 Field Documentation

8.1163.2.1 uint32_t* unpack_sar_SLQSGetRfSarState_t::pSarRFState

8.1164 unpack_sms_GetSMSCAddress_t Struct Reference

Data Fields

- uint8_t addressSize
- uint8_t * pSMSCAddress
- uint8_t typeSize
- uint8_t * pSMSCType

8.1164.1 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.

8.1164.2 Field Documentation

8.1164.2.1 uint8_t unpack_sms_GetSMSCAddress_t::addressSize

8.1164.2.2 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCAddress

8.1164.2.3 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCType

8.1164.2.4 uint8_t unpack_sms_GetSMSCAddress_t::typeSize

8.1165 unpack_sms_SaveSMS_t Struct Reference

Data Fields

- uint32_t * pMessageIndex

8.1165.1 Detailed Description

Parameters

pMessageIndex	• The message index assigned by the device

8.1165.2 Field Documentation

8.1165.2.1 uint32_t* unpack_sms_SaveSMS_t::pMessageIndex

8.1166 unpack_sms_SendSMS_t Struct Reference

Data Fields

- uint16_t messageID
- uint32_t messageFailureCode

8.1166.1 Detailed Description

Parameters

messageID	• WMS message ID
messageFailure-Code	• pointer to message failure code. If cause code is not provided, then value will be 0xFFFFFFFF

8.1166.2 Field Documentation

8.1166.2.1 uint32_t unpack_sms_SendSMS_t::messageFailureCode

8.1166.2.2 uint16_t unpack_sms_SendSMS_t::messageID

8.1167 unpack_sms_SetNewSMSCallback_ind_t Struct Reference

Data Fields

- struct newMTMessageTlv NewMMTlv
- struct transferRouteMessageTlv TRMessageTlv
- struct messageModeTlv MMTlv
- struct sMSEtwsMessageTlv ETWSTlv
- struct eTWSPLMNInfoTlv ETWSPLMNTlv
- struct sMSCAddressTlv SMSCTlv
- struct sMSOnIMSTlv IMSTlv

8.1167.1 Detailed Description

Parameters

NewMMTlv	• MT message
TRMessageTlv	• Transfer Route MT Message • See transferRouteMessageTlv for more information
MMTlv	• Message mode • See messageModeTlv for more information
ETWSTlv	• ETWS Message • See sMSEtwsMessageTlv for more information
ETWSPLMNTlv	• ETWS PLMN Information • See eTWSPLMNInfoTlv for more information
SMSCTlv	• SMSC Address • See sMSCAddressTlv for more information
IMSTlv	• SMS on IMS • See sMSOnIMSTlv for more information

8.1167.2 Field Documentation

8.1167.2.1 struct eTWSPLMNInfoTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv

8.1167.2.2 struct sMSEtwsMessageTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv

8.1167.2.3 struct sMSOnIMSTlv unpack_sms_SetNewSMSCallback_ind_t::IMSTlv

8.1167.2.4 struct messageModeTlv unpack_sms_SetNewSMSCallback_ind_t::MMTlv

8.1167.2.5 struct newMTMessageTlv unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv

8.1167.2.6 struct sMSCAddressTlv unpack_sms_SetNewSMSCallback_ind_t::SMSCTlv

8.1167.2.7 struct transferRouteMessageTlv unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv

8.1168 unpack_sms_SetNewSMSCallback_t Struct Reference

8.1169 unpack_sms_SLQSDeleteSMS_t Struct Reference

8.1170 unpack_sms_SLQSGetIndicationRegister_t Struct Reference

Data Fields

- sms_getIndicationReg * pGetIndicationRegInfo

8.1170.1 Detailed Description

Structure containing Parameters that provides registration state of different WMS indications.

Parameters

pGetIndication-RegInfo	• Pointer to structure of getIndicationReg – See sms_getIndicationReg for more information

8.1170.2 Field Documentation

8.1170.2.1 sms_getIndicationReg* unpack_sms_SLQSGetIndicationRegister_t::pGetIndicationRegInfo

8.1171 unpack_sms_SLQSGetMessageWaiting_t Struct Reference

Data Fields

- sms_getMsgWaitingInfo * pGetMsgWaitingInfoResp

8.1171.1 Detailed Description

Structure contain Parameter that provide information about the message waiting information.

Parameters

pGetMsg-WaitingInfoResp	• Pointer to structure of getMsgWaitingInfoResp – See sms_getMsgWaitingInfoResp for more information

8.1171.2 Field Documentation

8.1171.2.1 sms_getMsgWaitingInfo* unpack_sms_SLQSGetMessageWaiting_t::pGetMsgWaitingInfoResp

8.1172 unpack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t messageTag
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t message [2048]

8.1172.1 Detailed Description

Parameters

messageTag	• Message tag – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent
messageFormat	• Message format – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 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

8.1172.2 Field Documentation

8.1172.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[2048]

8.1172.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

8.1172.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

8.1172.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

8.1173 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

Data Fields

- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig

- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

8.1173.1 Detailed Description

Structure contain parameters that Provides Information about the SMS BroadcastConfiguration

Parameters

pBroadcast-Config	• The data for 3GPP Broadcast Information(Optional).
pCDMA-BroadcastConfig	• The data for 3GPP2 Broadcast Information(Optional).

8.1173.2 Field Documentation

8.1173.2.1 sms_qaQmi3GPPBroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pBroadcastConfig

8.1173.2.2 sms_qaQmi3GPP2BroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pCDMABroadcast-Config

8.1174 unpack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t messageListSize
- qmiSmsMessageList messageList [255]

8.1174.1 Detailed Description

Parameters

messageListSize	• Upon input the maximum number of elements that the message list array can contain. • Upon successful output the actual number of elements in the message list array.
messageList	• Message List • See qmiSmsMessageList for more information

8.1174.2 Field Documentation

8.1174.2.1 qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]

8.1174.2.2 uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize

8.1175 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference

Data Fields

- sms_getTransLayerInfo * pGetTransLayerInfo

8.1175.1 Detailed Description

Structure contain parameters that gives information about the transport layer.

Parameters

pGetTransLayer-Info	• Pointer to structure of getTransLayerInfo – See sms_getTransLayerInfo for more information

8.1175.2 Field Documentation

8.1175.2.1 sms_getTransLayerInfo* unpack_sms_SLQSGetTransLayerInfo_t::pGetTransLayerInfo

8.1176 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference

Data Fields

- sms_getTransNWRegInfo * pGetTransNWRegInfo

8.1176.1 Detailed Description

Structure containing parameters that provides transport layer network registration info.

Parameters

pGetTransNW-RegInfo	• Pointer to structure of getTransNWRegInfo – See sms_getTransNWRegInfo for more information

8.1176.2 Field Documentation

8.1176.2.1 sms_getTransNWRegInfo* unpack_sms_SLQSGetTransNWRegInfo_t::pGetTransNWRegInfo

8.1177 unpack_sms_SLQSModifySMSStatus_t Struct Reference

8.1178 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

Data Fields

- sms_maxStorageSizeResp * pMaxStorageSizeResp

8.1178.1 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 – See sms_maxStorageSizeResp for more information

8.1178.2 Field Documentation

8.1178.2.1 sms_maxStorageSizeResp∗ unpack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeResp

8.1179 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference

Data Fields

- sms_msgProtocolResp ∗ pMessageProtocol

8.1179.1 Detailed Description

Structure contain Parameters that get the message protocol currently in use for the WMS client.

Parameters

pMessage-Protocol	• Pointer to smsMsgprotocolResp – See sms_msgProtocolResp for more information

8.1179.2 Field Documentation

8.1179.2.1 sms_msgProtocolResp∗ unpack_sms_SLQSSmsGetMessageProtocol_t::pMessageProtocol

8.1180 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference

Data Fields

- uint32_t ∗ pSmsStorage

8.1180.1 Detailed Description

Structure contain Parameters that return current SMS configuration that is applied to all incoming and outgoing messages.

Parameters

pSmsStorage	• Values: – 0x01 - device's permanent memory – 0x02 - UICC

8.1180.2 Field Documentation

8.1180.2.1 uint32_t* unpack_sms_SLQSSwiGetSMSStorage_t::pSmsStorage

8.1181 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference

Data Fields

- uint8_t storageType
- uint8_t messageMode

8.1181.1 Detailed Description

Parameters

storageType	• SMS message storage type – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
messageMode	• 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)

8.1181.2 Field Documentation

8.1181.2.1 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode

8.1181.2.2 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType

8.1182 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

8.1182.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

Parameters

function	• Setting to indicate when modem should start an automatic GNSS fix – 0 - disabled – 1 - At bootup – 2 - When NMEA port is opened
function_reported	• 0 - not reported by modem • l - reported by modem
fix_type	• Type of GNSS fix: – 1 - Default Engine mode – 2 - MS-Based – 3 - MS-Assisted – 4 - Standalone
fix_type_reported	• 0 - not reported by modem • l - reported by modem
max_time	• Maximum time allowed for the receiver to get a fix in seconds • Valid range: 1-255
max_time_reported	• 0 - not reported by modem • l - reported by modem
max_dist	• Maximum uncertainty of a fix measured by distance in meters • Valid range: 1 - 4294967280
max_dist_reported	• 0 - not reported by modem • l - reported by modem
fix_rate	• Time between fixes in seconds • Valid range: 1–65535
fix_rate_reported	• 0 - not reported by modem • l - reported by modem

8.1182.2 Field Documentation

8.1182.2.1 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate

8.1182.2.2 int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported

8.1182.2.3 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type

8.1182.2.4 int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported

8.1182.2.5 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function

8.1182.2.6 int unpack_swiloc_SwiLocGetAutoStart_t::function_reported

8.1182.2.7 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist

8.1182.2.8 int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported

8.1182.2.9 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time

8.1182.2.10 int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported

8.1183 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference

Data Fields

- uint32_t eventType
- unpack_omaDmFotaTlv_t SessionInfoFota
- unpack_omaDmConfigTlv_t SessionInfoConfig
- unpack_omaDmNotificationsTlv_t SessionInfoNotification

8.1183.1 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 • 0x01 - SWIOMA-DM Config • 0x02 - SWIOMA-DM Notification
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

8.1183.2 Field Documentation

8.1183.2.1 uint32_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::eventType

8.1183.2.2 unpack_omaDmConfigTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoConfig

8.1183.2.3 unpack_omaDmFotaTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoFota

8.1183.2.4 unpack_omaDmNotificationsTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfo-Notification

8.1184 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

8.1184.1 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 – 0x01 - No Firmware available – 0x02 - Query Firmware Download – 0x03 - Firmware Downloading – 0x04 - Firmware Downloaded – 0x05 - Query Firmware Update – 0x06 - Firmware Updating – 0x07 - Firmware Updated
Update-CompleteStatus	• 2 byte parameter indicating Update Complete Status – See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h Update Complete Status

Severity	• 1 byte parameter indicating severity – 0x01 - Mandatory – 0x02 - Optional
SourceLength	• 2 byte parameter indicating Length of Vendor Name String in Bytes.
Source	• Variable length parameter indicating Vendor Name in ASCII • 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 • 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 • 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 • 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 • See LITEQMI_MAX_SWIOMA_STR_LEN for more information
SessionType	• 1 byte parameter reflects the last session started for Sprint – 0x00 - No session since boot – 0x01 - Sprint CI-DC Session – 0x02 - Sprint CI-PRL Session – 0x03 - Sprint CI-FUMO Session – 0x04 - Sprint HFA-DC Session – 0x05 - Sprint HFA-PRL Session – 0x06 - Sprint HFA-FUMO Session – 0x07 - Sprint NI Session

SessionState	• 1 byte parameter indicating session state – 0x01 - idle – 0x02 - active – 0x03 - pending
RetryCount	• 1 byte parameter indicating retries left count – valid values 0 to 6

8.1184.2 Field Documentation

8.1184.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

8.1184.2.2 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength

8.1184.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength

8.1184.2.4 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]

8.1184.2.5 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgName[255]

8.1184.2.6 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgNameLength

8.1184.2.7 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::RetryCount

8.1184.2.8 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionState

8.1184.2.9 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

8.1184.2.10 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Severity

8.1184.2.11 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Source[255]

8.1184.2.12 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SourceLength

8.1184.2.13 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Status

8.1184.2.14 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Time[255]

8.1184.2.15 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::TimeLength

8.1184.2.16 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus

8.1185 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference

Data Fields

- uint32_t OMADMEnabled
- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t Autosdm

- uint8_t FwAutoCheck

8.1185.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

Parameters

OMADMEnabled[OUT]	• Optional 4 byte parameter indicating OMADM service enabled – 0x00000001 - Client-initiated device configuration – 0x00000002 - Network-initiated device configuration – 0x00000010 - Client-initiated FUMO – 0x00000020 - Network-initiated FUMO • function SLQSOMADMGetSettings2() returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.
FOTAdownload[OUT]	• Optional 1 Byte parameter indicating support for FOTA Automatic download – 0x00 - Host permission required before downloading – 0x01 - Automatically start downloading, no host permission required – 0x02 - Automatically start downloading, while not roaming – 0x03 - Automatically reject download – 0x04 - Automatically reject download with “Enterprise Reject Policy” • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
FOTAUpdate[OUT]	• Optional 1 byte parameter indicating FOTA Automatic update – 0x00 - User permission required before updating firmware – 0x01 - No user permission required before updating firmware – 0x02 - User permission required, auto update on power up • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
Autosdm[OUT]	• Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
FwAutoCheck[OUT]	• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 0x01 - Enabled • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.

8.1185.2 Field Documentation

8.1185.2.1 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::Autosdm

8.1185.2.2 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAdownload

8.1185.2.3 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAUpdate

8.1185.2.4 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FwAutoCheck

8.1185.2.5 uint32_t unpack_swioma_SLQSOMADMGetSettings_t::OMADMEnabled

8.1186 unpack_swioma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t FwAvailability

8.1186.1 Detailed Description

Structure that contains the responses for OMA start session command

Parameters

pFwAvailability[-OUT]	• OMA-DM CHECK FW Available – 0x00000001 - FW Available. For CIDC and CIPRL, this value will be returned by the modem. CIDC and CIPRL are asynchronous OMADM sessions. – 0x00000002 - FW Not Available – 0x00000003 - FW Check Timed Out

8.1186.2 Field Documentation

8.1186.2.1 uint32_t unpack_swioma_SLQSOMADMStartSession_t::FwAvailability

8.1187 unpack_uim_ChangePin_t Struct Reference

Data Fields

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

8.1187.1 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.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information.
pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • 0xFFFFFFFF, if unavailable

Note

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

8.1187.2 Field Documentation

8.1187.2.1 uim_encryptedPIN1∗ unpack_uim_ChangePin_t::pEncryptedPIN1

8.1187.2.2 uint32_t∗ unpack_uim_ChangePin_t::pIndicationToken

8.1187.2.3 uim_remainingRetries∗ unpack_uim_ChangePin_t::pRemainingRetries

8.1187.2.4 uint16_t unpack_uim_ChangePin_t::Tlvresult

8.1188 unpack_uim_GetCardStatus_t Struct Reference

Data Fields

- uim_cardStatus ∗ pCardStatus
- uim_hotSwapStatus ∗ pHotSwapStatus
- uint16_t Tlvresult

8.1188.1 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.
pHotSwap-Status(optional)	• See uim_hotSwapStatus for more information.

Note

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

8.1188.2 Field Documentation

8.1188.2.1 uim_cardStatus* unpack_uim_GetCardStatus_t::pCardStatus

8.1188.2.2 uim_hotSwapStatus* unpack_uim_GetCardStatus_t::pHotSwapStatus

8.1188.2.3 uint16_t unpack_uim_GetCardStatus_t::Tlvresult

8.1189 unpack_uim_ReadTransparent_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_readResult * pReadResult
- uint32_t * pIndicationToken
- uint8_t * pEncryptedData
- uint16_t Tlvresult

8.1189.1 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. • When this TLV is present, it indicates that the result must be provided in a subsequent indication.
pEncrypted-Data(optional)	• Encrypted Data. • Indicates whether the data from the card passed in read_result is encrypted.

Note

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

8.1189.2 Field Documentation

8.1189.2.1 uim_cardResult* unpack_uim_ReadTransparent_t::pCardResult

8.1189.2.2 uint8_t* unpack_uim_ReadTransparent_t::pEncryptedData

8.1189.2.3 uint32_t* unpack_uim_ReadTransparent_t::pIndicationToken

8.1189.2.4 uim_readResult* unpack_uim_ReadTransparent_t::pReadResult

8.1189.2.5 uint16_t unpack_uim_ReadTransparent_t::Tlvresult

8.1190 unpack_uim_SetPinProtection_t Struct Reference

Data Fields

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

8.1190.1 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.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information.
pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • 0xFFFFFFFF, if unavailable

Note

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

8.1190.2 Field Documentation

8.1190.2.1 uim_encryptedPIN1* unpack_uim_SetPinProtection_t::pEncryptedPIN1

8.1190.2.2 uint32_t* unpack_uim_SetPinProtection_t::pIndicationToken

8.1190.2.3 uim_remainingRetries* unpack_uim_SetPinProtection_t::pRemainingRetries

8.1190.2.4 uint16_t unpack_uim_SetPinProtection_t::Tlvresult

8.1191 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference

Data Fields

- slots_t slotsstatusChange
- uint8_t bNumberOfPhySlots

8.1191.1 Detailed Description

Structure consist of card status params

Parameters

slotsstatus-Change	• See slot_t for more information
bNumberOfPhy-Slots	• Number of Physical Slot(s)

8.1191.2 Field Documentation

8.1191.2.1 uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots

8.1191.2.2 slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange

8.1192 unpack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_authenticateResult * pAuthenticateResult
- uint32_t * pIndicationToken

8.1192.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMAuthenticate.

Parameters

pCard-Result(optional)	• See uim_cardResult for more information.
pAuthenticate-Result(optional)	• See uim_authenticateResult for more information.
pIndication-Token(optional)	• Response in Indication. • 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.

8.1192.2 Field Documentation

8.1192.2.1 uim_authenticateResult* unpack_uim_SLQSUIMAuthenticate_t::pAuthenticateResult

8.1192.2.2 uim_cardResult* unpack_uim_SLQSUIMAuthenticate_t::pCardResult

8.1192.2.3 uint32_t* unpack_uim_SLQSUIMAuthenticate_t::pIndicationToken

8.1193 unpack_uim_SLQSUIMDepersonalization_t Struct Reference

Data Fields

- uim_remainingRetries * pRemainingRetries

8.1193.1 Detailed Description

This structure contains information of the response parameters associated with a Depersonalization.

Parameters

pRemaining-Retries(optional)	• See uim_remainingRetries for more information.

Note

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

8.1193.2 Field Documentation

8.1193.2.1 uim_remainingRetries* unpack_uim_SLQSUIMDepersonalization_t::pRemainingRetries

8.1194 unpack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t eventMask

8.1194.1 Detailed Description

Parameters

eventMask	- bit 0 - card status • bit 1 - SAP connection • bit 4 - physical slot status

8.1194.2 Field Documentation

8.1194.2.1 uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask

8.1195 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference

Data Fields

- uint8_t * pAutoSelection
- uim_personalizationStatus * pPersonalizationStatus
- uint8_t * pHaltSubscription

8.1195.1 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. • Valid values – 0 - Automatic provisioning is off – 1 - Automatic provisioning is on
p-Personalization-Status(optional)	• See uim_personalizationStatus for more information.
pHalt-Subscription(optional)	• Indicates if the modem is configured to publish the subscription after successful initialization. • Valid values – 0 - Modem proceeds with publishing the subscription – 1 - Modem does not publish the subscription

8.1195.2 Field Documentation

8.1195.2.1 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pAutoSelection

8.1195.2.2 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pHaltSubscription

8.1195.2.3 uim_personalizationStatus* unpack_uim_SLQSUIMGetConfiguration_t::pPersonalizationStatus

8.1196 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_fileAttributes * pFileAttributes
- uint32_t * pIndicationToken

8.1196.1 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.
pFile-Attributes(optional)	• See uim_fileAttributes for more information.
pIndication-Token(optional)	• Response in Indication. • 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.

8.1196.2 Field Documentation

8.1196.2.1 uim_cardResult* unpack_uim_SLQSUIMGetFileAttributes_t::pCardResult

8.1196.2.2 uim_fileAttributes* unpack_uim_SLQSUIMGetFileAttributes_t::pFileAttributes

8.1196.2.3 uint32_t* unpack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

8.1197 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference

Data Fields

- uint8_t * pNumberOfPhySlot
- slots_t * pUimSlotsStatus

8.1197.1 Detailed Description

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

Parameters

pNumberOfPhySlot	• Number of sets of the Slot Status.
pUimSlotsStatus	• Slots Status See slots_t for more information..

8.1197.2 Field Documentation

8.1197.2.1 uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot

8.1197.2.2 slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus

8.1198 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

Data Fields

- uim_refreshevent * pRefreshEvent

8.1198.1 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.

8.1198.2 Field Documentation

8.1198.2.1 uim_refreshevent∗ unpack_uim_SLQSUIMRefreshGetLastEvent_t::pRefreshEvent

8.1199 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference

Data Fields

- uim_cardStatus ∗ pCardStatus

8.1199.1 Detailed Description

This structure contains information about Status change callback.

Parameters

pCardStatus	Card Status • See uim_cardStatus for more information.

8.1199.2 Field Documentation

8.1199.2.1 uim_cardStatus∗ unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus

8.1200 unpack_uim_UnblockPin_t Struct Reference

Data Fields

- uim_remainingRetries ∗ pRemainingRetries
- uim_encryptedPIN1 ∗ pEncryptedPIN1
- uint32_t ∗ pIndicationToken
- uint16_t Tlvresult

8.1200.1 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.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information.
pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • 0xFFFFFFFF, if unavailable

8.1200.2 Field Documentation

8.1200.2.1 uim_encryptedPIN1∗ unpack_uim_UnblockPin_t::pEncryptedPIN1

8.1200.2.2 uint32_t∗ unpack_uim_UnblockPin_t::pIndicationToken

8.1200.2.3 uim_remainingRetries∗ unpack_uim_UnblockPin_t::pRemainingRetries

8.1200.2.4 uint16_t unpack_uim_UnblockPin_t::Tlvresult

8.1201 unpack_uim_VerifyPin_t Struct Reference

Data Fields

- uim_remainingRetries ∗ pRemainingRetries
- uim_encryptedPIN1 ∗ pEncryptedPIN1
- uint32_t ∗ pIndicationToken
- uint16_t Tlvresult

8.1201.1 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.
pEncryptedPI-N1(optional)	• See uim_encryptedPIN1 for more information.
pIndication-Token(optional)	• Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • 0xFFFFFFFF, if unavailable

Note

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

8.1201.2 Field Documentation

8.1201.2.1 uim_encryptedPIN1∗ unpack_uim_VerifyPin_t::pEncryptedPIN1

8.1201.2.2 uint32_t∗ unpack_uim_VerifyPin_t::pIndicationToken

8.1201.2.3 uim_remainingRetries∗ unpack_uim_VerifyPin_t::pRemainingRetries

8.1201.2.4 uint16_t unpack_uim_VerifyPin_t::Tlvresult

8.1202 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

8.1202.1 Field Documentation

8.1202.1.1 voice_alphaIDInfo* unpack_voice_SLQSOriginateUSSD_t::pAlphaIDInfo

8.1202.1.2 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCallId

8.1202.1.3 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCcResultType

8.1202.1.4 voice_ccSUPSType* unpack_voice_SLQSOriginateUSSD_t::pCCSuppsType

8.1202.1.5 uint16_t* unpack_voice_SLQSOriginateUSSD_t::pfailureCause

8.1202.1.6 struct voice_USSInfo* unpack_voice_SLQSOriginateUSSD_t::pUSSDInfo

8.1203 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.1203.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

pCallId	• Unique call identifier for the call that must be answered.

8.1203.2 Field Documentation

8.1203.2.1 uint8_t* unpack_voice_SLQSVoiceAnswerCall_t::pCallId

8.1204 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1204.1 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. A burst DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • If the call ID value received is 0, no value has been returned by the device

8.1204.2 Field Documentation

8.1204.2.1 uint8_t* unpack_voice_SLQSVoiceBurstDTMF_t::pCallID

8.1205 unpack_voice_SLQSVoiceDialCall_t Struct Reference

Data Fields

- uint8_t * pCallID
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- voice_ccSUPSType * pCCSUPSType

8.1205.1 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 – See voice_alphaIDInfo for more information
pCCResult-Type(optional)	• Call Control Result Type. – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service
pCCSUPS-Type(optional)	• Pointer to structure of ccSUPSType • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

8.1205.2 Field Documentation

8.1205.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceDialCall_t::pAlphaIDInfo

8.1205.2.2 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCallID

8.1205.2.3 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCCResultType

8.1205.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceDialCall_t::pCCSUPSType

8.1206 unpack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.1206.1 Detailed Description

Parameters

pCallId	• Unique call identifier for the call that must be ended

8.1206.2 Field Documentation

8.1206.2.1 uint8_t* unpack_voice_SLQSVoiceEndCall_t::pCallId

8.1207 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

8.1207.1 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.
pArrRemotePartyNum(optional)	• See voice_arrRemotePartyNum for more information.
pArrRemotePartyName(optional)	• See voice_arrRemotePartyName for more information.
pArrAlertingType(optional)	• See voice_arrAlertingType for more information.
pArrUUSInfo(optional)	• See voice_arrUUSInfo for more information.
pArrSvcOption(optional)	• See voice_arrSvcOption for more information.
pOTASPStatus(optional)	• OTASP status for the OTASP call. • Applicable only for 3GPP2 devices. – 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP – 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA) – 0xFF - Not Available

pVoicePrivacy(optional)	• Voice Privacy. • Values. – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available
pArrCallEndReason(optional)	• See voice_arrCallEndReason for more information.
pArrAlphaID(optional)	• See voice_arrAlphaID for more information.
pArrConnectPartyNum(optional)	• See voice_arrConnectPartyNum for more information.
pArrDiagInfo(optional)	• See voice_arrDiagInfo for more information.
pArrCalledPartyNum(optional)	• See voice_arrCalledPartyNum for more information.
pArrRedirPartyNum(optional)	• See voice_arrRedirPartyNum for more information.
pArrAlertingPattern(optional)	• See voice_arrAlertingPattern for more information.

8.1207.2 Field Documentation

8.1207.2.1 voice_arrAlertingPattern∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingPattern

8.1207.2.2 voice_arrAlertingType∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingType

8.1207.2.3 voice_arrAlphaID∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlphaID

8.1207.2.4 voice_arrCalledPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCalledPartyNum

8.1207.2.5 voice_arrCallEndReason∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallEndReason

8.1207.2.6 voice_arrCallInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallInfo

8.1207.2.7 voice_arrConnectPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrConnectPartyNum

8.1207.2.8 voice_arrDiagInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrDiagInfo

8.1207.2.9 voice_arrRedirPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRedirPartyNum

8.1207.2.10 voice_arrRemotePartyName∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyName

8.1207.2.11 voice_arrRemotePartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyNum

8.1207.2.12 voice_arrSvcOption∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrSvcOption

8.1207.2.13 voice_arrUUSInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrUUSInfo

8.1207.2.14 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pOTASPStatus

8.1207.2.15 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pVoicePrivacy

8.1208 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

8.1208.1 Detailed Description

This structure contains Voice Get Call Barring Response Parameters

Parameters

pSvcClass	• Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.

pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1208.2 Field Documentation

8.1208.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallBarring_t::pAlphaIDInfo

8.1208.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCallID

8.1208.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCCResType

8.1208.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallBarring_t::pCCSUPSType

8.1208.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallBarring_t::pFailCause

8.1208.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

8.1209 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

8.1209.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Unpack Parameters

Parameters

pGetCallFWInfo	• Pointer to structure of getCallFWInfo (optional) – See voice_getCallFWInfo for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available

pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information
pGetCallFWExt-Info	• Pointer to structure of getCallFWExtInfo (optional) – See voice_getCallFWExtInfo for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1209.2 Field Documentation

8.1209.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pAlphaIDInfo

8.1209.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCallID

8.1209.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCResType

8.1209.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCSUPSType

8.1209.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pFailCause

8.1209.2.6 voice_getCallFWExtInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWExtInfo

8.1209.2.7 voice_getCallFWInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWInfo

8.1210 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

8.1210.1 Detailed Description

This structure contains information of the unpack parameters associated with a call.

Parameters

pCall-Info(optional)	• See voice_callInfo for more information.
pRemoteParty-Num(optional)	• See voice_remotePartyNum for more information.
pSrvOpt	• Service option(optional) • Applicable only for 3GPP2 devices. • See Table8 qaGobiApiTableServiceOptions.h for standard service option number assignments.
pVoicePrivacy	• Voice Privacy.(optional) • Applicable only for 3GPP2 devices. • Values. – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available

pOTASPStatus	• OTASP status for the OTASP call.(optional) • Applicable only for 3GPP2 devices. – 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP – 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA) – 0xFF - Not Available
pRemoteParty-Name(optional)	• Applicable only for 3GPP devices. • See voice_remotePartyName for more information.
pUUS-Info(optional)	• Applicable only for 3GPP devices. • See voice_UUSInfo for more information.
pAlert-Type(optional)	• Alerting type. • Applicable only for 3GPP devices. – 0x00 - ALERTING_LOCAL - Local – 0x01 - ALERTING_REMOTE - Remote – 0xFF - Not Available
pAlphaID-Info(optional)	• Applicable only for 3GPP devices. • See voice_alphaIDInfo for more information.
pConnectNum-Info(optional)	• See vocie_connectNumInfo for more information.

pDiag-Info(optional)	• See voice_diagInfo for more information.
pAlertingPattern	• Alerting pattern.(optional) – 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1 – 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2 – 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3 – 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5 – 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6 – 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7 – 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8 – 0x08 - QMI_VOICE_ALERTING_PATTERN_9 - Pattern 9 – 0xFF - Not Available

8.1210.2 Field Documentation

8.1210.2.1 uint32_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertingPattern

8.1210.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertType

8.1210.2.3 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pAlphaIDInfo

8.1210.2.4 voice_callInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pCallInfo

8.1210.2.5 voice_connectNumInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pConnectNumInfo

8.1210.2.6 voice_diagInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pDiagInfo

8.1210.2.7 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pOTASPStatus

8.1210.2.8 voice_remotePartyName* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyName

8.1210.2.9 voice_remotePartyNum* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyNum

8.1210.2.10 uint16_t* unpack_voice_SLQSVoiceGetCallInfo_t::pSrvOpt

8.1210.2.11 voice_UUSInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pUUSInfo

8.1210.2.12 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pVoicePrivacy

8.1211 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

8.1211.1 Detailed Description

This structure contains Voice Get Call Waiting Unpack Parameters

Parameters

pSvcClass	• Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1211.2 Field Documentation

8.1211.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallWaiting_t::pAlphaIDInfo

8.1211.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCallID

8.1211.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCResType

8.1211.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCSUPSType

8.1211.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pFailCause

8.1211.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

8.1212 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

8.1212.1 Detailed Description

This structure contains Voice Get Calling Line Identification Presentation(CLIP) Response Parameters

Parameters

pCLIPResp	• Pointer to structure of CLIPResp (optional) – See voice_CLIPResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1212.2 Field Documentation

8.1212.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIP_t::pAlphaIDInfo

8.1212.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCallID

8.1212.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCCResType

8.1212.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIP_t::pCCSUPSType

8.1212.2.5 voice_CLIPResp* unpack_voice_SLQSVoiceGetCLIP_t::pCLIPResp

8.1212.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIP_t::pFailCause

8.1213 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

8.1213.1 Detailed Description

This structure contains Voice Get Calling Line Identification Restriction (CLIR) Response Parameters

Parameters

pCLIRResp	• Pointer to structure of CLIRResp (optional) – See voice_CLIRResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available

pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1213.2 Field Documentation

8.1213.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIR_t::pAlphaIDInfo

8.1213.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCallID

8.1213.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCCResType

8.1213.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIR_t::pCCSUPSType

8.1213.2.5 voice_CLIRResp* unpack_voice_SLQSVoiceGetCLIR_t::pCLIRResp

8.1213.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIR_t::pFailCause

8.1214 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

8.1214.1 Detailed Description

This structure contains Voice Get Calling Name Presentation(CNAP) Response Parameters

Parameters

pCNAPResp	• Pointer to structure of CNAPResp (optional) – See voice_CNAPResp for more information

pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1214.2 Field Documentation

8.1214.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCNAP_t::pAlphaIDInfo

8.1214.2.2 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCallID

8.1214.2.3 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCCResType

8.1214.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCNAP_t::pCCSUPSType

8.1214.2.5 voice_CNAPResp* unpack_voice_SLQSVoiceGetCNAP_t::pCNAPResp

8.1214.2.6 uint16_t* unpack_voice_SLQSVoiceGetCNAP_t::pFailCause

8.1215 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

8.1215.1 Detailed Description

This structure contains Voice Get Connected Line Identification Presentation(COLP) Response Parameters

Parameters

pCOLPResp	• Pointer to structure of COLPResp (optional) – See voice_COLPResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1215.2 Field Documentation

8.1215.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLP_t::pAlphaIDInfo

8.1215.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCallID

8.1215.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCCResType

8.1215.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLP_t::pCCSUPSType

8.1215.2.5 voice_COLPResp* unpack_voice_SLQSVoiceGetCOLP_t::pCOLPResp

8.1215.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLP_t::pFailCause

8.1216 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

8.1216.1 Detailed Description

This structure contains Voice Get Connected Line Identification Restriction(COLR) Response Parameters

Parameters

pCOLRResp	• Pointer to structure of COLRResp (optional) – See voice_COLRResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1216.2 Field Documentation

8.1216.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLR_t::pAlphaIDInfo

8.1216.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCallID

8.1216.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCCResType

8.1216.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLR_t::pCCSUPSType

8.1216.2.5 voice_COLRResp* unpack_voice_SLQSVoiceGetCOLR_t::pCOLRResp

8.1216.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLR_t::pFailCause

8.1217 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

8.1217.1 Detailed Description

This structure contains Voice Get Configuration Response Parameters.

Parameters

pAutoAnswer-Stat(optional)	• Auto Answer Status • Value returned is read from NV_AUTO_ANSWER_I. – 0x00 - Disabled – 0x01 - Enabled – 0xFF - Not Available
pAirTimer-Cnt(optional)	• Air Timer Count • Value returned is read from NV_AIR_CNT_I. • See airTimer for more information
pRoamTimer-Cnt(optional)	• Roam Timer Count • Value returned is read from NV_ROAM_CNT_I. • See roamTimer for more information

pCurrTTY-Mode(optional)	• Current TTY Mode • Value returned is read from NV_TTY_I. – 0x00 - TTY_MODE_FULL - Full – 0x01 - TTY_MODE_VCO - Voice carry over – 0x02 - TTY_MODE_HCO - Hearing carry over – 0x03 - TTY_MODE_OFF - Off – 0xFF - Not Available
pCurPrefVoiceS-O(optional)	• Current Preferred Voice SO • Value returned is read from NV_PREF_VOICE_SO_I. • See voice_prefVoiceSO for more information
pCurAMR-Config(optional)	• Current Adaptive Multi-Rate Configuration. • Values returned are read from NV_GSM_ARM_CALL_CONFIG_I and NV_UMTS_AMR_COD-EC_PREFERENCE_CONFIG_I. • See voice_curAMRConfig for more information
pCurVoice-Privacy-Pref(optional)	• Current Voice Privacy Preference • Value returned is read from NV_VOICE_PRIV_I. – 0x00 - Standard privacy – 0x01 - Enhanced privacy – 0xFF - Not Available
pCurVoice-Domain-Pref(optional)	• Current Voice Domain Preference. – 0x00 - Circuit-switched (CS) only – 0x01 - Packet-switched (PS) only – 0x02 - CS is preferred; PS is secondary – 0x03 - PS is preferred; CS is secondary – 0xFF - Not Available

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1217.2 Field Documentation

8.1217.2.1 voice_airTimer∗ unpack_voice_SLQSVoiceGetConfig_t::pAirTimerCnt

8.1217.2.2 uint8_t∗ unpack_voice_SLQSVoiceGetConfig_t::pAutoAnswerStat

8.1217.2.3 voice_curAMRConfig∗ unpack_voice_SLQSVoiceGetConfig_t::pCurAMRConfig

8.1217.2.4 voice_prefVoiceSO∗ unpack_voice_SLQSVoiceGetConfig_t::pCurPrefVoiceSO

8.1217.2.5 uint8_t∗ unpack_voice_SLQSVoiceGetConfig_t::pCurrTTYMode

8.1217.2.6 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoiceDomainPref

8.1217.2.7 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoicePrivacyPref

8.1217.2.8 voice_roamTimer* unpack_voice_SLQSVoiceGetConfig_t::pRoamTimerCnt

8.1218 unpack_voice_SLQSVoiceManageCalls_t Struct Reference

Data Fields

- uint16_t * pFailCause

8.1218.1 Detailed Description

This structure contains Failure cause Information. Populated when API Fails.

Parameters

pFailCause	• Supplementary service failure causes (optional, supply NULL if not required). • See Table8 qaGobiApiTableVoiceCallEndReasons.h for supplementary services failure cause – 0xFFFF is the value when the information is not received from device

8.1218.2 Field Documentation

8.1218.2.1 uint16_t* unpack_voice_SLQSVoiceManageCalls_t::pFailCause

8.1219 unpack_voice_SLQSVoiceSendFlash_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1219.1 Detailed Description

This structure contains the flash information associated with a call.

Parameters

pCallID	• Unique call identifier associated with the current call.

8.1219.2 Field Documentation

8.1219.2.1 uint8_t* unpack_voice_SLQSVoiceSendFlash_t::pCallID

8.1220 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

8.1220.1 Detailed Description

This structure contains Voice Set Call Barring Password Unpack Parameters

Parameters

pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See voice_alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See voice_ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1220.2 Field Documentation

8.1220.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pAlphaIDInfo

8.1220.2.2 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCallID

8.1220.2.3 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCResType

8.1220.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCSUPSType

8.1220.2.5 uint16_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pFailCause

8.1221 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

8.1221.1 Detailed Description

This structure contains information about the Set Configuration Response Parameters.

Parameters

pAutoAnsStatus	• Auto Answer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pAirTimerStatus	• Air Timer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pRoamTimer-Status	• Roam Timer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pTTYConfig-Status	• TTY Config Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.

pPrefVoiceSO-Status	• Preferred Voice SO Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pVoiceDomain-PrefStatus	• Voice-Domain Preference Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.

Note

Parameters which are mentioned as NULL will be ignored.

8.1221.2 Field Documentation

8.1221.2.1 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAirTimerStatus

8.1221.2.2 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAutoAnsStatus

8.1221.2.3 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSOStatus

8.1221.2.4 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pRoamTimerStatus

8.1221.2.5 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pTTYConfigStatus

8.1221.2.6 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pVoiceDomainPrefStatus

8.1222 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference

Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

8.1222.1 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). – 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) – See voice_alphaIDInfo for more information
pCCResultType	• Call control result types (optional, supply NULL if not required) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - if the device does not provide this information
pCallID	• Unique call identifier for the dialed call (optional, supply NULL if not required) – 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) – See voice_ccSUPSType for more information

8.1222.2 Field Documentation

8.1222.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetSUPSService_t::pAlphaIDInfo

8.1222.2.2 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCallID

8.1222.2.3 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCCResultType

8.1222.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetSUPSService_t::pCCSUPSType

8.1222.2.5 uint16_t* unpack_voice_SLQSVoiceSetSUPSService_t::pFailCause

8.1223 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1223.1 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. Start continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • If the call ID value received is 0, no value has been returned by the device

8.1223.2 Field Documentation

8.1223.2.1 uint8_t* unpack_voice_SLQSVoiceStartContDTMF_t::pCallID

8.1224 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference

Data Fields

- uint8_t callID

8.1224.1 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. Stop continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • If the call ID value received is 0, no value has been returned by the device

8.1224.2 Field Documentation

8.1224.2.1 uint8_t unpack_voice_SLQSVoiceStopContDTMF_t::callID

8.1225 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference

Data Fields

- wds_DHCPProfileIdTlv ProfileIdTlv
- wds_DHCPLeaseStateTlv DHCPv4LeaseStateTlv
- wds_IPv4AdTlv IPv4AddrTlv
- wds_DHCPLeaseOptTlv DHCPv4LeaseOptTlv

8.1225.1 Field Documentation

8.1225.1.1 wds_DHCPLeaseOptTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseOptTlv

8.1225.1.2 wds_DHCPLeaseStateTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseStateTlv

8.1225.1.3 wds_IPv4AdTlv unpack_wds_DHCPv4ClientLease_ind_t::IPv4AddrTlv

8.1225.1.4 wds_DHCPProfileIdTlv unpack_wds_DHCPv4ClientLease_ind_t::ProfileIdTlv

8.1226 unpack_wds_GetAutoconnect_t Struct Reference

Data Fields

- uint32_t * psetting

8.1226.1 Detailed Description

auto connect data session setting parameter.

Parameters

pSetting	• NDIS auto connect setting – 0 - Disabled – 1 - Enabled

8.1226.2 Field Documentation

8.1226.2.1 uint32_t* unpack_wds_GetAutoconnect_t::psetting

8.1227 unpack_wds_GetByteTotals_t Struct Reference

Data Fields

- uint64_t * pTXTotalBytes
- uint64_t * pRXTotalBytes

8.1227.1 Detailed Description

Parameters

pTXTotalBytes	• Bytes transmitted without error
pRXTotalBytes	• Bytes received without error

8.1227.2 Field Documentation

8.1227.2.1 uint64_t* unpack_wds_GetByteTotals_t::pRXTotalBytes

8.1227.2.2 uint64_t* unpack_wds_GetByteTotals_t::pTXTotalBytes

8.1228 unpack_wds_GetConnectionRate_t Struct Reference

Data Fields

- uint32_t currentChannelTXRate
- uint32_t currentChannelRXRate
- uint32_t maxChannelTXRate
- uint32_t maxChannelRXRate

8.1228.1 Detailed Description

Parameters

currentChannelTXRate	Instantaneous channel Tx rate
currentChannelRXRate	Instantaneous channel Rx rate
maxChannelTXRate	Maximum Tx rate
maxChannelRXRate	Maximum Rx rate

8.1228.2 Field Documentation

8.1228.2.1 uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate

8.1228.2.2 uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate

8.1228.2.3 uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate

8.1228.2.4 uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate

8.1229 unpack_wds_GetDataBearerTechnology_t Struct Reference

Data Fields

- uint32_t * pDataBearer

8.1229.1 Detailed Description

Parameters

pDataBearer[OUT]	• Data bearer technology – 0x01 - CDMA2000 1x – 0x02 - CDMA 1xEV-DO Rev 0 – 0x03 - GSM – 0x04 - UMTS – 0x05 - CDMA2000 HRPD (1xEV-DO Rev A) – 0x06 - EDGE – 0x07 - HSDPA AND WCDMA – 0x08 - WCDMA AND HSUPA – 0x09 - HSDPA AND HSUPA – 0x0A - LTE – 0x0B - CDMA2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 0xFF - Unknown

8.1229.2 Field Documentation

8.1229.2.1 uint32_t* unpack_wds_GetDataBearerTechnology_t::pDataBearer

8.1230 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]

8.1230.1 Detailed Description

Parameters

pdptype	pdp type
ipaddr	ip address
pridns	primary dns
secdns	secondry dns
ipaddrv6	ip address v6
pridnsv6	primary dns v6
secdnsv6	secondry dns v6
namesize	profile name size
name	profile name
apnsize	apn size
apnname	apn name
usersize	username size
username	username

8.1230.2 Field Documentation

8.1230.2.1 int8_t unpack_wds_GetDefaultProfile_t::apnname[255]

8.1230.2.2 uint8_t unpack_wds_GetDefaultProfile_t::apnsize

8.1230.2.3 uint32_t unpack_wds_GetDefaultProfile_t::auth

8.1230.2.4 uint32_t unpack_wds_GetDefaultProfile_t::ipaddr

8.1230.2.5 uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6

8.1230.2.6 int8_t unpack_wds_GetDefaultProfile_t::name[255]

8.1230.2.7 uint8_t unpack_wds_GetDefaultProfile_t::namesize

8.1230.2.8 uint32_t unpack_wds_GetDefaultProfile_t::pdptype

8.1230.2.9 uint32_t unpack_wds_GetDefaultProfile_t::pridns

8.1230.2.10 uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6

8.1230.2.11 uint32_t unpack_wds_GetDefaultProfile_t::secdns

8.1230.2.12 uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6

8.1230.2.13 int8_t unpack_wds_GetDefaultProfile_t::username[255]

8.1230.2.14 uint8_t unpack_wds_GetDefaultProfile_t::usersize

8.1231 unpack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t index

8.1231.1 Detailed Description

Parameters

index	profile index

8.1231.2 Field Documentation

8.1231.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index

8.1232 unpack_wds_GetDormancyState_t Struct Reference

Data Fields

- uint32_t dormancyState

8.1232.1 Detailed Description

Parameters

dormancyState	dormancy status

8.1232.2 Field Documentation

8.1232.2.1 uint32_t unpack_wds_GetDormancyState_t::dormancyState

8.1233 unpack_wds_GetLastMobileIPError_t Struct Reference

Data Fields

- uint32_t error

8.1233.1 Detailed Description

Parameters

error	last mip status 0-success >0- error code

8.1233.2 Field Documentation

8.1233.2.1 uint32_t unpack_wds_GetLastMobileIPError_t::error

8.1234 unpack_wds_GetMobileIP_t Struct Reference

Data Fields

- uint32_t mipMode

8.1234.1 Detailed Description

Parameters

mipMode	mobile IP mode

8.1234.2 Field Documentation

8.1234.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

8.1235 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

8.1235.1 Detailed Description

Parameters

enabled	mobile ip profile state enabled/disabled
address	mobile ip profile home address
primaryHA	mobile ip profile home agent primary
secondaryHA	mobile ip profile secondary home agent address
revTunneling	mobile ip profile rev tunneling
naiSize	mobile ip profile NAI size
NAI	NAI string in ASCII text.
HASPI	HA security parameter index.
AAASPI	AAA server security parameter index.
HAState	Mobile IP Profile HA Key State
AAAState	Mobile IP Profile AAA Key State

8.1235.2 Field Documentation

8.1235.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

8.1235.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAAState

8.1235.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address

8.1235.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled

8.1235.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI

8.1235.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HAState

8.1235.2.7 int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]

8.1235.2.8 uint8_t unpack_wds_GetMobileIPProfile_t::naiSize

8.1235.2.9 uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA

8.1235.2.10 uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling

8.1235.2.11 uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA

8.1236 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

8.1236.1 Detailed Description

Parameters

pTXPacket-Successes	• No of transmitted Packets without error.
pRXPacket-Successes	• No 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	• No of bytes transmitted without error.
pRXOkBytes-Count	• No of bytes received without error.
pTXOKBytes-LastCall	• 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

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.

8.1236.2 Field Documentation

8.1236.2.1 uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount

8.1236.2.2 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount

8.1236.2.3 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOKBytesLastCall

8.1236.2.4 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors

8.1236.2.5 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows

8.1236.2.6 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses

8.1236.2.7 uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount

8.1236.2.8 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount

8.1236.2.9 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOKBytesLastCall

8.1236.2.10 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors

8.1236.2.11 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows

8.1236.2.12 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses

8.1237 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

8.1237.1 Detailed Description

Parameters

tXPacket-Successes	Tx Packets OK
rXPacket-Successes	Rx Packets OK
tXPacketErrors	Tx Packet Errors
rXPacketErrors	Rx Packet Errors
tXPacket-Overflows	Tx Overflows
rXPacket-Overflows	Rx Overflows
tXOkBytesCount	Tx Bytes OK
rXOkBytesCount	Rx Bytes OK
tXOKBytesLast-Call	Last call Tx Bytes OK
rXOKBytesLast-Call	Last call Rx Bytes OK
tXDroppedCount	Tx Packets Dropped
rXDroppedCount	Rx Packets Dropped

8.1237.2 Field Documentation

8.1237.2.1 uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount

8.1237.2.2 uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount

8.1237.2.3 uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall

8.1237.2.4 uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors

8.1237.2.5 uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows

8.1237.2.6 uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses

8.1237.2.7 uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount

8.1237.2.8 uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount

8.1237.2.9 uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall

8.1237.2.10 uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors

8.1237.2.11 uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows

8.1237.2.12 uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses

8.1238 unpack_wds_GetSessionDuration_t Struct Reference

Data Fields

- uint64_t callDuration

8.1238.1 Detailed Description

Parameters

callDuration	call duration in milliseconds

8.1238.2 Field Documentation

8.1238.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

8.1239 unpack_wds_GetSessionState_t Struct Reference

Data Fields

- uint32_t connectionStatus

8.1239.1 Detailed Description

Parameters

connection-Status	state of the current packet data session

8.1239.2 Field Documentation

8.1239.2.1 uint32_t unpack_wds_GetSessionState_t::connectionStatus

8.1240 unpack_wds_RMSetTransferStatistics_t Struct Reference

8.1241 unpack_wds_SetMobileIPProfile_t Struct Reference

8.1242 unpack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

- PackCreateProfileOut * pCreateProfileOut
- uint8_t * pProfileID
- uint16_t Tlvresult

8.1242.1 Detailed Description

Parameters

profile	type
profile	index
extended	error

8.1242.2 Field Documentation

8.1242.2.1 PackCreateProfileOut* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut

8.1242.2.2 uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID

8.1242.2.3 uint16_t unpack_wds_SLQSCreateProfile_t::Tlvresult

8.1243 unpack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint16_t extendedErrorCode

8.1243.1 Detailed Description

Parameters

extendedError-Code	extended error code

8.1243.2 Field Documentation

8.1243.2.1 uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode

8.1244 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

8.1244.1 Detailed Description

Parameters

	pLTEAttach-Profile	• Optional parameter • LTE Attach Profile – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) • This setting is deprecated on MC/EM74xx
	profileList	Profile List • an array of 4 profile configurations • Each element points to a single WORD value indicating profile • Optional parameter with possible values – 1 - 16 (MC/EM73xx and before) – 1 - 24 (MC/EM74xx and onwards) • function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present • Note: the 5th entry is currently ignored, please set it to zero

out	*defaultPDN-Enabled*	• 0 - disabled • 1 - enabled
out	*_3gppRelease*	3GPP release • 0 - Release_99 • 1 - Release_5 • 2 - Release_6 • 3 - Release_7 • 4 - Release_8 • 5 - Release_9 (In 9x30 and toworads) • 6 - Release_10 (In 9x30 and toworads) • 7 - Release_11 (In 9x30 and toworads)
out	*LTEAttach-ProfileList*	• pointer to WORD array indiciating LTE Attach Profile List – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – Please provide attach profiles in order of decreasing priority in this list.
in,out	*LTEAttach-ProfileListLen*	• Number of element in pLTEAttachProfileList – valid range: 1-24 – This setting is only supported for MC/EM74xx onwards

8.1244.2 Field Documentation

8.1244.2.1 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease

8.1244.2.2 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled

8.1244.2.3 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile

8.1244.2.4 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]

8.1244.2.5 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen

8.1244.2.6 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]

8.1245 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

Data Fields

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

8.1245.1 Detailed Description

Parameters

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

8.1245.2 Field Documentation

8.1245.2.1 currNetworkInfo unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]

8.1245.2.2 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen

8.1245.2.3 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork

8.1246 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

8.1246.1 Detailed Description

Parameters

current_channel-_tx_rate	• Current Channel Tx Rate. • Instantaneous channel Tx rate in bits per second. • In 9x15, this is the total current channel rate for all PDNs combined. • In 9x30 and later, this is the channel rate for a specific PDN.
current_channel-_rx_rate	• Current Channel Rx Rate. • Instantaneous channel Rx rate in bits per second. • In 9x15, this is the total current channel rate for all PDNs combined. • In 9x30 and later, this is the channel rate for a specific PDN
max_channel_-tx_rate	• Max Channel Tx Rate. • Maximum total Tx rate that modem is able to support in current serving system in bits per second. • In 9x15, this is a default hard coded value for the current serving system.
max_channel_-rx_rate	• Max Channel Rx Rate. • Maximum total Rx rate that modem is able to support in current serving system in bits per second. • In 9x15, this is a default hard coded value for the current serving system.

8.1246.2 Field Documentation

8.1246.2.1 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_rx_rate

8.1246.2.2 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_tx_rate

8.1246.2.3 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_rx_rate

8.1246.2.4 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_tx_rate

8.1247 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference

Data Fields

- uint8_t dataBearerMask
- qmiWDSDataBearerTechnology curDataBearerTechnology
- qmiWDSDataBearerTechnology lastCallDataBearerTechnology

8.1247.1 Detailed Description

Parameters

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

8.1247.2 Field Documentation

8.1247.2.1 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::curDataBearerTechnology

8.1247.2.2 uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask

8.1247.2.3 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearer-Technology

8.1248 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

8.1248.1 Detailed Description

Parameters

connectionStatus	Connection Status
callEndReason	Last Modem Call End Reason
txOKBytesCount	Tx Bytes OK
rxOKBytesCount	Rx Bytes OK
dormancyStatus	Dormancy Status
dataBearerTech	data bearer technology
channelRate	data Channel Rate
lastCallTXOKBytesCnt	Last Call Tx Bytes OK
lastCallRXOKBytesCnt	Last Call Rx Bytes OK
mdmCallDurationActive	Call active duration
lastCallDataBearerTech	Last Call Data Bearer Technology

8.1248.2 Field Documentation

8.1248.2.1 uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason

8.1248.2.2 dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate

8.1248.2.3 connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus

8.1248.2.4 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech

8.1248.2.5 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus

8.1248.2.6 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech

8.1248.2.7 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt

8.1248.2.8 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt

8.1248.2.9 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive

8.1248.2.10 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount

8.1248.2.11 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount

8.1249 unpack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- UnPackGetProfileSettingOut * pProfileSettings
- uint8_t ProfileType
- uint16_t Tlvresult

8.1249.1 Field Documentation

8.1249.1.1 UnPackGetProfileSettingOut* unpack_wds_SLQSGetProfileSettings_t::pProfileSettings

8.1249.1.2 uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType

8.1249.1.3 uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult

8.1250 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

8.1250.1 Detailed Description

Parameters

IPv4	ipv4 address
ProfileName	profile name
PDPType	PDP type
APNName	APN name
PrimaryDNSV4	
SecondaryDNS-V4	
UMTSGranted-QoS	UMTS Granted Qos
GPRSGranted-QoS	GPRS Granted QoS
Username	
Authentication	
ProfielID	

GWAddressV4	Gateway IPv4
SubnetMaskV4	Subnet mask IPV4
PCSCFAddrPCO	
PrimaryDNSV6	Primary DNS IPV6
SecondryDNSV6	Secondry DNS IPV6
UMTSGrantedQoS	UMTS Granted Qos
SecondaryDNSV4	
Mtu	Maximum Transfer Unit
DomainList	
IPFamilyPreference	

8.1250.2 Field Documentation

8.1250.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

8.1250.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

8.1250.2.3 struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList

8.1250.2.4 struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

8.1250.2.5 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4

8.1250.2.6 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag

8.1250.2.7 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference

8.1250.2.8 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4

8.1250.2.9 struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPV6AddrInfo

8.1250.2.10 struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPV6GWAddrInfo

8.1250.2.11 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu

8.1250.2.12 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO

8.1250.2.13 struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList

8.1250.2.14 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPType

8.1250.2.15 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4

8.1250.2.16 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]

8.1250.2.17 struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID

8.1250.2.18 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]

8.1250.2.19 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4

8.1250.2.20 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]

8.1250.2.21 struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList

8.1250.2.22 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4

8.1250.2.23 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology

8.1250.2.24 LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS

8.1250.2.25 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]

8.1251 unpack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint16_t * pExtErrorCode

8.1251.1 Detailed Description

Parameters

extended	error

8.1251.2 Field Documentation

8.1251.2.1 uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode

8.1252 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1252.1 Detailed Description

Parameters

Tlvresult	unpack result

8.1252.2 Field Documentation

8.1252.2.1 uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

8.1253 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

8.1253.1 Detailed Description

Parameters

conn_status	connection status
reconfigReqd	Indicates whether the network interface on the host needs to be reconfigured.
sessionEnd-Reason	Call End Reason
verboseSessn-EndReasonType	Verbose call end reason type
verboseSessn-EndReason	Reason the call ended (verbose)
ipFamily	IP family of the packet data connection.
techName	Technology name of the packet data connection.
bearerID	• bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection. • Valid Values - 0 to 16 • 0xFF - Invalid value.

8.1253.2 Field Documentation

8.1253.2.1 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID

8.1253.2.2 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status

8.1253.2.3 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily

8.1253.2.4 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd

8.1253.2.5 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason

8.1253.2.6 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName

8.1253.2.7 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason

8.1253.2.8 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType

8.1254 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

8.1254.1 Detailed Description

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	CData bearer technology Extended SO Mask

8.1254.2 Field Documentation

8.1254.2.1 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail

8.1254.2.2 wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]

8.1254.2.3 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail

8.1254.2.4 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail

8.1254.2.5 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechExtAvail

8.1254.2.6 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtRatValue

8.1254.2.7 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtSoMask

8.1254.2.8 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology

8.1254.2.9 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechnologyExt

8.1254.2.10 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail

8.1254.2.11 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus

8.1254.2.12 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail

8.1254.2.13 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus

8.1254.2.14 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen

8.1254.2.15 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork

8.1254.2.16 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask

8.1254.2.17 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes

8.1254.2.18 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts

8.1254.2.19 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask

8.1254.2.20 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes

8.1254.2.21 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts

8.1254.2.22 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail

8.1255 unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wdsDhcpv4HwConfig * pHwConfig
- wdsDhcpv4OptionList * pRequestOptionList

8.1255.1 Detailed Description

Parameters

pHwConfig	pointer to HW Config structure
pRequestOption-List	pointer to Option List structure to be sent in DHCP request

8.1255.2 Field Documentation

8.1255.2.1 wdsDhcpv4HwConfig∗ unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig

8.1255.2.2 wdsDhcpv4OptionList∗ unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList

8.1256 unpack_wds_SLQSSGetLoopback_t Struct Reference

Data Fields

- uint8_t ByteLoopbackMode
- uint8_t ByteLoopbackMultiplier

8.1256.1 Detailed Description

Parameters

ByteLoopback-Mode	• Loopback Mode. – 0 - Disable – 1 - Enable
ByteLoopback-Multiplier	• Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.1256.2 Field Documentation

8.1256.2.1 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode

8.1256.2.2 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier

8.1257 unpack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

- uint32_t ∗ psid
- uint32_t ∗ pFailureReason
- uint32_t ∗ pVerboseFailReasonType
- uint32_t ∗ pVerboseFailureReason

8.1257.1 Detailed Description

Parameters

psid	• Assigned session ID when starting a data session
pFailureReason	• Reason data session failed to be established • See qaGobiApiTableCallEndReasons.h for Call End Reason

pVerboseFail-ReasonType	• Parameter describing type of verbose failure reason • See qaGobiApiTableCallEndReasons.h for Call End Reason Type
pVerboseFailure-Reason	• Verbose reason explaining why call failed. Depends on verbFailReasonType parameter • See qaGobiApiTableCallEndReasons.h for Call End Reason

8.1257.2 Field Documentation

8.1257.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

8.1257.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

8.1257.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

8.1257.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

8.1258 unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

- uint8_t contextId
- uint8_t bearerId
- int8_t apnName [100]
- uint32_t ipv4Address
- uint32_t ipv4GWAddress
- uint32_t prDNSIPv4Address
- uint32_t seDNSIPv4Address
- struct ipv6AddressInfo ipv6Address
- struct ipv6AddressInfo ipv6GWAddress
- uint16_t prDNSIPv6Address [8]
- uint16_t seDNSIPv6Address [8]
- uint32_t prPCSCFIPv4Address
- uint32_t sePCSCFIPv4Address
- uint16_t prPCSCFIPv6Address [8]
- uint16_t sePCSCFIPv6Address [8]

8.1258.1 Detailed Description

Parameters

contextId	Context Identifier
bearerId	Bearer Identity
apnName	APN name associated with the context id
ipv4Address	IPv4 Address
ipv4GWAddress	IPv4 Gateway Address
prDNSIPv4-Address	Primary DNS IPv4 Address
seDNSIPv4-Address	Secondary DNS IPv4 Address
ipv6Address	IPv6 Address

ipv6GWAddress	IPv6 Gateway Address
prDNSIPv6-Address	Primary IPv6 DNS Address
seDNSIPv6-Address	Secondary IPv6 DNS Address
prPCSCFIPv4-Address	Primary PCSCF IPv4 Address
sePCSCFIPv4-Address	Secondary PCSCF IPv4 Address
prPCSCFIPv6-Address	Primary PCSCF IPv6 Address
sePCSCFIPv6-Address	Secondary PCSCF IPv6 Address

8.1258.2 Field Documentation

8.1258.2.1 int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]

8.1258.2.2 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId

8.1258.2.3 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.1258.2.4 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address

8.1258.2.5 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress

8.1258.2.6 struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address

8.1258.2.7 struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress

8.1258.2.8 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address

8.1258.2.9 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]

8.1258.2.10 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address

8.1258.2.11 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]

8.1258.2.12 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address

8.1258.2.13 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]

8.1258.2.14 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address

8.1258.2.15 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]

8.1259 UnPackGetProfileSettingOut Struct Reference

Data Fields

- UnpackQmiProfileInfo curProfile
- uint16_t * pExtErrCode

8.1259.1 Field Documentation

8.1259.1.1 UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile

8.1259.1.2 uint16_t* UnPackGetProfileSettingOut::pExtErrCode

8.1260 unpackWdsProfileParam Union Reference

Data Fields

- LibpackProfile3GPP SlqsProfile3GPP
- LibpackProfile3GPP2 SlqsProfile3GPP2

8.1260.1 Field Documentation

8.1260.1.1 LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP

8.1260.1.2 LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2

8.1261 USBCompConfig Struct Reference

Data Fields

- BYTE * pUSBComp

8.1261.1 Detailed Description

This structure is used to store USB composition information

Parameters

pUSBComp[IN]	• Current USB Composition • Values: – 0..5 - Reserved (non-QMI) – 6 - DM NMEA AT QMI – 7 - DM NMEA AT QMI1 QMI2 QMI3 – 8 - DM NMEA AT MBIM – 9 - MBIM – 10 - NMEA MBIM – 11 - DM MBIM – 12 - DM NMEA MBIM 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces – 13 - 6 for QMI, 8 for MBIM – 14 - 6 for QMI, 9 for MBIM – 15 - 6 for QMI, 10 for MBIM – 16 - 6 for QMI, 11 for MBIM – 17 - 6 for QMI, 12 for MBIM – 18 - 7 for QMI, 8 for MBIM – 19 - 7 for QMI, 9 for MBIM – 20 - 7 for QMI, 10 for MBIM – 21 - 7 for QMI, 11 for MBIM – 22 - 7 for QMI, 12 for MBIM

8.1261.2 Field Documentation

8.1261.2.1 BYTE* USBCompConfig::pUSBComp

8.1262 USBCompParams Struct Reference

Data Fields

- BYTE * pUSBComp
- BYTE * pNumSupUSBComps
- BYTE * pSupUSBComps

8.1262.1 Detailed Description

This structure is used to store retrieved USB Composition

Parameters

pUSBComp[OUT]	• Current USB Composition(optional parameter) • Values: – 0..5 - Reserved (non-QMI) – 6 - DM NMEA AT QMI – 7 - DM NMEA AT QMI1 QMI2 QMI3 – 8 - DM NMEA AT MBIM – 9 - MBIM – 10 - NMEA MBIM – 11 - DM MBIM – 12 - DM NMEA MBIM 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces – 13 - 6 for QMI, 8 for MBIM – 14 - 6 for QMI, 9 for MBIM – 15 - 6 for QMI, 10 for MBIM – 16 - 6 for QMI, 11 for MBIM – 17 - 6 for QMI, 12 for MBIM – 18 - 7 for QMI, 8 for MBIM – 19 - 7 for QMI, 9 for MBIM – 20 - 7 for QMI, 10 for MBIM – 21 - 7 for QMI, 11 for MBIM – 22 - 7 for QMI, 12 for MBIM
pNumSupUSBComps[OUT]	• Number of supported USB compositions in the parameter to follow • Range - 0-255

pSupUSB-Comps[OUT]	• Optional parameter • List of supported USB compositions(1 Byte each - Max 255) • Total length is defined by pNumSupUSBComps parameter • Values: – 0..5 - Reserved (non-QMI) – 6 - DM NMEA AT QMI – 7 - DM NMEA AT QMI1 QMI2 QMI3 – 8 - DM NMEA AT MBIM – 9 - MBIM – 10 - NMEA MBIM – 11 - DM MBIM – 12 - DM NMEA MBIM 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces – 13 - 6 for QMI, 8 for MBIM – 14 - 6 for QMI, 9 for MBIM – 15 - 6 for QMI, 10 for MBIM – 16 - 6 for QMI, 11 for MBIM – 17 - 6 for QMI, 12 for MBIM – 18 - 7 for QMI, 8 for MBIM – 19 - 7 for QMI, 9 for MBIM – 20 - 7 for QMI, 10 for MBIM – 21 - 7 for QMI, 11 for MBIM – 22 - 7 for QMI, 12 for MBIM

8.1262.2 Field Documentation

8.1262.2.1 BYTE* USBCompParams::pNumSupUSBComps

8.1262.2.2 BYTE* USBCompParams::pSupUSBComps

8.1262.2.3 BYTE* USBCompParams::pUSBComp

8.1263 USSDNoWaitIndicationInfo Struct Reference

Data Fields

- BYTE * pError
- BYTE * pFailureCause
- struct USSInfo * pUSSDData
- alphaIDInfo * pAlphaIdentifier

8.1263.1 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. • See USSInfo for more details.

8.1263.2 Field Documentation

8.1263.2.1 alphaIDInfo* USSDNoWaitIndicationInfo::pAlphaIdentifier

8.1263.2.2 BYTE* USSDNoWaitIndicationInfo::pError

8.1263.2.3 BYTE* USSDNoWaitIndicationInfo::pFailureCause

8.1263.2.4 struct USSInfo* USSDNoWaitIndicationInfo::pUSSDData

8.1264 USSDRespFNetwork Struct Reference

Data Fields

- char * pTypeCode
- char * pRespData

8.1264.1 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

8.1264.2 Field Documentation

8.1264.2.1 char* USSDRespFNetwork::pRespData

8.1264.2.2 char* USSDRespFNetwork::pTypeCode

8.1265 USSInfo Struct Reference

Data Fields

- BYTE ussDCS
- BYTE ussLen
- BYTE ussData [182]

8.1265.1 Detailed Description

This structure contains USS Information

Parameters

ussDCS	• 1 - ASCII coding scheme • 2 - 8-BIT coding scheme • 3 - UCS2
ussLen	• Range 1 to 182
ussData	• Data encoded as per the DCS

8.1265.2 Field Documentation

8.1265.2.1 BYTE USSInfo::ussData[182]

8.1265.2.2 BYTE USSInfo::ussDCS

8.1265.2.3 BYTE USSInfo::ussLen

8.1266 USSResp Struct Reference

Data Fields

- WORD * pfailureCause
- alphaIDInfo * pAlphaIDInfo
- struct USSInfo * pUSSDInfo
- BYTE * pCcResultType
- BYTE * pCallId
- ccSUPSType * pCCSuppsType

8.1266.1 Field Documentation

8.1266.1.1 alphaIDInfo* USSResp::pAlphaIDInfo

8.1266.1.2 BYTE* USSResp::pCallId

8.1266.1.3 BYTE* USSResp::pCcResultType

8.1266.1.4 ccSUPSType* USSResp::pCCSuppsType

8.1266.1.5 WORD* USSResp::pfailureCause

8.1266.1.6 struct USSInfo* USSResp::pUSSDInfo

8.1267 UUSInfo Struct Reference

Data Fields

- BYTE UUSType
- BYTE UUSDcs
- BYTE UUSDatalen
- BYTE UUSData [255]

8.1267.1 Detailed Description

This structure contains User to User Signaling Service Information.

Parameters

UUSType	• UUS type values are: – 0x00 - UUS_DATA – 0x01 - UUS_TYPE1_IMPLICIT – 0x02 - UUS_TYPE1_REQUIRED – 0x03 - UUS_TYPE1_NOT_REQUIRED – 0x04 - UUS_TYPE2_REQUIRED – 0x05 - UUS_TYPE2_NOT_REQUIRED – 0x06 - UUS_TYPE3_REQUIRED – 0x07 - UUS_TYPE3_NOT_REQUIRED – 0xFF - Not Available
UUSDcs	• UUS data coding scheme values are: – 0x01 - UUS_DCS_USP – 0x02 - UUS_DCS_OHLP – 0x03 - UUS_DCS_X244 – 0x04 - UUS_DCS_SMCF – 0x05 - UUS_DCS_IA5 – 0x06 - UUS_DCS_RV12RD – 0x07 - UUS_DCS_Q931UNCCM – 0xFF - Not Available
UUSDatalen	• Number of sets of the following elements. – UUSData • If zero(0) then no further information exists.
UUSData[MAX_-DESCRIPTION-_LENGTH]	• UUS data encoded as per coding scheme

8.1267.2 Field Documentation

8.1267.2.1 BYTE UUSInfo::UUSData[255]

8.1267.2.2 BYTE UUSInfo::UUSDatalen

8.1267.2.3 BYTE UUSInfo::UUSDcs

8.1267.2.4 BYTE UUSInfo::UUSType

8.1268 verifyUIMPIN Struct Reference

Data Fields

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

8.1268.1 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. – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN – 4 - Hidden key
pinLen	• Length of the following elements i.e. pin value.
pinVal[MAX_DESCRIPTION_LENGTH]	• PIN value. • This value is a sequence of ASCII characters.

8.1268.2 Field Documentation

8.1268.2.1 BYTE verifyUIMPIN::pinID

8.1268.2.2 BYTE verifyUIMPIN::pinLen

8.1268.2.3 BYTE verifyUIMPIN::pinVal[255]

8.1269 voice_airTimer Struct Reference

Data Fields

- uint8_t namID
- uint32_t airTimerValue

8.1269.1 Detailed Description

This structure contains information about the Air Timer.

Parameters

namID	• Index of the NAM(Number Assignment Module) to be configured. • Range 0 to 3. • Some modems support only 1 or 2 NAMs. • 0xFF,if not available.
airTimerValue	• Time in minutes. • Cumulative air time is slammed. • 0xFFFFFFFF,if not available.

8.1269.2 Field Documentation

8.1269.2.1 uint32_t voice_airTimer::airTimerValue

8.1269.2.2 uint8_t voice_airTimer::namID

8.1270 voice_allCallsAlphaIDInfo Struct Reference

Data Fields

- uint8_t callID
- voice_alphaIDInfo AlphaIDInfo

8.1270.1 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.

8.1270.2 Field Documentation

8.1270.2.1 voice_alphaIDInfo voice_allCallsAlphaIDInfo::AlphaIDInfo

8.1270.2.2 uint8_t voice_allCallsAlphaIDInfo::callID

8.1271 voice_allCallsDiagInfo Struct Reference

Data Fields

- uint8_t callID
- voice_diagInfo DiagInfo

8.1271.1 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.

8.1271.2 Field Documentation

8.1271.2.1 uint8_t voice_allCallsDiagInfo::callID

8.1271.2.2 voice_diagInfo voice_allCallsDiagInfo::DiagInfo

8.1272 voice_allCallsUUSInfo Struct Reference

Data Fields

- uint8_t callID
- voice_UUSInfo uusInfo

8.1272.1 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.

8.1272.2 Field Documentation

8.1272.2.1 uint8_t voice_allCallsUUSInfo::callID

8.1272.2.2 voice_UUSInfo voice_allCallsUUSInfo::uusInfo

8.1273 voice_alphaIDInfo Struct Reference

Data Fields

- uint8_t alphaDcs
- uint8_t alphaLen
- uint8_t alphaText [255]

8.1273.1 Detailed Description

This structure contains information about the Alpha Identifier.

Parameters

alphaDcs	• Alpha coding scheme – 0x01 - GSM Default_Char – 0x02 - UCS2 – 0xFF - Not Available
alphaLen	• Number of sets of the following elements: – pAlpha_text • If zero(0) then no further information exists.
alphaText[MAX_VOICE_DESCRIPTION_LENGTH]	• Data encoded as per the alpha_dcs

8.1273.2 Field Documentation

8.1273.2.1 uint8_t voice_alphaIDInfo::alphaDcs

8.1273.2.2 uint8_t voice_alphaIDInfo::alphaLen

8.1273.2.3 uint8_t voice_alphaIDInfo::alphaText[255]

8.1274 voice_arrAlertingPattern Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint32_t alertingPattern [20]

8.1274.1 Detailed Description

This structure contains an array of Alerting Pattern.

Parameters

numInstances	• Number of callID, alertingPattern that follow. • If zero(0) then no further information exists.

callID[VOICE_-MAX_NO_OF_-CALLS]	• Array of Unique call identifier for the call.
alertingPattern[-VOICE_MAX_N-O_OF_CALLS]	• Array of Alerting pattern. – 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1 – 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2 – 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3 – 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5 – 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6 – 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7 – 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8 – 0x08 - QMI_VOICE_ALERTING_PATTERN_9 - Pattern 9

8.1274.2 Field Documentation

8.1274.2.1 uint32_t voice_arrAlertingPattern::alertingPattern[20]

8.1274.2.2 uint8_t voice_arrAlertingPattern::callID[20]

8.1274.2.3 uint8_t voice_arrAlertingPattern::numInstances

8.1275 voice_arrAlertingType Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint8_t AlertingType [20]

8.1275.1 Detailed Description

This structure contains an array of Alerting Type.

Parameters

numInstances	• Number of callID, AlertingType that follow. • If zero(0) then no further information exists.
callID[VOICE_-MAX_NO_OF_-CALLS]	• Array of Unique call identifier for the call.
AlertingType[V-OICE_MAX_NO-_OF_CALLS]	• Array of Alerting type. – 0x00 - ALERTING_LOCAL - Local – 0x01 - ALERTING_REMOTE - Remote

8.1275.2 Field Documentation

8.1275.2.1 uint8_t voice_arrAlertingType::AlertingType[20]

8.1275.2.2 uint8_t voice_arrAlertingType::callID[20]

8.1275.2.3 uint8_t voice_arrAlertingType::numInstances

8.1276 voice_arrAlphaID Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsAlphaIDInfo allCallsAlphaIDInfoArr [20]

8.1276.1 Detailed Description

This structure contains an array of Alpha ID Info

Parameters

numInstances	• Number of allCallsAlphaIDInfo that follow. • If zero(0) then no further information exists.
allCallsAlphaIDInfo[VOICE_MAX_NO_OF_CALLS]	• Array of allCallsAlphaIDInfo. • See voice_allCallsAlphaIDInfo for more information.

8.1276.2 Field Documentation

8.1276.2.1 voice_allCallsAlphaIDInfo voice_arrAlphaID::allCallsAlphaIDInfoArr[20]

8.1276.2.2 uint8_t voice_arrAlphaID::numInstances

8.1277 voice_arrCalledPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo CalledPartyNum [20]

8.1277.1 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. • If zero(0) then no further information exists.

CalledPartyNum[VOICE_MAX_NO_OF_CALLS]	• Array of CalledPartyNum. • See voice_peerNumberInfo for more information.

8.1277.2 Field Documentation

8.1277.2.1 voice_peerNumberInfo voice_arrCalledPartyNum::CalledPartyNum[20]

8.1277.2.2 uint8_t voice_arrCalledPartyNum::numInstances

8.1278 voice_arrCallEndReason Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t callEndReason [20]

8.1278.1 Detailed Description

This structure contains an array of Call End Reasons.

Parameters

numInstances	• Number of callID, callEndReason that follow. • If zero(0) then no further information exists.
callID[VOICE_MAX_NO_OF_CALLS]	• Array of Unique call identifier for the call.
callEndReason[VOICE_MAX_NO_OF_CALLS]	• Array of Call End Reason . • See Table9 qaGobiApiTableVoiceCallEndReasons.h for a list of valid voice-related call end reasons

8.1278.2 Field Documentation

8.1278.2.1 uint16_t voice_arrCallEndReason::callEndReason[20]

8.1278.2.2 uint8_t voice_arrCallEndReason::callID[20]

8.1278.2.3 uint8_t voice_arrCallEndReason::numInstances

8.1279 voice_arrCallInfo Struct Reference

Data Fields

- uint8_t numInstances

- voice_getAllCallInformation getAllCallInfo [20]

8.1279.1 Detailed Description

This structure contains an array of Call Info

Parameters

numInstances	• Number of getAllCallInfo that follow. • If zero(0) then no further information exists.
getAllCallInfo[VOICE_MAX_NO_OF_CALLS]	• Array of CallInfo. • See voice_getAllCallInfo for more information.

8.1279.2 Field Documentation

8.1279.2.1 voice_getAllCallInformation voice_arrCallInfo::getAllCallInfo[20]

8.1279.2.2 uint8_t voice_arrCallInfo::numInstances

8.1280 voice_arrConnectPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo ConnectedPartyNum [20]

8.1280.1 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. • If zero(0) then no further information exists.
ConnectedPartyNum[VOICE_MAX_NO_OF_CALLS]	• Array of ConnectedPartyNum. • See voice_peerNumberInfo for more information.

8.1280.2 Field Documentation

8.1280.2.1 voice_peerNumberInfo voice_arrConnectPartyNum::ConnectedPartyNum[20]

8.1280.2.2 uint8_t voice_arrConnectPartyNum::numInstances

8.1281 voice_arrDiagInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsDiagInfo DiagInfo [20]

8.1281.1 Detailed Description

This structure contains an array of Diagnostic Information.

Parameters

numInstances	• Number of DiagInfo that follow. • If zero(0) then no further information exists.
DiagInfo[VOICE_MAX_NO_OF_CALLS]	• Array of DiagInfo. • See voice_allCallsDiagInfo for more information.

8.1281.2 Field Documentation

8.1281.2.1 voice_allCallsDiagInfo voice_arrDiagInfo::DiagInfo[20]

8.1281.2.2 uint8_t voice_arrDiagInfo::numInstances

8.1282 voice_arrRedirPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo RedirPartyNum [20]

8.1282.1 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. • If zero(0) then no further information exists.
RedirPartyNum[VOICE_MAX_NO_OF_CALLS]	• Array of RedirPartyNum. • See voice_peerNumberInfo for more information.

8.1282.2 Field Documentation

8.1282.2.1 uint8_t voice_arrRedirPartyNum::numInstances

8.1282.2.2 voice_peerNumberInfo voice_arrRedirPartyNum::RedirPartyNum[20]

8.1283 voice_arrRemotePartyName Struct Reference

Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyName GetAllCallRmtPtyName [20]

8.1283.1 Detailed Description

This structure contains an array of Remote Party Names

Parameters

numInstances	• Number of remotePartyName that follow. • If zero(0) then no further information exists.
GetAllCallRmtPtyName[MAX_NO_OF_CALLS]	• Array of remotePartyName. • See voice_getAllCallRmtPtyName for more information.

8.1283.2 Field Documentation

8.1283.2.1 voice_getAllCallRmtPtyName voice_arrRemotePartyName::GetAllCallRmtPtyName[20]

8.1283.2.2 uint8_t voice_arrRemotePartyName::numInstances

8.1284 voice_arrRemotePartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyNum RmtPtyNum [20]

8.1284.1 Detailed Description

This structure contains an array of Remote Party Numbers

Parameters

numInstances	• Number of remotePartyNum that follow. • If zero(0) then no further information exists.
RmtPtyNum[VOICE_MAX_NO_OF_CALLS]	• Array of remotePartyNum. • See voice_getAllCallRmtPtyNum for more information.

8.1284.2 Field Documentation

8.1284.2.1 uint8_t voice_arrRemotePartyNum::numInstances

8.1284.2.2 voice_getAllCallRmtPtyNum voice_arrRemotePartyNum::RmtPtyNum[20]

8.1285 voice_arrSvcOption Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t srvOption [20]

8.1285.1 Detailed Description

This structure contains array an of Servicing option.

Parameters

numInstances	• Number of callID, srvOption that follow. • 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. • See Table9 qaGobiApiTableServiceOptions.h for standard service option number assignments.

8.1285.2 Field Documentation

8.1285.2.1 uint8_t voice_arrSvcOption::callID[20]

8.1285.2.2 uint8_t voice_arrSvcOption::numInstances

8.1285.2.3 uint16_t voice_arrSvcOption::srvOption[20]

8.1286 voice_arrUUSInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsUUSInfo AllCallsUUSInfo [20]

8.1286.1 Detailed Description

This structure contains an array of User to User Signaling Service Information

Parameters

numInstances	• Number of allCallsUUSInfo that follow. • If zero(0) then no further information exists.
AllCallsUUSInfo[VOICE_MAX_NO_OF_CALLS]	• Array of allCallsUUSInfo. • See voice_allCallsUUSInfo for more information.

8.1286.2 Field Documentation

8.1286.2.1 voice_allCallsUUSInfo voice_arrUUSInfo::AllCallsUUSInfo[20]

8.1286.2.2 uint8_t voice_arrUUSInfo::numInstances

8.1287 voice_burstDTMFInfo Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t digitCnt
- uint8_t pDigitBuff [255]

8.1287.1 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. A burst DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • 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_VOICE_DESCRIPTION_LENGTH]	• DTMF digit buffer in ASCII, NULL terminated

8.1287.2 Field Documentation

8.1287.2.1 uint8_t voice_burstDTMFInfo::digitCnt

8.1287.2.2 uint8_t* voice_burstDTMFInfo::pCallID

8.1287.2.3 uint8_t voice_burstDTMFInfo::pDigitBuff[255]

8.1288 voice_calledPartySubAdd Struct Reference

Data Fields

- uint8_t extBit
- uint8_t subAddrType
- uint8_t oddEvenInd
- uint8_t subAddrLen
- uint8_t subAddr [255]

8.1288.1 Detailed Description

This structure contains information about the Called Sub Party Addresses.

Parameters

extBit	• Extension bit.
subAddrType	• Subaddress type. – 0x00 - NSAP – 0x01 - USER
oddEvenInd	• Even/odd indicator. – 0x00 - Even number of address signals – 0x01 - Odd number of address signals
subAddrLen	• Number of sets of the following elements: – SubAddress
subAddr[MAX_VOICE_DESCRIPTION_LENGTH]	• Array of the SubAddress in BCD number format.

8.1288.2 Field Documentation

8.1288.2.1 uint8_t voice_calledPartySubAdd::extBit

8.1288.2.2 uint8_t voice_calledPartySubAdd::oddEvenInd

8.1288.2.3 uint8_t voice_calledPartySubAdd::subAddr[255]

8.1288.2.4 uint8_t voice_calledPartySubAdd::subAddrLen

8.1288.2.5 uint8_t voice_calledPartySubAdd::subAddrType

8.1289 voice_callFwdTypeAndPlan Struct Reference

Data Fields

- uint8_t numberType
- uint8_t numberPlan

8.1289.1 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 – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION Reserved extension
numberPlan	• Call forwarding number plan – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION Reserved extension

8.1289.2 Field Documentation

8.1289.2.1 uint8_t voice_callFwdTypeAndPlan::numberPlan

8.1289.2.2 uint8_t voice_callFwdTypeAndPlan::numberType

8.1290 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]

8.1290.1 Detailed Description

This structure contains information for Get Call Forwarding Extended Information.

Parameters

SvcStatus	• Service status. Values: – 0x00 - SERVICE_STATUS_INACTIVE - Inactive – 0x01 - SERVICE_STATUS_ACTIVE - Active
SvcClass	• Service Class is a combination (sum) of information class constants • See qaGobiApiTableSupServiceInfoClasses.h for service classes.
noReplyTimer	• No reply timer value in seconds • 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. • Values: – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
numType	• Number type. • Values: – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension

numPlan	• Number plan. • Values: – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 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.

8.1290.2 Field Documentation

8.1290.2.1 uint8_t voice_callFWExtInfo::noReplyTimer

8.1290.2.2 uint8_t voice_callFWExtInfo::number[255]

8.1290.2.3 uint8_t voice_callFWExtInfo::numLen

8.1290.2.4 uint8_t voice_callFWExtInfo::numPlan

8.1290.2.5 uint8_t voice_callFWExtInfo::numType

8.1290.2.6 uint8_t voice_callFWExtInfo::PI

8.1290.2.7 uint8_t voice_callFWExtInfo::SI

8.1290.2.8 uint8_t voice_callFWExtInfo::SvcClass

8.1290.2.9 uint8_t voice_callFWExtInfo::SvcStatus

8.1291 voice_callFWInfo Struct Reference

Data Fields

- uint8_t SvcStatus
- uint8_t SvcClass
- uint8_t numLen
- uint8_t number [255]
- uint8_t noReplyTimer

8.1291.1 Detailed Description

This structure contains information for Get Call Forwarding Information.

Parameters

SvcStatus	• Service status. Values: – 0x00 - SERVICE_STATUS_INACTIVE - Inactive – 0x01 - SERVICE_STATUS_ACTIVE - Active
SvcClass	• Service Class is a combination (sum) of information class constants • 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 • A value of 0 indicates that noReplyTimer is ignored.

8.1291.2 Field Documentation

8.1291.2.1 uint8_t voice_callFWInfo::noReplyTimer

8.1291.2.2 uint8_t voice_callFWInfo::number[255]

8.1291.2.3 uint8_t voice_callFWInfo::numLen

8.1291.2.4 uint8_t voice_callFWInfo::SvcClass

8.1291.2.5 uint8_t voice_callFWInfo::SvcStatus

8.1292 voice_callInfo Struct Reference

Data Fields

- uint8_t callID
- uint8_t callState
- uint8_t callType
- uint8_t direction
- uint8_t mode

8.1292.1 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. • If zero(0) then invalid.
callState	• Call state. – 0x01 - CALL_STATE_ORIGINATION - Origination – 0x02 - CALL_STATE_INCOMING - Incoming – 0x03 - CALL_STATE_CONVERSATION - Conversation – 0x04 - CALL_STATE_CC_IN_PROGRESS - Call is originating but waiting for call control to complete – 0x05 - CALL_STATE_ALERTING - Alerting – 0x06 - CALL_STATE_HOLD - Hold – 0x07 - CALL_STATE_WAITING - Waiting – 0x08 - CALL_STATE_DISCONNECTING - Disconnecting – 0x09 - CALL_STATE_END - End – 0x0A - CALL_STATE_SETUP - MT call is in Setup state in 3GPP – 0xFF - Not Available
callType	• Call type. – 0x00 - CALL_TYPE_VOICE - Voice – 0x02 - CALL_TYPE_VOICE_IP - Voice over IP – 0x06 - CALL_TYPE_OTAPA - OTAPA – 0x07 - CALL_TYPE_STD_OTASP - Standard OTASP – 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP – 0x09 - CALL_TYPE_EMERGENCY - Emergency – 0xFF - Not Available
direction	• Direction. – 0x01 - CALL_DIRECTION_MO - MO call – 0x02 - CALL_DIRECTION_MT - MT call – 0xFF - Not Available
mode	• Mode. • 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. – 0x01 - CALL_MODE_CDMA - CDMA – 0x02 - CALL_MODE_GSM - GSM – 0x03 - CALL_MODE_UMTS - UMTS – 0x04 - CALL_MODE_LTE - LTE – 0x05 - CALL_MODE_TDS - TD-SCDMA – 0xFF - Not Available

8.1292.2 Field Documentation

8.1292.2.1 uint8_t voice_callInfo::callID

8.1292.2.2 uint8_t voice_callInfo::callState

8.1292.2.3 uint8_t voice_callInfo::callType

8.1292.2.4 uint8_t voice_callInfo::direction

8.1292.2.5 uint8_t voice_callInfo::mode

8.1293 voice_ccSUPSType Struct Reference

Data Fields

- uint8_t svcType
- uint8_t reason

8.1293.1 Detailed Description

This structure contains information about the Call Control Supplementary Service Types

Parameters

svcType	• Service type. – 0x01 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ACTIVATE - Activate – 0x02 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_DEACTIVATE - Deactivate – 0x03 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER - Register – 0x04 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ERASE - Erase – 0x05 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_INTERROGATE - Interrogate – 0x06 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER_PASSWORD - Register password – 0x07 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_USSD - USSD – 0xFF - Not Available
reason	• Call control supplementary service result reason • Values: – See qaGobiApiTableCallControlReturnReasons.h for return reasons.

8.1293.2 Field Documentation

8.1293.2.1 uint8_t voice_ccSUPSType::reason

8.1293.2.2 uint8_t voice_ccSUPSType::svcType

8.1294 voice_CLIPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1294.1 Detailed Description

This structure contains information about the Calling Line Identification Presentation (CLIP) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.1294.2 Field Documentation

8.1294.2.1 uint8_t voice_CLIPResp::ActiveStatus

8.1294.2.2 uint8_t voice_CLIPResp::ProvisionStatus

8.1295 voice_CLIRResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1295.1 Detailed Description

This structure contains information about the Calling Line Identification Restriction (CLIR) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available

ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED_PERMANENT - Permanently provisioned – 0x02 - PROVISION_STATUS_PRESENTATION_RESTRICTED - Restricted presentation – 0x03 - PROVISION_STATUS_PRESENTATION_ALLOWED - Allowed presentation – 0xFF - Not Available

8.1295.2 Field Documentation

8.1295.2.1 uint8_t voice_CLIRResp::ActiveStatus

8.1295.2.2 uint8_t voice_CLIRResp::ProvisionStatus

8.1296 voice_CNAPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1296.1 Detailed Description

This structure contains information about the Calling Name Presentation (CNAP) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.1296.2 Field Documentation

8.1296.2.1 uint8_t voice_CNAPResp::ActiveStatus

8.1296.2.2 uint8_t voice_CNAPResp::ProvisionStatus

8.1297 voice_COLPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1297.1 Detailed Description

This structure contains information about the Connected Line Identification Presentation (COLP) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.1297.2 Field Documentation

8.1297.2.1 uint8_t voice_COLPResp::ActiveStatus

8.1297.2.2 uint8_t voice_COLPResp::ProvisionStatus

8.1298 voice_COLRResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1298.1 Detailed Description

This structure contains information about the Connected Line Identification Restriction (COLR) supplementary service responses.

Parameters

ActiveStatus	• Active status. • Values: – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
ProvisionStatus	• Provisioned status. • Values: – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.1298.2 Field Documentation

8.1298.2.1 uint8_t voice_COLRResp::ActiveStatus

8.1298.2.2 uint8_t voice_COLRResp::ProvisionStatus

8.1299 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]

8.1299.1 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 – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available

screeningInd	• Screening indicator. – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network – 0xFF - Not Available
numType	• Number type. – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension – 0xFF - Not Available
numPlan	• Number plan. – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension – 0xFF - Not Available
callerIDLen	• Provides the length of caller ID which follow. • If zero(0) then no further information exists.
callerID[MAX_VOICE_CALL_NO_LEN]	• callerID of numLen length, NULL terminated.

8.1299.2 Field Documentation

8.1299.2.1 uint8_t voice_connectNumInfo::callerID[81]

8.1299.2.2 uint8_t voice_connectNumInfo::callerIDLen

8.1299.2.3 uint8_t voice_connectNumInfo::numPlan

8.1299.2.4 uint8_t voice_connectNumInfo::numPresInd

8.1299.2.5 uint8_t voice_connectNumInfo::numType

8.1299.2.6 uint8_t voice_connectNumInfo::screeningInd

8.1300 voice_CUGInfo Struct Reference

Data Fields

- uint16_t CUGIndex
- uint8_t SuppPrefCUG
- uint8_t SuppOA

8.1300.1 Detailed Description

This structure contains Closed User Group Information

Parameters

CUGIndex	• Range 0x00... 0x7FFF
SuppPrefCUG	• Suppress preferential CUG – 0x00 - FALSE – 0x01 - TRUE
SuppOA	• Suppress OA subscription option – 0x00 - FALSE – 0x01 - TRUE

8.1300.2 Field Documentation

8.1300.2.1 uint16_t voice_CUGInfo::CUGIndex

8.1300.2.2 uint8_t voice_CUGInfo::SuppOA

8.1300.2.3 uint8_t voice_CUGInfo::SuppPrefCUG

8.1301 voice_curAMRConfig Struct Reference

Data Fields

- uint8_t gsmAmrStat
- uint8_t wcdmaAmrStat

8.1301.1 Detailed Description

This structure contains the Current Adaptive Multi Rate Configuration Information.

Parameters

gsmAmrStat	• GSM AMR Status – 0x00 - Disable – 0x01 - Enable – 0xFF - Not Available
wcdmaAmrStat	• WCDMA AMR Status • One or a combination of the following bitmask values: – Bit 0 - AMR codec advertised is not supported – Bit 1 - Controls WCDMA AMR wideband – Bit 2 - Controls GSM half rate AMR – Bit 3 - Controls GSM AMR wideband – Bit 4 - Controls GSM AMR narrowband • 0xFF, if not available

8.1301.2 Field Documentation

8.1301.2.1 uint8_t voice_curAMRConfig::gsmAmrStat

8.1301.2.2 uint8_t voice_curAMRConfig::wcdmaAmrStat

8.1302 voice_diagInfo Struct Reference

Data Fields

- uint8_t diagInfoLen
- uint8_t diagnosticInfo [255]

8.1302.1 Detailed Description

This structure contains Diagnostic Information

Parameters

diagInfoLen	• Provides the length of information which follow. • If zero(0) then no further information exists.
diagnosticInfo[MAX_VOICE_DESCRIPTION_LENGTH]	• Diagnostic information.

8.1302.2 Field Documentation

8.1302.2.1 uint8_t voice_diagInfo::diagInfoLen

8.1302.2.2 uint8_t voice_diagInfo::diagnosticInfo[255]

8.1303 voice_DTMFLengths Struct Reference

Data Fields

- uint8_t DTMFPulseWidth
- uint8_t DTMFInterdigitInterval

8.1303.1 Detailed Description

This structure contains Voice Burst DTMF pulse length information

Parameters

DTMFPulse-Width	• DTMF pulse width. Values: – 0x00 - DTMF_ONLENGTH_95MS - 95 ms – 0x01 - DTMF_ONLENGTH_150MS - 150 ms – 0x02 - DTMF_ONLENGTH_200MS - 200 ms – 0x03 - DTMF_ONLENGTH_250MS - 250 ms – 0x04 - DTMF_ONLENGTH_300MS - 300 ms – 0x05 - DTMF_ONLENGTH_350MS - 350 ms – 0x06 - DTMF_ONLENGTH_SMS SMS Tx special pulse width
DTMFInterdigit-Interval	• DTMF interdigit interval Values: – 0x00 - DTMF_OFFLENGTH_60MS - 60 ms – 0x01 - DTMF_OFFLENGTH_100MS - 100 ms – 0x02 - DTMF_OFFLENGTH_150MS - 150 ms – 0x03 - DTMF_OFFLENGTH_200MS - 200 ms

8.1303.2 Field Documentation

8.1303.2.1 uint8_t voice_DTMFLengths::DTMFInterdigitInterval

8.1303.2.2 uint8_t voice_DTMFLengths::DTMFPulseWidth

8.1304 voice_getAllCallInformation Struct Reference

Data Fields

- voice_callInfo CallInfo
- uint8_t isEmpty
- uint8_t ALS

8.1304.1 Detailed Description

This structure contains information related to call state change.

Parameters

CallInfo	• See callInfo for more information.
isEmpty	• Multiparty indicator. – 0x00 - False – 0x01 - True
ALS	• Alternate Line Service line indicator. • Feature for supporting two different phone numbers on the same mobile device. – 0x00 - ALS_LINE1 - Line 1 (default) – 0x01 - ALS_LINE2 - Line 2

8.1304.2 Field Documentation

8.1304.2.1 uint8_t voice_getAllCallInformation::ALS

8.1304.2.2 voice_callInfo voice_getAllCallInformation::CallInfo

8.1304.2.3 uint8_t voice_getAllCallInformation::isEmpty

8.1305 voice_getAllCallRmtPtyName Struct Reference

Data Fields

- uint8_t callID
- voice_remotePartyName RemotePartyName

8.1305.1 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.

8.1305.2 Field Documentation

8.1305.2.1 uint8_t voice_getAllCallRmtPtyName::callID

8.1305.2.2 voice_remotePartyName voice_getAllCallRmtPtyName::RemotePartyName

8.1306 voice_getAllCallRmtPtyNum Struct Reference

Data Fields

- uint8_t callID
- voice_remotePartyNum RemotePartyNum

8.1306.1 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.

8.1306.2 Field Documentation

8.1306.2.1 uint8_t voice_getAllCallRmtPtyNum::callID

8.1306.2.2 voice_remotePartyNum voice_getAllCallRmtPtyNum::RemotePartyNum

8.1307 voice_getCallFWExtInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_callFWExtInfo CallFWExtInfo [20]

8.1307.1 Detailed Description

This structure contains an array of Call Forwarded Extended Information.

Parameters

numInstances	• Number of callFWExtInfo that follow. • If zero(0) then no further information exists.
CallFWExtInfo[-VOICE_MAX_N-O_OF_CALLS]	• Array of CallFWExtInfo. – See voice_CallFWExtInfo for more information.

8.1307.2 Field Documentation

8.1307.2.1 voice_callFWExtInfo voice_getCallFWExtInfo::CallFWExtInfo[20]

8.1307.2.2 uint8_t voice_getCallFWExtInfo::numInstances

8.1308 voice_getCallFWInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_callFWInfo CallFWInfo [20]

8.1308.1 Detailed Description

This structure contains an array of Call Forwarded Information.

Parameters

numInstances	• Number of callFWInfo that follow. • If zero(0) then no further information exists.
CallFWInfo[VOICE_MAX_NO_OF_CALLS]	• Array of callFWInfo. – See voice_callFWInfo for more information.

8.1308.2 Field Documentation

8.1308.2.1 voice_callFWInfo voice_getCallFWInfo::CallFWInfo[20]

8.1308.2.2 uint8_t voice_getCallFWInfo::numInstances

8.1309 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]

8.1309.1 Detailed Description

This structure contains information for Connected Peer Numbers.

Parameters

callID	• Unique call identifier for the call.
numPI	• Number presentation indicator. – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
numSI	• Number screening indicator. – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
numType	• Number type. – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
numPlan	• Number plan. – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 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.

8.1309.2 Field Documentation

8.1309.2.1 uint8_t voice_peerNumberInfo::callID

8.1309.2.2 uint8_t voice_peerNumberInfo::number[81]

8.1309.2.3 uint8_t voice_peerNumberInfo::numLen

8.1309.2.4 uint8_t voice_peerNumberInfo::numPI

8.1309.2.5 uint8_t voice_peerNumberInfo::numPlan

8.1309.2.6 uint8_t voice_peerNumberInfo::numSI

8.1309.2.7 uint8_t voice_peerNumberInfo::numType

8.1310 voice_prefVoiceSO Struct Reference

Data Fields

- uint8_t namID
- uint8_t evrcCapability
- uint16_t homePageVoiceSO
- uint16_t homeOrigVoiceSO
- uint16_t roamOrigVoiceSO

8.1310.1 Detailed Description

This structure contains information about the Preferred Voice Service Options.

Parameters

namID	• Index of the NAM(Number Assignment Module) to be configured. • Range 0 to 3. • Some modems support only 1 or 2 NAMs. • 0xFF,if not available.
evrcCapability	• EVRC capability. • Values: – 0x00 - Disable – 0x01 - Enable – 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. • Values: – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 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. • Values: – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 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. • Values: – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 0xFFFF - Not Available

8.1310.2 Field Documentation

8.1310.2.1 uint8_t voice_prefVoiceSO::evrcCapability

8.1310.2.2 uint16_t voice_prefVoiceSO::homeOrigVoiceSO

8.1310.2.3 uint16_t voice_prefVoiceSO::homePageVoiceSO

8.1310.2.4 uint8_t voice_prefVoiceSO::namID

8.1310.2.5 uint16_t voice_prefVoiceSO::roamOrigVoiceSO

8.1311 voice_remotePartyName Struct Reference

Data Fields

- uint8_t namePI
- uint8_t codingScheme
- uint8_t nameLen
- uint8_t callerName [255]

8.1311.1 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. – 0x00 - PRESENTATION_NAME_PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_NAME_PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NAME_UNAVAILABLE - Unavailable presentation – 0x03 - PRESENTATION_NAME_NAME_PRESENTATION_RESTRICTED - Restricted name presentation – 0xFF - Not Available
codingScheme	• Refer to Table10 qaGobiApiTableCodingScheme.h for coding schemes • 0xFF - Not Available
nameLen	• Provides the length of name which follow. • If zero(0) then no further information exists.
callerName[MAX_VOICE_DESCRIPTION_LENGTH]	• Name in ASCII, NULL ending.

8.1311.2 Field Documentation

8.1311.2.1 uint8_t voice_remotePartyName::callerName[255]

8.1311.2.2 uint8_t voice_remotePartyName::codingScheme

8.1311.2.3 uint8_t voice_remotePartyName::nameLen

8.1311.2.4 uint8_t voice_remotePartyName::namePI

8.1312 voice_remotePartyNum Struct Reference

Data Fields

- uint8_t presentationInd
- uint8_t numLen
- uint8_t remPartyNumber [81]

8.1312.1 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. – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
numLen	• Provides the length of number which follow. • If zero(0) then no further information exists.
remParty-Number[MAX_-CALL_NO_LEN]	• Array of numbers in ASCII, NULL ending.

8.1312.2 Field Documentation

8.1312.2.1 uint8_t voice_remotePartyNum::numLen

8.1312.2.2 uint8_t voice_remotePartyNum::presentationInd

8.1312.2.3 uint8_t voice_remotePartyNum::remPartyNumber[81]

8.1313 voice_roamTimer Struct Reference

Data Fields

- uint8_t namID
- uint32_t roamTimerValue

8.1313.1 Detailed Description

This structure contains information about the Roam Timer.

Parameters

namID	• Index of the NAM(Number Assignment Module) to be configured. • Range 0 to 3. • Some modems support only 1 or 2 NAMs. • 0xFF,if not available.
roamTimerValue	• Time in minutes. • Cumulative air time is slammed. • 0xFFFFFFFF,if not available.

8.1313.2 Field Documentation

8.1313.2.1 uint8_t voice_roamTimer::namID

8.1313.2.2 uint32_t voice_roamTimer::roamTimerValue

8.1314 voice_USSInfo Struct Reference

Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

8.1314.1 Detailed Description

This structure contains USS Information

Parameters

ussDCS	• 1 - ASCII coding scheme • 2 - 8-BIT coding scheme • 3 - UCS2
ussLen	• Range 1 to 182
ussData	• Data encoded as per the DCS

8.1314.2 Field Documentation

8.1314.2.1 uint8_t voice_USSInfo::ussData[182]

8.1314.2.2 uint8_t voice_USSInfo::ussDCS

8.1314.2.3 uint8_t voice_USSInfo::ussLen

8.1315 voice_UUSInfo Struct Reference

Data Fields

- uint8_t UUSType
- uint8_t UUSDcs
- uint8_t UUSDatalen
- uint8_t UUSData [255]

8.1315.1 Detailed Description

This structure contains User to User Signaling Service Information.

Parameters

UUSType	• UUS type values are: – 0x00 - UUS_DATA – 0x01 - UUS_TYPE1_IMPLICIT – 0x02 - UUS_TYPE1_REQUIRED – 0x03 - UUS_TYPE1_NOT_REQUIRED – 0x04 - UUS_TYPE2_REQUIRED – 0x05 - UUS_TYPE2_NOT_REQUIRED – 0x06 - UUS_TYPE3_REQUIRED – 0x07 - UUS_TYPE3_NOT_REQUIRED – 0xFF - Not Available
UUSDcs	• UUS data coding scheme values are: – 0x01 - UUS_DCS_USP – 0x02 - UUS_DCS_OHLP – 0x03 - UUS_DCS_X244 – 0x04 - UUS_DCS_SMCF – 0x05 - UUS_DCS_IA5 – 0x06 - UUS_DCS_RV12RD – 0x07 - UUS_DCS_Q931UNCCM – 0xFF - Not Available
UUSDatalen	• Number of sets of the following elements. – UUSData • If zero(0) then no further information exists.
UUSData[MAX_VOICE_DESCRIPTION_LENGTH]	• UUS data encoded as per coding scheme

8.1315.2 Field Documentation

8.1315.2.1 uint8_t voice_UUSInfo::UUSData[255]

8.1315.2.2 uint8_t voice_UUSInfo::UUSDatalen

8.1315.2.3 uint8_t voice_UUSInfo::UUSDcs

8.1315.2.4 uint8_t voice_UUSInfo::UUSType

8.1316 voiceALSSelectLineInfo Struct Reference

Data Fields

- BYTE lineValue

8.1316.1 Detailed Description

This structure contains ALS Select Line Information Parameters.

Parameters

lineValue	• ALS Line Value. – 0x00 - ALS_LINE1 - Line 1 (default) – 0x01 - ALS_LINE2 - Line 2

8.1316.2 Field Documentation

8.1316.2.1 BYTE voiceALSSelectLineInfo::lineValue

8.1317 voiceALSSetLineSwitchInfo Struct Reference

Data Fields

- BYTE switchOption

8.1317.1 Detailed Description

This structure contains ALS Set Line Switching Information Parameters.

Parameters

switchOption	• Switch Option. – 0x00 - VOICE_LINE_SWITCHING_NOT_ALLOWED - Line switching is not allowed – 0x01 - VOICE_LINE_SWITCHING_ALLOWED - Line switching is allowed

8.1317.2 Field Documentation

8.1317.2.1 BYTE voiceALSSetLineSwitchInfo::switchOption

8.1318 voiceAnswerCall Struct Reference

Data Fields

- BYTE * pCallId

8.1318.1 Detailed Description

Contains the parameters passed for SLQSVoiceAnswerCall.

Parameters

pCallId[IN/OUT]	• Unique call identifier for the call that must be answered.

8.1318.2 Field Documentation

8.1318.2.1 BYTE* voiceAnswerCall::pCallId

8.1319 voiceBindSubscriptionInfo Struct Reference

Data Fields

- BYTE subsType

8.1319.1 Detailed Description

This structure contains Bind Subscription Information Parameters.

Parameters

subsType	• Subscription Type. – 0x00 - VOICE_SUBS_TYPE_PRIMARY - Primary – 0x01 - VOICE_SUBS_TYPE_SECONDARY - Secondary

8.1319.2 Field Documentation

8.1319.2.1 BYTE voiceBindSubscriptionInfo::subsType

8.1320 voiceBurstDTMFInfo Struct Reference

Data Fields

- burstDTMFInfo BurstDTMFInfo
- DTMFLengths * pBurstDTMFLengths

8.1320.1 Detailed Description

This structure contains parameters of burst Dual-Tone Multifrequency (DTMF)

Parameters

BurstDTMFInfo	• Burst DTMF Information – See burstDTMFInfo for more information
pBurstDTMF-Lengths	[optional] • DTMF Lengths – See DTMFLengths for more information

8.1320.2 Field Documentation

8.1320.2.1 burstDTMFInfo voiceBurstDTMFInfo::BurstDTMFInfo

8.1320.2.2 DTMFLengths* voiceBurstDTMFInfo::pBurstDTMFLengths

8.1321 voiceCallInfoReq Struct Reference

Data Fields

- BYTE callID

8.1321.1 Detailed Description

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

Parameters

callID	• Call identifier for the call queried for information.

8.1321.2 Field Documentation

8.1321.2.1 BYTE voiceCallInfoReq::callID

8.1322 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

8.1322.1 Detailed Description

This structure contains information of the response parameters associated with a call.

Parameters

pCall-Info(optional)	• See callInfo for more information.
pRemoteParty-Num(optional)	• See remotePartyNum for more information.
pSrvOpt	• Service option(optional) • Applicable only for 3GPP2 devices. • See Table8 qaGobiApiTableServiceOptions.h for standard service option number assignments.
pVoicePrivacy	• Voice Privacy.(optional) • Applicable only for 3GPP2 devices. • Values. – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available

pOTASPStatus	• OTASP status for the OTASP call.(optional) • Applicable only for 3GPP2 devices. – 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP – 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA) – 0xFF - Not Available
pRemoteParty-Name(optional)	• Applicable only for 3GPP devices. • See remotePartyName for more information.
pUUS-Info(optional)	• Applicable only for 3GPP devices. • See UUSInfo for more information.
pAlert-Type(optional)	• Alerting type. • Applicable only for 3GPP devices. – 0x00 - ALERTING_LOCAL - Local – 0x01 - ALERTING_REMOTE - Remote – 0xFF - Not Available
pAlphaID-Info(optional)	• Applicable only for 3GPP devices. • See alphaIDInfo for more information.
pConnectNum-Info(optional)	• See connectNumInfo for more information.

pDiag-Info(optional)	• See diagInfo for more information.
pAlertingPattern	• Alerting pattern.(optional) – 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1 – 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2 – 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3 – 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5 – 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6 – 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7 – 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8 – 0x08 - QMI_VOICE_ALERTING_PATTERN_9 - Pattern 9 – 0xFF - Not Available

8.1322.2 Field Documentation

8.1322.2.1 ULONG* voiceCallInfoResp::pAlertingPattern

8.1322.2.2 BYTE* voiceCallInfoResp::pAlertType

8.1322.2.3 alphaIDInfo* voiceCallInfoResp::pAlphaIDInfo

8.1322.2.4 callInfo* voiceCallInfoResp::pCallInfo

8.1322.2.5 connectNumInfo* voiceCallInfoResp::pConnectNumInfo

8.1322.2.6 diagInfo* voiceCallInfoResp::pDiagInfo

8.1322.2.7 BYTE* voiceCallInfoResp::pOTASPStatus

8.1322.2.8 remotePartyName* voiceCallInfoResp::pRemotePartyName

8.1322.2.9 remotePartyNum* voiceCallInfoResp::pRemotePartyNum

8.1322.2.10 WORD* voiceCallInfoResp::pSrvOpt

8.1322.2.11 UUSInfo* voiceCallInfoResp::pUUSInfo

8.1322.2.12 BYTE* voiceCallInfoResp::pVoicePrivacy

8.1323 voiceCallRequestParams Struct Reference

Data Fields

- BYTE callNumber [81]
- BYTE * pCallType
- BYTE * pCLIRType
- UUSInfo * pUUSInFo
- CUGInfo * pCUGInfo
- BYTE * pEmergencyCategory

- calledPartySubAdd * pCallPartySubAdd
- ULONG * pSvcType

8.1323.1 Detailed Description

This structure contains Voice Call Request Parameters

Parameters

callNumber[81]	• Number to be dialed in ASCII string, NULL terminated. • 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: – 0x00 - CALL_TYPE_VOICE - Voice (automatic selection) – 0x01 - CALL_TYPE_VOICE_FORCED - Avoid modem call classification – 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP* – 0x09 - CALL_TYPE_EMERGENCY - Emergency
pCLIRType(optional)	• CLIR type values are: – 0x01 - CLIR_SUPPRESSION - Suppression – 0x02 - CLIR_INVOCATION - Invocation
pUUSInFo(optional)	• Pointer to structure of UUSInfo – See UUSInfo for more information
pCUGInfo(optional)	• Pointer to structure of CUGInfo – See CUGInfo for more information
pEmergencyCategory(optional)	• Bit mask of emergency number categories. This is only applicable when the call type is set to Emergency. – Bit 0 - VOICE_EMER_CAT_POLICE_BIT - Police – Bit 1 - VOICE_EMER_CAT_AMBULANCE_BIT - Ambulance – Bit 2 - VOICE_EMER_CAT_FIRE_BRIGADE_BIT- Fire brigade – Bit 3 - VOICE_EMER_CAT_MARINE_GUARD_BIT - Marine guard – Bit 4 - VOICE_EMER_CAT_MOUNTAIN_RESCUE_BIT - Mountain rescue – Bit 5 - VOICE_EMER_CAT_MANUAL_ECALL_BIT - Manual emergency call – Bit 6 - VOICE_EMER_CAT_AUTO_ECALL_BIT - Automatic emergency call – Bit 7 - VOICE_EMER_CAT_SPARE_BIT - Spare bit
pCallPartySubAdd(optional)	• Pointer to structure of calledPartySubAdd – See calledPartySubAdd for more information

pSvc-Type(optional)	• Service Type. – 0x01 - VOICE_DIAL_CALL_SRV_TYPE_AUTOMATIC - Automatic – 0x02 - VOICE_DIAL_CALL_SRV_TYPE_GSM - GSM – 0x03 - VOICE_DIAL_CALL_SRV_TYPE_WCDMA - WCDMA – 0x04 - VOICE_DIAL_CALL_SRV_TYPE_CDMA_AUTOMATIC - CDMA automatic – 0x05 - VOICE_DIAL_CALL_SRV_TYPE_GSM_WCDMA - GSM or WCDMA – 0x06 - VOICE_DIAL_CALL_SRV_TYPE_LTE -LTE

8.1323.2 Field Documentation

8.1323.2.1 BYTE voiceCallRequestParams::callNumber[81]

8.1323.2.2 calledPartySubAdd* voiceCallRequestParams::pCallPartySubAdd

8.1323.2.3 BYTE* voiceCallRequestParams::pCallType

8.1323.2.4 BYTE* voiceCallRequestParams::pCLIRType

8.1323.2.5 CUGInfo* voiceCallRequestParams::pCUGInfo

8.1323.2.6 BYTE* voiceCallRequestParams::pEmergencyCategory

8.1323.2.7 ULONG* voiceCallRequestParams::pSvcType

8.1323.2.8 UUSInfo* voiceCallRequestParams::pUUSInFo

8.1324 voiceCallResponseParams Struct Reference

Data Fields

- BYTE * pCallID
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- ccSUPSType * pCCSUPSType

8.1324.1 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 – See alphaIDInfo for more information

pCCResult-Type(optional)	• Call Control Result Type. – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service
pCCSUPS-Type(optional)	• Pointer to structure of ccSUPSType • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

8.1324.2 Field Documentation

8.1324.2.1 alphaIDInfo∗ voiceCallResponseParams::pAlphaIDInfo

8.1324.2.2 BYTE∗ voiceCallResponseParams::pCallID

8.1324.2.3 BYTE∗ voiceCallResponseParams::pCCResultType

8.1324.2.4 ccSUPSType∗ voiceCallResponseParams::pCCSUPSType

8.1325 voiceContDTMFinfo Struct Reference

Data Fields

- BYTE ∗ pCallID
- BYTE DTMFdigit

8.1325.1 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. Start continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • If the call ID value received is 0, no value has been returned by the device
DTMFdigit[IN]	• DTMF digit in ASCII.

8.1325.2 Field Documentation

8.1325.2.1 BYTE voiceContDTMFinfo::DTMFdigit

8.1325.2.2 BYTE* voiceContDTMFinfo::pCallID

8.1326 voiceDTMFEventInfo Struct Reference

Data Fields

- DTMFInfo DTMFInformation
- BYTE * pOnLength
- BYTE * pOffLength

8.1326.1 Detailed Description

This structure contains the parameters passed for SLQSVoiceSetDTMFEventCallBack by the device.

Parameters

DTMFInformation(mandatory)	See DTMFInfo for more information.
pOnLength(optional)	• DTMF Pulse Width – 0x00 - DTMF_ONLENGTH_95MS - 95 ms – 0x01 - DTMF_ONLENGTH_150MS - 150 ms – 0x02 - DTMF_ONLENGTH_200MS - 200 ms – 0x03 - DTMF_ONLENGTH_250MS - 250 ms – 0x04 - DTMF_ONLENGTH_300MS - 300 ms – 0x05 - DTMF_ONLENGTH_350MS - 350 ms – 0x06 - DTMF_ONLENGTH_SMS - SMS Tx special pulse width
pOffLength(optional)	• DTMF Interdigit Interval – 0x00 - DTMF_OFFLENGTH_60MS - 60 ms – 0x01 - DTMF_OFFLENGTH_100MS - 100 ms – 0x02 - DTMF_OFFLENGTH_150MS - 150 ms – 0x03 - DTMF_OFFLENGTH_200MS - 200 ms

Note

None

8.1326.2 Field Documentation

8.1326.2.1 DTMFInfo voiceDTMFEventInfo::DTMFInformation

8.1326.2.2 BYTE* voiceDTMFEventInfo::pOffLength

8.1326.2.3 BYTE* voiceDTMFEventInfo::pOnLength

8.1327 voiceFlashInfo Struct Reference

Data Fields

- BYTE * pCallID
- BYTE * pFlashPayLd
- BYTE * pFlashType

8.1327.1 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. • Variable Length, NULL terminated.
pFlashType[IN](optional)	• Flash type. – 0 - Simple Flash (default) – 1 - Activate answer hold – 2 - Deactivate answer hold

8.1327.2 Field Documentation

8.1327.2.1 BYTE* voiceFlashInfo::pCallID

8.1327.2.2 BYTE* voiceFlashInfo::pFlashPayLd

8.1327.2.3 BYTE* voiceFlashInfo::pFlashType

8.1328 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

8.1328.1 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.
pOTASPStatus(optional)	• OTASP status for the OTASP call. • Applicable only for 3GPP2 devices. – 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP – 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA) – 0xFF - Not Available

pVoicePrivacy(optional)	• Voice Privacy. • Values. – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available
pArrCallEndReason(optional)	• See arrCallEndReason for more information.
pArrAlphaID(optional)	• See arrAlphaID for more information.
pArrConnectPartyNum(optional)	• See arrConnectPartyNum for more information.
pArrDiagInfo(optional)	• See arrDiagInfo for more information.
pArrCalledPartyNum(optional)	• See arrCalledPartyNum for more information.
pArrRedirPartyNum(optional)	• See arrRedirPartyNum for more information.
pArrAlertingPattern(optional)	• See arrAlertingPattern for more information.

8.1328.2 Field Documentation

8.1328.2.1 arrAlertingPattern∗ voiceGetAllCallInfo::pArrAlertingPattern

8.1328.2.2 arrAlertingType∗ voiceGetAllCallInfo::pArrAlertingType

8.1328.2.3 arrAlphaID∗ voiceGetAllCallInfo::pArrAlphaID

8.1328.2.4 arrCalledPartyNum∗ voiceGetAllCallInfo::pArrCalledPartyNum

8.1328.2.5 arrCallEndReason∗ voiceGetAllCallInfo::pArrCallEndReason

8.1328.2.6 arrCallInfo∗ voiceGetAllCallInfo::pArrCallInfo

8.1328.2.7 arrConnectPartyNum∗ voiceGetAllCallInfo::pArrConnectPartyNum

8.1328.2.8 arrDiagInfo∗ voiceGetAllCallInfo::pArrDiagInfo

8.1328.2.9 arrRedirPartyNum∗ voiceGetAllCallInfo::pArrRedirPartyNum

8.1328.2.10 arrRemotePartyName∗ voiceGetAllCallInfo::pArrRemotePartyName

8.1328.2.11 arrRemotePartyNum∗ voiceGetAllCallInfo::pArrRemotePartyNum

8.1328.2.12 arrSvcOption* voiceGetAllCallInfo::pArrSvcOption

8.1328.2.13 arrUUSInfo* voiceGetAllCallInfo::pArrUUSInfo

8.1328.2.14 BYTE* voiceGetAllCallInfo::pOTASPStatus

8.1328.2.15 BYTE* voiceGetAllCallInfo::pVoicePrivacy

8.1329 voiceGetCallBarringReq Struct Reference

Data Fields

- BYTE reason
- BYTE * pSvcClass

8.1329.1 Detailed Description

This structure contains Voice Get Call Barring Request Parameters

Parameters

reason	• Call Barring Reason • Values: – 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
pSvcClass[IN/OUT]	• Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available

8.1329.2 Field Documentation

8.1329.2.1 BYTE* voiceGetCallBarringReq::pSvcClass

8.1329.2.2 BYTE voiceGetCallBarringReq::reason

8.1330 voiceGetCallBarringResp Struct Reference

Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1330.1 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) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1330.2 Field Documentation

8.1330.2.1 alphaIDInfo* voiceGetCallBarringResp::pAlphaIDInfo

8.1330.2.2 BYTE* voiceGetCallBarringResp::pCallID

8.1330.2.3 BYTE* voiceGetCallBarringResp::pCCResType

8.1330.2.4 ccSUPSType* voiceGetCallBarringResp::pCCSUPSType

8.1330.2.5 WORD* voiceGetCallBarringResp::pFailCause

8.1330.2.6 BYTE* voiceGetCallBarringResp::pSvcClass

8.1331 voiceGetCallFWReq Struct Reference

Data Fields

- BYTE Reason
- BYTE * pSvcClass

8.1331.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Request Parameters

Parameters

Reason	• Call Forwarding Reason • Values: – 0x01 - QMI_VOICE_REASON_FWDREASON_UNCONDITIONAL - Unconditional call forwarding – 0x02 - QMI_VOICE_REASON_FWDREASON_MOBILEBUSY - Forward when the mobile is busy – 0x03 - QMI_VOICE_REASON_FWDREASON_NOREPLY - Forward when there is no reply – 0x04 - QMI_VOICE_REASON_FWDREASON_UNREACHABLE - Forward when the call is unreachable – 0x05 - QMI_VOICE_REASON_FWDREASON_ALLFORWARDING - All forwarding – 0x06 - QMI_VOICE_REASON_FWDREASON_ALLCONDITIONAL - All conditional forwarding
pSvcClass(optional)	• Service Class is a combination (sum) of information class constants • See qaGobiApiTableSupServiceInfoClasses.h for service classes.

8.1331.2 Field Documentation

8.1331.2.1 BYTE* voiceGetCallFWReq::pSvcClass

8.1331.2.2 BYTE voiceGetCallFWReq::Reason

8.1332 voiceGetCallFWResp Struct Reference

Data Fields

- getCallFWInfo * pGetCallFWInfo
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType
- getCallFWExtInfo * pGetCallFWExtInfo

8.1332.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Response Parameters

Parameters

pGetCallFWInfo	• Pointer to structure of getCallFWInfo (optional) – See getCallFWInfo for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information
pGetCallFWExt-Info	• Pointer to structure of getCallFWExtInfo (optional) – See getCallFWExtInfo for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1332.2 Field Documentation

8.1332.2.1 alphaIDInfo* voiceGetCallFWResp::pAlphaIDInfo

8.1332.2.2 BYTE* voiceGetCallFWResp::pCallID

8.1332.2.3 BYTE* voiceGetCallFWResp::pCCResType

8.1332.2.4 ccSUPSType* voiceGetCallFWResp::pCCSUPSType

8.1332.2.5 WORD* voiceGetCallFWResp::pFailCause

8.1332.2.6 getCallFWExtInfo* voiceGetCallFWResp::pGetCallFWExtInfo

8.1332.2.7 getCallFWInfo* voiceGetCallFWResp::pGetCallFWInfo

8.1333 voiceGetCallWaitInfo Struct Reference

Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1333.1 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) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available
pFailCause[OUT]	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo[OUT]	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information

pCCResType[OUT]	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID[OUT]	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType[OUT]	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1333.2 Field Documentation

8.1333.2.1 alphaIDInfo* voiceGetCallWaitInfo::pAlphaIDInfo

8.1333.2.2 BYTE* voiceGetCallWaitInfo::pCallID

8.1333.2.3 BYTE* voiceGetCallWaitInfo::pCCResType

8.1333.2.4 ccSUPSType* voiceGetCallWaitInfo::pCCSUPSType

8.1333.2.5 WORD* voiceGetCallWaitInfo::pFailCause

8.1333.2.6 BYTE* voiceGetCallWaitInfo::pSvcClass

8.1334 voiceGetCLIPResp Struct Reference

Data Fields

- CLIPResp * pCLIPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1334.1 Detailed Description

This structure contains Voice Get Calling Line Identification Presentation(CLIP) Response Parameters

Parameters

pCLIPResp	• Pointer to structure of CLIPResp (optional) – See CLIPResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1334.2 Field Documentation

8.1334.2.1 alphaIDInfo* voiceGetCLIPResp::pAlphaIDInfo

8.1334.2.2 BYTE* voiceGetCLIPResp::pCallID

8.1334.2.3 BYTE* voiceGetCLIPResp::pCCResType

8.1334.2.4 ccSUPSType* voiceGetCLIPResp::pCCSUPSType

8.1334.2.5 CLIPResp* voiceGetCLIPResp::pCLIPResp

8.1334.2.6 WORD* voiceGetCLIPResp::pFailCause

8.1335 voiceGetCLIRResp Struct Reference

Data Fields

- CLIRResp * pCLIRResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1335.1 Detailed Description

This structure contains Voice Get Calling Line Identification Restriction (CLIR) Response Parameters

Parameters

pCLIRResp	• Pointer to structure of CLIRResp (optional) – See CLIRResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1335.2 Field Documentation

8.1335.2.1 alphaIDInfo* voiceGetCLIRResp::pAlphaIDInfo

8.1335.2.2 BYTE* voiceGetCLIRResp::pCallID

8.1335.2.3 BYTE* voiceGetCLIRResp::pCCResType

8.1335.2.4 ccSUPSType* voiceGetCLIRResp::pCCSUPSType

8.1335.2.5 CLIRResp* voiceGetCLIRResp::pCLIRResp

8.1335.2.6 WORD* voiceGetCLIRResp::pFailCause

8.1336 voiceGetCNAPResp Struct Reference

Data Fields

- CNAPResp * pCNAPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1336.1 Detailed Description

This structure contains Voice Get Calling Name Presentation(CNAP) Response Parameters

Parameters

pCNAPResp	• Pointer to structure of CNAPResp (optional) – See CNAPResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.

pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1336.2 Field Documentation

8.1336.2.1 alphaIDInfo* voiceGetCNAPResp::pAlphaIDInfo

8.1336.2.2 BYTE* voiceGetCNAPResp::pCallID

8.1336.2.3 BYTE* voiceGetCNAPResp::pCCResType

8.1336.2.4 ccSUPSType* voiceGetCNAPResp::pCCSUPSType

8.1336.2.5 CNAPResp* voiceGetCNAPResp::pCNAPResp

8.1336.2.6 WORD* voiceGetCNAPResp::pFailCause

8.1337 voiceGetCOLPResp Struct Reference

Data Fields

- COLPResp * pCOLPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1337.1 Detailed Description

This structure contains Voice Get Connected Line Identification Presentation(COLP) Response Parameters

Parameters

pCOLPResp	• Pointer to structure of COLPResp (optional) – See COLPResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information

pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1337.2 Field Documentation

8.1337.2.1 alphaIDInfo* voiceGetCOLPResp::pAlphaIDInfo

8.1337.2.2 BYTE* voiceGetCOLPResp::pCallID

8.1337.2.3 BYTE* voiceGetCOLPResp::pCCResType

8.1337.2.4 ccSUPSType* voiceGetCOLPResp::pCCSUPSType

8.1337.2.5 COLPResp* voiceGetCOLPResp::pCOLPResp

8.1337.2.6 WORD* voiceGetCOLPResp::pFailCause

8.1338 voiceGetCOLRResp Struct Reference

Data Fields

- COLRResp * pCOLRResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1338.1 Detailed Description

This structure contains Voice Get Connected Line Identification Restriction(COLR) Response Parameters

Parameters

pCOLRResp	• Pointer to structure of COLRResp (optional) – See COLRResp for more information
pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1338.2 Field Documentation

8.1338.2.1 alphaIDInfo* voiceGetCOLRResp::pAlphaIDInfo

8.1338.2.2 BYTE* voiceGetCOLRResp::pCallID

8.1338.2.3 BYTE* voiceGetCOLRResp::pCCResType

8.1338.2.4 ccSUPSType* voiceGetCOLRResp::pCCSUPSType

8.1338.2.5 COLRResp* voiceGetCOLRResp::pCOLRResp

8.1338.2.6 WORD* voiceGetCOLRResp::pFailCause

8.1339 voiceGetConfigReq Struct Reference

Data Fields

- BYTE * pAutoAnswer
- BYTE * pAirTimer
- BYTE * pRoamTimer
- BYTE * pTTYMode
- BYTE * pPrefVoiceSO
- BYTE * pAMRStatus
- BYTE * pPrefVoicePrivacy
- BYTE * pNamID
- BYTE * pVoiceDomainPref

8.1339.1 Detailed Description

This structure contains Voice Get Configuration Request Parameters

Parameters

pAutoAnswer(optional)	• Indicator to retrieve the Auto Answer Information. – 0x01 - Include auto answer information
pAirTimer(optional)	• Indicator to retrieve the Air Timer Information. – 0x01 - Include air calls timer count information • Currently Not Supported.
pRoamTimer(optional)	• Indicator to retrieve the Roam Timer Information. – 0x01 - Include roam calls timer information • Currently Not Supported.
pTTYMode(optional)	• Indicator to retrieve the TTY Mode Information. – 0x01 - Include TTY configuration status information
pPrefVoiceSO(optional)	• Indicator to retrieve the Preferred Voice SO Information. – 0x01 - Include preferred voice configuration status information • Currently Not Supported.
pAMRStatus(optional)	• Indicator to retrieve the AMR Status Information. – 0x01 - Include AMR status information
pPrefVoicePrivacy(optional)	• Indicator to retrieve the Preferred Voice Privacy Information. – 0x01 - Include preferred voice privacy status information

pNamI-D(optional)	• Index of the Number Assignment Module Index (CDMA subscription) to be configured • Range: 0 to 3. • Some modems support only 1 or 2 NAMs. • The NAM Index is valid only when the request contains at least one of Air Timer, Roam Timer, and Preferred Voice SO. • 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. – 0x01 - Include voice domain preference information

Note

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

8.1339.2 Field Documentation

8.1339.2.1 BYTE* voiceGetConfigReq::pAirTimer

8.1339.2.2 BYTE* voiceGetConfigReq::pAMRStatus

8.1339.2.3 BYTE* voiceGetConfigReq::pAutoAnswer

8.1339.2.4 BYTE* voiceGetConfigReq::pNamID

8.1339.2.5 BYTE* voiceGetConfigReq::pPrefVoicePrivacy

8.1339.2.6 BYTE* voiceGetConfigReq::pPrefVoiceSO

8.1339.2.7 BYTE* voiceGetConfigReq::pRoamTimer

8.1339.2.8 BYTE* voiceGetConfigReq::pTTYMode

8.1339.2.9 BYTE* voiceGetConfigReq::pVoiceDomainPref

8.1340 voiceGetConfigResp Struct Reference

Data Fields

- BYTE * pAutoAnswerStat
- airTimer * pAirTimerCnt
- roamTimer * pRoamTimerCnt
- BYTE * pCurrTTYMode
- prefVoiceSO * pCurPrefVoiceSO
- curAMRConfig * pCurAMRConfig
- BYTE * pCurVoicePrivacyPref
- BYTE * pCurVoiceDomainPref

8.1340.1 Detailed Description

This structure contains Voice Get Configuration Response Parameters.

Parameters

pAutoAnswer-Stat(optional)	• Auto Answer Status • Value returned is read from NV_AUTO_ANSWER_I. – 0x00 - Disabled – 0x01 - Enabled – 0xFF - Not Available
pAirTimer-Cnt(optional)	• Air Timer Count • Value returned is read from NV_AIR_CNT_I. • See airTimer for more information
pRoamTimer-Cnt(optional)	• Roam Timer Count • Value returned is read from NV_ROAM_CNT_I. • See roamTimer for more information
pCurrTTY-Mode(optional)	• Current TTY Mode • Value returned is read from NV_TTY_I. – 0x00 - TTY_MODE_FULL - Full – 0x01 - TTY_MODE_VCO - Voice carry over – 0x02 - TTY_MODE_HCO - Hearing carry over – 0x03 - TTY_MODE_OFF - Off – 0xFF - Not Available
pCurPrefVoiceS-O(optional)	• Current Preferred Voice SO • Value returned is read from NV_PREF_VOICE_SO_I. • See prefVoiceSO for more information
pCurAMR-Config(optional)	• Current Adaptive Multi-Rate Configuration. • Values returned are read from NV_GSM_ARM_CALL_CONFIG_I and NV_UMTS_AMR_CODEC_PREFERENCE_CONFIG_I. • See curAMRConfig for more information
pCurVoice-Privacy-Pref(optional)	• Current Voice Privacy Preference • Value returned is read from NV_VOICE_PRIV_I. – 0x00 - Standard privacy – 0x01 - Enhanced privacy – 0xFF - Not Available

pCurVoice-Domain-Pref(optional)	• Current Voice Domain Preference. – 0x00 - Circuit-switched (CS) only – 0x01 - Packet-switched (PS) only – 0x02 - CS is preferred; PS is secondary – 0x03 - PS is preferred; CS is secondary – 0xFF - Not Available

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1340.2 Field Documentation

8.1340.2.1 airTimer∗ voiceGetConfigResp::pAirTimerCnt

8.1340.2.2 BYTE∗ voiceGetConfigResp::pAutoAnswerStat

8.1340.2.3 curAMRConfig∗ voiceGetConfigResp::pCurAMRConfig

8.1340.2.4 prefVoiceSO∗ voiceGetConfigResp::pCurPrefVoiceSO

8.1340.2.5 BYTE∗ voiceGetConfigResp::pCurrTTYMode

8.1340.2.6 BYTE∗ voiceGetConfigResp::pCurVoiceDomainPref

8.1340.2.7 BYTE∗ voiceGetConfigResp::pCurVoicePrivacyPref

8.1340.2.8 roamTimer∗ voiceGetConfigResp::pRoamTimerCnt

8.1341 voiceIndicationRegisterInfo Struct Reference

Data Fields

- BYTE ∗ pRegDTMFEvents
- BYTE ∗ pRegVoicePrivacyEvents
- BYTE ∗ pSuppsNotifEvents

8.1341.1 Detailed Description

This structure contains parameters of Indication Register Information

Parameters

pRegDTMF-Events(optional)	• Registration Indication For DTMF Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_DTMF_IND indication. – 0x00 - Disable – 0x01 - Enable

pRegVoice-Privacy-Events(optional)	• Registration Indication For Voice Privacy Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_PRIVACY_IND indication. – 0x00 - Disable – 0x01 - Enable
pSuppsNotif-Events(optional)	• Registration Indication For Supplementary Service Notification Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_SUPS_NOTIFICATION_IND indication. – 0x00 - Disable – 0x01 - Enable

Note

One of the optional parameter is mandatory to be present in the request.

8.1341.2 Field Documentation

8.1341.2.1 BYTE* voiceIndicationRegisterInfo::pRegDTMFEvents

8.1341.2.2 BYTE* voiceIndicationRegisterInfo::pRegVoicePrivacyEvents

8.1341.2.3 BYTE* voiceIndicationRegisterInfo::pSuppsNotifEvents

8.1342 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

8.1342.1 Detailed Description

This structure contains Voice record Information

Parameters

callID	[Mandatory] • Call identifier for the call.
pSignalInfo[-Optional]	• Signal Information • See signalInfo for more information
pCallerIDInfo[-Optional]	• Caller ID Information • 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 • see connectNumInfo for more information
pCallingParty-Info[Optional]	• Calling Party Number Information • This structure is having exactly same elements as connectNumInfo • see connectNumInfo for more information
pCalledParty-Info[Optional]	• Called Party Number Information • see calledPartyInfo for more information
pRedirNumInfo[-Optional]	• Redirecting Number Information • see redirNumInfo for more information
pCLIRCause[-Optional]	• National Supplementary Services - CLIR • see NSSAudioCtrl for more information
pNSSAudioCtrl[-Optional]	• National Supplementary Services - Audio Control
pNSSRelease[-Optional]	• National Supplementary Services - Release

pLineCtrlInfo[-Optional]	• Line Control Information • see lineCtrlInfo for more information
pExtDispRec-Info[Optional]	• Extended Display Record Information • see extDispRecInfo for more information

8.1342.2 Field Documentation

8.1342.2.1 BYTE voiceInfoRec::callID

8.1342.2.2 calledPartyInfo* voiceInfoRec::pCalledPartyInfo

8.1342.2.3 callerIDInfo* voiceInfoRec::pCallerIDInfo

8.1342.2.4 BYTE* voiceInfoRec::pCallerNameInfo

8.1342.2.5 connectNumInfo* voiceInfoRec::pCallingPartyInfo

8.1342.2.6 BYTE* voiceInfoRec::pCallWaitInd

8.1342.2.7 BYTE* voiceInfoRec::pCLIRCause

8.1342.2.8 connectNumInfo* voiceInfoRec::pConnectNumInfo

8.1342.2.9 BYTE* voiceInfoRec::pDispInfo

8.1342.2.10 BYTE* voiceInfoRec::pExtDispInfo

8.1342.2.11 extDispRecInfo* voiceInfoRec::pExtDispRecInfo

8.1342.2.12 lineCtrlInfo* voiceInfoRec::pLineCtrlInfo

8.1342.2.13 NSSAudioCtrl* voiceInfoRec::pNSSAudioCtrl

8.1342.2.14 BYTE* voiceInfoRec::pNSSRelease

8.1342.2.15 redirNumInfo* voiceInfoRec::pRedirNumInfo

8.1342.2.16 signalInfo* voiceInfoRec::pSignalInfo

8.1343 voiceManageCallsReq Struct Reference

Data Fields

- BYTE SUPSType
- BYTE * pCallID

8.1343.1 Detailed Description

This structure contains Manage Calls Information.

Parameters

SUPSType	• Supplementary service type during the call. – 0x01 - SUPS_TYPE_RELEASE_HELD_OR_WAITING * Release is held or waiting – 0x02 - SUPS_TYPE_RELEASE_ACTIVE_ACCEPT_HELD_OR_WAITING * Release is active and accepting held or waiting – 0x03 - SUPS_TYPE_HOLD_ACTIVE_ACCEPT_WAITING_OR_HELD * Hold is active and accepting waiting or held – 0x04 - SUPS_TYPE_HOLD_ALL_EXCEPT_SPECIFIED_CALL * Hold all calls except a specified one – 0x05 - SUPS_TYPE_MAKE_CONFERENCE_CALL * Make a conference call – 0x06 - SUPS_TYPE_EXPLICIT_CALL_TRANSFER * Explicit call transfer – 0x07 - SUPS_TYPE_CCBS_ACTIVATION * Activate completion of calls to busy subscriber – 0x08 - SUPS_TYPE_END_ALL_CALLS * End all calls – 0x09 - SUPS_TYPE_RELEASE_SPECIFIED_CALL * Release a specified call
pCallID[Optional]	• Applicable only for SUPSType 0x04, 0x07, and 0x09

8.1343.2 Field Documentation

8.1343.2.1 BYTE* voiceManageCallsReq::pCallID

8.1343.2.2 BYTE voiceManageCallsReq::SUPSType

8.1344 voiceManageCallsResp Struct Reference

Data Fields

- WORD * pFailCause

8.1344.1 Detailed Description

This structure contains Failure cause Information. Populated when API Fails.

Parameters

pFailCause	• Supplementary service failure causes (optional, supply NULL if not required). • See Table8 qaGobiApiTableVoiceCallEndReasons.h for supplementary services failure cause – 0xFFFF is the value when the information is not received from device

8.1344.2 Field Documentation

8.1344.2.1 WORD* voiceManageCallsResp::pFailCause

8.1345 voiceOrigUSSDNoWaitInfo Struct Reference

Data Fields

- struct USSInfo USSInformation

8.1345.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

Parameters

USSInformation	• See USSInfo for more information.

8.1345.2 Field Documentation

8.1345.2.1 struct USSInfo voiceOrigUSSDNoWaitInfo::USSInformation

8.1346 voiceOTASPStatusInfo Struct Reference

Data Fields

- BYTE callID
- BYTE OTASPStatus

8.1346.1 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: – 0x00 - OTASP_STATUS_SPL_UNLOCKED.SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED. SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED.A-key exchanged; only for user-initiated O-TASP – 0x03 - OTASP_STATUS_SSD_UPDATED. SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP(OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP(OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP(OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP(OTAPA)

8.1346.2 Field Documentation

8.1346.2.1 BYTE voiceOTASPStatusInfo::callID

8.1346.2.2 BYTE voiceOTASPStatusInfo::OTASPStatus

8.1347 voicePrivacyInfo Struct Reference

Data Fields

- BYTE callID
- BYTE voicePrivacy

8.1347.1 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) – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy

Note

None

8.1347.2 Field Documentation

8.1347.2.1 BYTE voicePrivacyInfo::callID

8.1347.2.2 BYTE voicePrivacyInfo::voicePrivacy

8.1348 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

8.1348.1 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] • Array of Call Information This must be populated if Indication is received See arrCallInfo for more information. – Applicable for both "3GPP/3GPP2"
pArrRemote-PartyNum	[optional] • Array of Remote Party Name.(NULL when not present) See arrRemotePartyNum for more information. – Applicable only for "3GPP/3GPP2"

pArrRemote-PartyName	[optional] • Array of Alerting Type.(NULL when not present) See arrRemotePartyName for more information. – Applicable only for "3GPP"
pArrAlertingType	[optional] • Array of Alerting Type(NULL when not present) See arrAlertingType for more information. – Applicable only for "3GPP"
pArrSvcOption	[optional] • Array of Service Option.(NULL when not present) See arrSvcOption for more information. – Applicable only for "3GPP"
pArrCallEnd-Reason	[optional] • Array of Call End Reason.(NULL when not present) See arrCallEndReason for more information. – Applicable only for "3GPP"
pArrAlphaID	[optional] • Array of Alpha Identifier(NULL when not present) See arrAlphaID for more information. – Applicable only for "3GPP"
pArrConnect-PartyNum	[optional] • Array of Connected Party Number.(NULL when not present) See arrConnectPartyNum for more information. – Applicable for both "3GPP/3GPP2"
pArrDiagInfo	[optional] • Array of Diagnostic Information.(NULL when not present) See arrDiagInfo for more information. – Applicable only for "3GPP"
pArrCalledParty-Num	[optional] • Array of Called Party Number.(NULL when not present) See arrCalledPartyNum for more information. – Applicable only for "3GPP"
pArrRedirParty-Num	[optional] • Array of Redirecting Party Number.(NULL when not present) See arrRedirPartyNum for more information. – Applicable only for "3GPP"
pArrAlerting-Pattern	[optional] • Array of Alerting Pattern.(NULL when not present) See arrAlertingPattern for more information. – Applicable only for "3GPP"

Note

Optional paramters would be NULL, if not received from the device.

8.1348.2 Field Documentation

8.1348.2.1 arrCallInfo voiceSetAllCallStatusCbkInfo::arrCallInfomation

8.1348.2.2 arrAlertingPattern* voiceSetAllCallStatusCbkInfo::pArrAlertingPattern

8.1348.2.3 arrAlertingType* voiceSetAllCallStatusCbkInfo::pArrAlertingType

8.1348.2.4 arrAlphaID* voiceSetAllCallStatusCbkInfo::pArrAlphaID

8.1348.2.5 arrCalledPartyNum* voiceSetAllCallStatusCbkInfo::pArrCalledPartyNum

8.1348.2.6 arrCallEndReason* voiceSetAllCallStatusCbkInfo::pArrCallEndReason

8.1348.2.7 arrConnectPartyNum* voiceSetAllCallStatusCbkInfo::pArrConnectPartyNum

8.1348.2.8 arrDiagInfo* voiceSetAllCallStatusCbkInfo::pArrDiagInfo

8.1348.2.9 arrRedirPartyNum* voiceSetAllCallStatusCbkInfo::pArrRedirPartyNum

8.1348.2.10 arrRemotePartyName* voiceSetAllCallStatusCbkInfo::pArrRemotePartyName

8.1348.2.11 arrRemotePartyNum* voiceSetAllCallStatusCbkInfo::pArrRemotePartyNum

8.1348.2.12 arrSvcOption* voiceSetAllCallStatusCbkInfo::pArrSvcOption

8.1349 voiceSetCallBarringPwdInfo Struct Reference

Data Fields

- BYTE Reason
- BYTE oldPasswd [4]
- BYTE newPasswd [4]
- BYTE newPasswdAgain [4]

8.1349.1 Detailed Description

This structure contains Voice Set Call Barring Password Request Parameters

Parameters

Reason	• Call Barring Reason • Values: – 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
oldPasswd[PASSWORD_LENGTH]	• Old password. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.
newPasswd[PASSWORD_LENGTH]	• New password. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.
newPasswdAgain[PASSWORD_LENGTH]	• New password Again. – Password consists of 4 ASCII digits. – Range: 0000 to 9999.

8.1349.2 Field Documentation

8.1349.2.1 BYTE voiceSetCallBarringPwdInfo::newPasswd[4]

8.1349.2.2 BYTE voiceSetCallBarringPwdInfo::newPasswdAgain[4]

8.1349.2.3 BYTE voiceSetCallBarringPwdInfo::oldPasswd[4]

8.1349.2.4 BYTE voiceSetCallBarringPwdInfo::Reason

8.1350 voiceSetCallBarringPwdResp Struct Reference

Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1350.1 Detailed Description

This structure contains Voice Set Call Barring Password Response Parameters

Parameters

pFailCause	• Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available
pAlphaIDInfo	• Pointer to structure of alphaIDInfo (optional) – See alphaIDInfo for more information
pCCResType	• Call Control Result Type (optional) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available
pCallID	• Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid.
pCCSUPSType	• Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. – See ccSUPSType for more information

Note

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

8.1350.2 Field Documentation

8.1350.2.1 alphaIDInfo* voiceSetCallBarringPwdResp::pAlphaIDInfo

8.1350.2.2 BYTE* voiceSetCallBarringPwdResp::pCallID

8.1350.2.3 BYTE* voiceSetCallBarringPwdResp::pCCResType

8.1350.2.4 ccSUPSType* voiceSetCallBarringPwdResp::pCCSUPSType

8.1350.2.5 WORD* voiceSetCallBarringPwdResp::pFailCause

8.1351 voiceSetConfigReq Struct Reference

Data Fields

- BYTE * pAutoAnswer

- airTimer * pAirTimerConfig
- roamTimer * pRoamTimerConfig
- BYTE * pTTYMode
- prefVoiceSO * pPrefVoiceSO
- BYTE * pPrefVoiceDomain

8.1351.1 Detailed Description

This structure contains information about the Set Configuration Request Parameters.

Parameters

pAutoAnswer	• Value specified is written to NV_AUTO_ANSWER_I. (optional) • Values: – 0x00 - Disable – 0x01 - Enable
pAirTimerConfig	• Value specified is written to NV_AIR_CNT_I. (optional) • See airTimer for more information
pRoamTimer-Config	• Value specified is written to NV_ROAM_CNT_I. (optional) • See roamTimer for more information
pTTYMode	• Value specified is written to NV_TTY_I. (optional) • Values: – 0x00 - TTY_MODE_FULL - Full – 0x01 - TTY_MODE_VCO - Voice carry over – 0x02 - TTY_MODE_HCO - Hearing carry over – 0x03 - TTY_MODE_OFF - Off
pPrefVoiceSO	• Value specified is written to NV_PREF_VOICE_SO_I. (optional) • See prefVoiceSO for more information
pPrefVoice-Domain	• Preferred Voice-Domain. (optional) • Values: – 0x00 - VOICE_DOMAIN_PREF_CS_ONLY - Circuit-switched (CS) only – 0x01 - VOICE_DOMAIN_PREF_PS_ONLY - Packet-switched (PS) only – 0x02 - VOICE_DOMAIN_PREF_CS_PREF - CS is preferred, PS is secondary – 0x03 - VOICE_DOMAIN_PREF_PS_PREF - PS is preferred, CS is secondary

Note

One of the optional parameters must be present in the request.

8.1351.2 Field Documentation

8.1351.2.1 airTimer* voiceSetConfigReq::pAirTimerConfig

8.1351.2.2 BYTE* voiceSetConfigReq::pAutoAnswer

8.1351.2.3 BYTE* voiceSetConfigReq::pPrefVoiceDomain

8.1351.2.4 prefVoiceSO* voiceSetConfigReq::pPrefVoiceSO

8.1351.2.5 roamTimer* voiceSetConfigReq::pRoamTimerConfig

8.1351.2.6 BYTE* voiceSetConfigReq::pTTYMode

8.1352 voiceSetConfigResp Struct Reference

Data Fields

- BYTE * pAutoAnsStatus
- BYTE * pAirTimerStatus
- BYTE * pRoamTimerStatus
- BYTE * pTTYConfigStatus
- BYTE * pPrefVoiceSOStatus
- BYTE * pVoiceDomainPrefStatus

8.1352.1 Detailed Description

This structure contains information about the Set Configuration Response Parameters.

Parameters

pAutoAnsStatus	• Auto Answer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pAirTimerStatus	• Air Timer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pRoamTimer-Status	• Roam Timer Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.

pTTYConfig-Status	• TTY Config Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pPrefVoiceSO-Status	• Preferred Voice SO Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.
pVoiceDomain-PrefStatus	• Voice-Domain Preference Status. (optional) • Values: – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available.

Note

Parameters which are mentioned as NULL will be ignored.

8.1352.2 Field Documentation

8.1352.2.1 BYTE* voiceSetConfigResp::pAirTimerStatus

8.1352.2.2 BYTE* voiceSetConfigResp::pAutoAnsStatus

8.1352.2.3 BYTE* voiceSetConfigResp::pPrefVoiceSOStatus

8.1352.2.4 BYTE* voiceSetConfigResp::pRoamTimerStatus

8.1352.2.5 BYTE* voiceSetConfigResp::pTTYConfigStatus

8.1352.2.6 BYTE* voiceSetConfigResp::pVoiceDomainPrefStatus

8.1353 voiceSetPrefPrivacy Struct Reference

Data Fields

- BYTE privacyPref

8.1353.1 Detailed Description

This structure contains the preferred voice privacy values.

Parameters

privacyPref	• Voice Privacy Preference – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy

8.1353.2 Field Documentation

8.1353.2.1 BYTE voiceSetPrefPrivacy::privacyPref

8.1354 voiceSetSUPSServiceReq Struct Reference

Data Fields

- BYTE voiceSvc
- BYTE reason
- BYTE * pServiceClass
- BYTE * pCallBarringPasswd
- BYTE * pCallForwardingNumber
- BYTE * pTimerVal
- callFwdTypeAndPlan * pCallFwdTypeAndPlan

8.1354.1 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) – 0x01 - VOICE_SERVICE_ACTIVATE – 0x02 - VOICE_SERVICE_DEACTIVATE – 0x03 - VOICE_SERVICE_REGISTER – 0x04 - VOICE_SERVICE_ERASE
reason	• supplementary service reason values (mandatory) – 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 calls are barred – 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT Outgoing internal calls are barred – 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME Outgoing calls external to home are barred – 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING All incoming calls are barred – 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING Roaming incoming calls are barred – 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
pServiceClass	• Service class is a combination (sum) of information class constants (optional) – 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) – 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) – See callFwdTypeAndPlan for more information

8.1354.2 Field Documentation

8.1354.2.1 BYTE* voiceSetSUPSServiceReq::pCallBarringPasswd

8.1354.2.2 BYTE* voiceSetSUPSServiceReq::pCallForwardingNumber

8.1354.2.3 callFwdTypeAndPlan* voiceSetSUPSServiceReq::pCallFwdTypeAndPlan

8.1354.2.4 BYTE* voiceSetSUPSServiceReq::pServiceClass

8.1354.2.5 BYTE* voiceSetSUPSServiceReq::pTimerVal

8.1354.2.6 BYTE voiceSetSUPSServiceReq::reason

8.1354.2.7 BYTE voiceSetSUPSServiceReq::voiceSvc

8.1355 voiceSetSUPSServiceResp Struct Reference

Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1355.1 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). – 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) – See alphaIDInfo for more information
pCCResultType	• Call control result types (optional, supply NULL if not required) – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - if the device does not provide this information
pCallID	• Unique call identifier for the dialed call (optional, supply NULL if not required) – 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) – See ccSUPSType for more information

8.1355.2 Field Documentation

8.1355.2.1 alphaIDInfo* voiceSetSUPSServiceResp::pAlphaIDInfo

8.1355.2.2 BYTE* voiceSetSUPSServiceResp::pCallID

8.1355.2.3 BYTE* voiceSetSUPSServiceResp::pCCResultType

8.1355.2.4 ccSUPSType* voiceSetSUPSServiceResp::pCCSUPSType

8.1355.2.5 WORD* voiceSetSUPSServiceResp::pFailCause

8.1356 voiceStopContDTMFinfo Struct Reference

Data Fields

- BYTE callID

8.1356.1 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. Stop continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF. • 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. • If the call ID value received is 0, no value has been returned by the device

8.1356.2 Field Documentation

8.1356.2.1 BYTE voiceStopContDTMFinfo::callID

8.1357 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

8.1357.1 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) • 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. • ASCII String, NULL terminated.
pCallFWTimer-Val(optional)	• Call Forwarding No Reply Timer. – 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. – Password consists of 4 ASCII digits. – Range: 0000 to 9999. • 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. • Used to distinguish between the supplementary service data sent to the network and the response received from the network. • 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. • 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

8.1357.2 Field Documentation

8.1357.2.1 alphaIDInfo* voiceSUPSInfo::pAlphaIDInfo

8.1357.2.2 BYTE* voiceSUPSInfo::pCallBarPasswd

8.1357.2.3 getCallFWInfo* voiceSUPSInfo::pCallFwdInfo

8.1357.2.4 BYTE* voiceSUPSInfo::pCallFWNum

8.1357.2.5 BYTE* voiceSUPSInfo::pCallFWTimerVal

8.1357.2.6 BYTE* voiceSUPSInfo::pCallID

8.1357.2.7 CLIPResp* voiceSUPSInfo::pCLIPstatus

8.1357.2.8 CLIRResp* voiceSUPSInfo::pCLIRstatus

8.1357.2.9 CNAPResp* voiceSUPSInfo::pCNAPstatus

8.1357.2.10 COLPResp* voiceSUPSInfo::pCOLPstatus

8.1357.2.11 COLRResp* voiceSUPSInfo::pCOLRstatus

8.1357.2.12 BYTE* voiceSUPSInfo::pDataSrc

8.1357.2.13 WORD* voiceSUPSInfo::pFailCause

8.1357.2.14 newPwdData* voiceSUPSInfo::pNewPwdData

8.1357.2.15 BYTE* voiceSUPSInfo::pReason

8.1357.2.16 BYTE* voiceSUPSInfo::pSvcClass

8.1357.2.17 struct USSInfo* voiceSUPSInfo::pUSSInfo

8.1357.2.18 SUPSInfo voiceSUPSInfo::SUPSInformation

8.1358 voiceSUPSNotification Struct Reference

Data Fields

- BYTE callID
- BYTE notifType
- WORD * pCUGIndex
- ECTNum * pECTNum

8.1358.1 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) – 0x01 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_FORWARDED Originated MO call is being forwarded to another user – 0x02 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_WAITING Originated MO call is waiting at the called user – 0x03 - NOTIFICATION_TYPE_OUTGOING_CUG_CALL Outgoing call is a CUG call – 0x04 - NOTIFICATION_TYPE_OUTGOING_CALLS_BARRED Outgoing calls are barred – 0x05 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_DEFLECTED Outgoing call is deflected – 0x06 - NOTIFICATION_TYPE_INCOMING_CUG_CALL Incoming call is a CUG call – 0x07 - NOTIFICATION_TYPE_INCOMING_CALLS_BARRED Incoming calls are barred – 0x08 - NOTIFICATION_TYPE_INCOMING_FORWARDED_CALL Incoming call received is a forwarded call – 0x09 - NOTIFICATION_TYPE_INCOMING_DEFLECTED_CALL Incoming call is a deflected call – 0x0A - NOTIFICATION_TYPE_INCOMING_CALL_IS_FORWARDED Incoming call is forwarded to another user – 0x0B - NOTIFICATION_TYPE_UNCOND_CALL_FORWARD_ACTIVE Unconditional call forwarding is active – 0x0C - NOTIFICATION_TYPE_COND_CALL_FORWARD_ACTIVE Conditional call forwarding is active – 0x0D - NOTIFICATION_TYPE_CLIR_SUPPRESSION_REJECTED CLIR suppression is rejected – 0x0E - NOTIFICATION_TYPE_CALL_IS_ON_HOLD Call is put on hold at the remote party – 0x0F - NOTIFICATION_TYPE_CALL_IS_RETRIEVED Call is retrieved at the remote party from the hold state – 0x10 - NOTIFICATION_TYPE_CALL_IS_IN_MPTY Call is in a conference – 0x11 - NOTIFICATION_TYPE_INCOMING_CALL_IS_ECT 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) 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) Refer ECTNum for details.

Note

None

8.1358.2 Field Documentation

8.1358.2.1 BYTE voiceSUPSNotification::callID

8.1358.2.2 BYTE voiceSUPSNotification::notifType

8.1358.2.3 WORD* voiceSUPSNotification::pCUGIndex

8.1358.2.4 ECTNum* voiceSUPSNotification::pECTNum

8.1359 wcdmaCellInfo Struct Reference

Data Fields

- WORD psc
- SHORT cpich_rscp
- SHORT cpich_ecno
- SHORT srxlev

8.1359.1 Detailed Description

This structure contains information about the WCDMA Cell.

Parameters

psc	• Primary scrambling code. • Range: 0 to 511.
cpich_rscp	• Absolute power level (in 1/10 dBm) of the common pilot channel as received by the UE. • Range: -120.0 dBm to -25.0 dBm
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. • Range: -50.0 dB to 0.
srxlev	• Cell selection Rx level (Srxlev) value. • Range: -128 to 128. • This field is only valid when ue_in_idle is TRUE.

8.1359.2 Field Documentation

8.1359.2.1 SHORT wcdmaCellInfo::cpich_ecno

8.1359.2.2 SHORT wcdmaCellInfo::cpich_rscp

8.1359.2.3 WORD wcdmaCellInfo::psc

8.1359.2.4 SHORT wcdmaCellInfo::srxlev

8.1360 WCDMAECIOThresh Struct Reference

Data Fields

- BYTE WCDMAECIOThreshListLen
- WORD * pWCDMAECIOThreshList

8.1360.1 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) • Maximum of 32 values • Range for ECIO values: -31.5 to 0 (in dB)

8.1360.2 Field Documentation

8.1360.2.1 WORD* WCDMAECIOThresh::pWCDMAECIOThreshList

8.1360.2.2 BYTE WCDMAECIOThresh::WCDMAECIOThreshListLen

8.1361 WCDMAInfoLTENeighborCell Struct Reference

Data Fields

- ULONG wcdmaRRCState
- BYTE umtsLTENbrCellLen
- umtsLTENbrCell UMTSLTENbrCell [255]

8.1361.1 Detailed Description

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

Parameters

wcdmaRRC-State	• WCDMA RRC states. • Defined in 3GPP TS 25.331 • Values: – 0x00 - NAS_WCDMA_RRC_STATE_DISCONNECTED * WCDMA RRC State is IDLE – 0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH * WCDMA RRC state is CELL_PCH – 0x02 - NAS_WCDMA_RRC_STATE_URA_PCH * WCDMA RRC state is URA_PCH – 0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH * WCDMA RRC state is CELL_FACH – 0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH * WCDMA RRC state is CELL_DCH
umtsLTENbr-CellLen	• Number of sets of UMTS LTE Neighbors.
UMTSLTENbr-Cell	• See umtsLTENbrCell for more information.

8.1361.2 Field Documentation

8.1361.2.1 umtsLTENbrCell WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.1361.2.2 BYTE WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.1361.2.3 ULONG WCDMAInfoLTENeighborCell::wcdmaRRCState

8.1362 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

8.1362.1 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
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 • concatenated SMS.

8.1362.2 Field Documentation

8.1362.2.1 BYTE wcdmaLongMsgDecodingParams::Date[0x09]

8.1362.2.2 BOOL* wcdmaLongMsgDecodingParams::pIsUDHPresent

8.1362.2.3 BYTE* wcdmaLongMsgDecodingParams::pMessage

8.1362.2.4 BYTE* wcdmaLongMsgDecodingParams::pPartNum

8.1362.2.5 BYTE* wcdmaLongMsgDecodingParams::pReferenceNum

8.1362.2.6 CHAR* wcdmaLongMsgDecodingParams::pScAddr

8.1362.2.7 BYTE* wcdmaLongMsgDecodingParams::pScAddrLength

8.1362.2.8 CHAR* wcdmaLongMsgDecodingParams::pSenderAddr

8.1362.2.9 BYTE* wcdmaLongMsgDecodingParams::pSenderAddrLength

8.1362.2.10 CHAR* wcdmaLongMsgDecodingParams::pTextMsg

8.1362.2.11 BYTE* wcdmaLongMsgDecodingParams::pTextMsgLength

8.1362.2.12 BYTE* wcdmaLongMsgDecodingParams::pTotalNum

8.1362.2.13 BYTE wcdmaLongMsgDecodingParams::Time[0x09]

8.1363 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]

8.1363.1 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

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
pDate	• Date fecthed from message

8.1363.2 Field Documentation

8.1363.2.1 BYTE wcdmaMsgDecodingParams::Date[0x09]

8.1363.2.2 BYTE* wcdmaMsgDecodingParams::pMessage

8.1363.2.3 CHAR* wcdmaMsgDecodingParams::pScAddr

8.1363.2.4 BYTE* wcdmaMsgDecodingParams::pScAddrLength

8.1363.2.5 CHAR* wcdmaMsgDecodingParams::pSenderAddr

8.1363.2.6 BYTE* wcdmaMsgDecodingParams::pSenderAddrLength

8.1363.2.7 CHAR* wcdmaMsgDecodingParams::pTextMsg

8.1363.2.8 BYTE* wcdmaMsgDecodingParams::pTextMsgLength

8.1363.2.9 BYTE wcdmaMsgDecodingParams::Time[0x09]

8.1364 wcdmaMsgEncodingParams Struct Reference

Data Fields

- ULONG messageSize
- CHAR * pDestAddr
- CHAR * pTextMsg
- CHAR * pPDUMessage
- BYTE alphabet

8.1364.1 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 – 0 - 7 bit encoding – 4 - 8 bit encoding – 8 - 16 bit UCS2 encoding – others value will be treated as default 7 bit encoding

8.1364.2 Field Documentation

8.1364.2.1 BYTE wcdmaMsgEncodingParams::alphabet

8.1364.2.2 ULONG wcdmaMsgEncodingParams::messageSize

8.1364.2.3 CHAR∗ wcdmaMsgEncodingParams::pDestAddr

8.1364.2.4 CHAR∗ wcdmaMsgEncodingParams::pPDUMessage

8.1364.2.5 CHAR∗ wcdmaMsgEncodingParams::pTextMsg

8.1365 WCDMARSSIThresh Struct Reference

Data Fields

- BYTE WCDMARSSIThreshListLen
- WORD ∗ pWCDMARSSIThreshList

8.1365.1 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) • Maximum of 32 values. • Range for RSSI values: -121 to 0 (in dBm)

8.1365.2 Field Documentation

8.1365.2.1 WORD* WCDMARSSIThresh::pWCDMARSSIThreshList

8.1365.2.2 BYTE WCDMARSSIThresh::WCDMARSSIThreshListLen

8.1366 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

8.1366.1 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.. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

lac	• Location area code. • Only applies to 3GPP. – 0xFFFF - Not Available
cellIdValid	• Indicates whether the cell ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
cellId	• Cell ID. – 0xFFFFFFFF - Not Available
regRejectInfo-Valid	• Indicates whether the registration reject information is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
rejectSrvDomain	• Type of service domain in which the registration is rejected. – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
rejCause	• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. – 0xFF - Not Available
networkIdValid	• Indicates whether the network ID is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
MCC[PLMN_LE-NGTH]	• Mobile Country Code. • MCC digits in ASCII characters
MNC[PLMN_LE-NGTH]	• Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.

hsCallStatus-Valid	• Indicates whether the high-speed call status is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
hsCallStatus	• Call status on high speed. • Only applicable for WCDMA. – 0x00 - HSDPA and HSUPA are unsupported – 0x01 - HSDPA is supported – 0x02 - HSUPA is supported – 0x03 - HSDPA and HSUPA are supported – 0x04 - HSDPA+ is supported – 0x05 - HSDPA+ and HSUPA are supported – 0x06 - Dual-cell HSDPA+ is supported – 0x07 - Dual-cell HSDPA+ and HSUPA are supported – 0xFF - Not Available
hsIndValid	• Indicates whether high-speed service indication is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
hsInd	• High-speed service indication • Only applicable for WCDMA. – 0x00 - HSDPA and HSUPA are unsupported – 0x01 - HSDPA is supported – 0x02 - HSUPA is supported – 0x03 - HSDPA and HSUPA are supported – 0x04 - HSDPA+ is supported – 0x05 - HSDPA+ and HSUPA are supported – 0x06 - Dual-cell HSDPA+ is supported – 0x07 - Dual-cell HSDPA+ and HSUPA are supported – 0xFF - Not Available
pscValid	• Indicates whether primary scrambling code is valid. – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
psc	• Primary scrambling code. – 0xFFFF - Not Available

8.1366.2 Field Documentation

8.1366.2.1 ULONG WCDMASysInfo::cellId

8.1366.2.2 BYTE WCDMASysInfo::cellIdValid

8.1366.2.3 BYTE WCDMASysInfo::hsCallStatus

8.1366.2.4 BYTE WCDMASysInfo::hsCallStatusValid

8.1366.2.5 BYTE WCDMASysInfo::hsInd

8.1366.2.6 BYTE WCDMASysInfo::hsIndValid

8.1366.2.7 WORD WCDMASysInfo::lac

8.1366.2.8 BYTE WCDMASysInfo::lacValid

8.1366.2.9 BYTE WCDMASysInfo::MCC[3]

8.1366.2.10 BYTE WCDMASysInfo::MNC[3]

8.1366.2.11 BYTE WCDMASysInfo::networkIdValid

8.1366.2.12 WORD WCDMASysInfo::psc

8.1366.2.13 BYTE WCDMASysInfo::pscValid

8.1366.2.14 BYTE WCDMASysInfo::regRejectInfoValid

8.1366.2.15 BYTE WCDMASysInfo::rejCause

8.1366.2.16 BYTE WCDMASysInfo::rejectSrvDomain

8.1366.2.17 sysInfoCommon WCDMASysInfo::sysInfoWCDMA

8.1367 wcdmaUARFCN Struct Reference

Data Fields

- BYTE status
- ULONG uarfcn

8.1367.1 Detailed Description

This structure contains the parameters for WCDMA UARFCN.

Parameters

status	• 0 - Disable • 1 - Enable
uarfcn	• UARFCN to which UE is locked

8.1367.2 Field Documentation

8.1367.2.1 BYTE wcdmaUARFCN::status

8.1367.2.2 ULONG wcdmaUARFCN::uarfcn

8.1368 wds_currNetworkInfo Struct Reference

Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

8.1368.1 Detailed Description

Network information structure

Parameters

NetworkType	• Values: – 0 - 3GPP – 1 - 3GPP2
RATMask	• Radio Access Technology (RAT) mask to indicate the type of technology. • Values: – 0 - Don't Care – 0x8000 - NULL Bearer • CDMA RAT mask values: – 0x01 - CDMA_1x – 0x02 - EVDO_REV0 – 0x05 - HRPD – 0x0B - EHRPD • UMTS RAT mask values: – 0x03 - GPRS – 0x04 - WCDMA – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - HSUPA and WCDMA – 0x08 - HSDPA and HSUPA – 0x0A - LTE – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA

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

8.1368.2 Field Documentation

8.1368.2.1 uint8_t wds_currNetworkInfo::NetworkType

8.1368.2.2 uint32_t wds_currNetworkInfo::RATMask

8.1368.2.3 uint32_t wds_currNetworkInfo::SOMask

8.1369 wds_DataULongLongTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t ullData

8.1369.1 Field Documentation

8.1369.1.1 uint8_t wds_DataULongLongTlv::TlvPresent

8.1369.1.2 uint64_t wds_DataULongLongTlv::ullData

8.1370 wds_DataULongTlv Struct Reference

Data Fields

- uint8_t TlvPresent

- uint32_t ulData

8.1370.1 Field Documentation

8.1370.1.1 uint8_t wds_DataULongTlv::TlvPresent

8.1370.1.2 uint32_t wds_DataULongTlv::ulData

8.1371 wds_DHCPLeaseOptTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t numOpt
- wds_DHCPOpt optList [30]
- uint8_t optListData [2048]

8.1371.1 Field Documentation

8.1371.1.1 uint8_t wds_DHCPLeaseOptTlv::numOpt

8.1371.1.2 wds_DHCPOpt wds_DHCPLeaseOptTlv::optList[30]

8.1371.1.3 uint8_t wds_DHCPLeaseOptTlv::optListData[2048]

8.1371.1.4 uint8_t wds_DHCPLeaseOptTlv::TlvPresent

8.1372 wds_DHCPLeaseStateTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t leaseState

8.1372.1 Field Documentation

8.1372.1.1 uint8_t wds_DHCPLeaseStateTlv::leaseState

8.1372.1.2 uint8_t wds_DHCPLeaseStateTlv::TlvPresent

8.1373 wds_DHCPOpt Struct Reference

Data Fields

- uint8_t optCode
- uint8_t optValLen
- uint8_t * pOptVal

8.1373.1 Field Documentation

8.1373.1.1 uint8_t wds_DHCPOpt::optCode

8.1373.1.2 uint8_t wds_DHCPOpt::optValLen

8.1373.1.3 uint8_t* wds_DHCPOpt::pOptVal

8.1374 wds_DHCPProfileIdTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t profileType
- uint8_t profileId

8.1374.1 Field Documentation

8.1374.1.1 uint8_t wds_DHCPProfileIdTlv::profileId

8.1374.1.2 uint8_t wds_DHCPProfileIdTlv::profileType

8.1374.1.3 uint8_t wds_DHCPProfileIdTlv::TlvPresent

8.1375 wds_DHCPv4HWConfig Struct Reference

Data Fields

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

8.1375.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

Parameters

hwType	• HW Type 1 - Ethernet 20 - Serial
chaddrlen	• chaddrlen
chaddr	• chaddr. Max size 16 bytes

8.1375.2 Field Documentation

8.1375.2.1 uint8_t wds_DHCPv4HWConfig::chaddr[16]

8.1375.2.2 uint8_t wds_DHCPv4HWConfig::chaddrLen

8.1375.2.3 uint8_t wds_DHCPv4HWConfig::hwType

8.1376 wds_DHCPv4Option Struct Reference

Data Fields

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

8.1376.1 Detailed Description

WDS SWI DHCPv4 Option Structure

Parameters

optCode	• Option code – 0 - 255
optValLen	• Option value length – 0 - 255
optVal	• Option value

8.1376.2 Field Documentation

8.1376.2.1 uint8_t wds_DHCPv4Option::optCode

8.1376.2.2 uint8_t wds_DHCPv4Option::optVal[255]

8.1376.2.3 uint8_t wds_DHCPv4Option::optValLen

8.1377 wds_DHCPv4OptionList Struct Reference

Data Fields

- uint8_t numOpt
- wds_DHCPv4Option * pOptList

8.1377.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

Parameters

numOpt	• number of options – 0 - 255
pOptList	• pointer to list of DHCP Options

8.1377.2 Field Documentation

8.1377.2.1 uint8_t wds_DHCPv4OptionList::numOpt

8.1377.2.2 wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList

8.1378 wds_DHCPv4ProfileId Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileId

8.1378.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

Parameters

profileType	• 0 for 3GPP
profileId	• 1 to 24 for 3GPP profile

8.1378.2 Field Documentation

8.1378.2.1 uint8_t wds_DHCPv4ProfileId::profileId

8.1378.2.2 uint8_t wds_DHCPv4ProfileId::profileType

8.1379 wds_Domain Struct Reference

Data Fields

- uint16_t domainLen
- uint8_t domainName [256]

8.1379.1 Detailed Description

This structure contains the DomainName Information

Parameters

domainLen	• length of the recieved Domain name
domainName	• Domain name(Max 256 characters)

8.1379.2 Field Documentation

8.1379.2.1 uint16_t wds_Domain::domainLen

8.1379.2.2 uint8_t wds_Domain::domainName[256]

8.1380 wds_DomainNameList Struct Reference

Data Fields

- uint8_t numInstances
- struct wds_Domain domain [10]

8.1380.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

numInstances	• Number of Domain name receieved
domain	• Domain name information(Max 10 Domain names)

8.1380.2 Field Documentation

8.1380.2.1 struct wds_Domain wds_DomainNameList::domain[10]

8.1380.2.2 uint8_t wds_DomainNameList::numInstances

8.1381 wds_GPRSQoS Struct Reference

Data Fields

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

8.1381.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

Parameters

precedence-Class	• Precedence class
delayClass	• Delay class
reliabilityClass	• Reliability class
peak-Throughput-Class	• Peak throughput class
mean-Throughput-Class	• Mean throughput class

8.1381.2 Field Documentation

8.1381.2.1 uint32_t wds_GPRSQoS::delayClass

8.1381.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

8.1381.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

8.1381.2.4 uint32_t wds_GPRSQoS::precedenceClass

8.1381.2.5 uint32_t wds_GPRSQoS::reliabilityClass

8.1382 wds_IPv4AdTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t IPv4Addr

8.1382.1 Field Documentation

8.1382.1.1 uint32_t wds_IPv4AdTlv::IPv4Addr

8.1382.1.2 uint8_t wds_IPv4AdTlv::TlvPresent

8.1383 wds_IPV6AddressInfo Struct Reference

Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

8.1383.1 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 – 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.

8.1383.2 Field Documentation

8.1383.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

8.1383.2.2 uint8_t wds_IPV6AddressInfo::IPV6PrefixLen

8.1384 wds_IPV6GWAddressInfo Struct Reference

Data Fields

- uint8_t gwV6PrefixLen
- uint16_t gwAddressV6 [8]

8.1384.1 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.

8.1384.2 Field Documentation

8.1384.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

8.1384.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

8.1385 wds_PCSCFFQDNAddress Struct Reference

Data Fields

- uint16_t fqdnLen

- uint8_t fqdnAddr [256]

8.1385.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

Parameters

fqdnLen	• length of the received FQDN address
fqdnAddr	• FQDN address(Max 256 characters)

8.1385.2 Field Documentation

8.1385.2.1 uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]

8.1385.2.2 uint16_t wds_PCSCFFQDNAddress::fqdnLen

8.1386 wds_PCSCFFQDNAddressList Struct Reference

Data Fields

- uint8_t numInstances
- struct wds_PCSCFFQDNAddress pcsfFQDNAddress [10]

8.1386.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

Parameters

numInstances	• Number of FQDN addresses received
pcsfFQDN-Address	• FQDN address information(Max 10 addresses)

8.1386.2 Field Documentation

8.1386.2.1 uint8_t wds_PCSCFFQDNAddressList::numInstances

8.1386.2.2 struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]

8.1387 wds_PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- uint8_t numInstances
- uint32_t pscsfIPv4Addr [64]

8.1387.1 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)

8.1387.2 Field Documentation

8.1387.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

8.1387.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

8.1388 wds_ProfileIdentifier Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileIndex

8.1388.1 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

8.1388.2 Field Documentation

8.1388.2.1 uint8_t wds_ProfileIdentifier::profileIndex

8.1388.2.2 uint8_t wds_ProfileIdentifier::profileType

8.1389 wds_profileInfo Union Reference

Data Fields

- LibPackprofile_3GPP SlqsProfile3GPP
- LibPackprofile_3GPP2 SlqsProfile3GPP2

8.1389.1 Detailed Description

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

8.1389.2 Field Documentation

8.1389.2.1 LibPackprofile_3GPP wds_profileInfo::SlqsProfile3GPP

8.1389.2.2 LibPackprofile_3GPP2 wds_profileInfo::SlqsProfile3GPP2

8.1390 wds_TrStatInd Struct Reference

Data Fields

- uint8_t statsPeriod
- uint32_t statsMask

8.1390.1 Detailed Description

This structure contains the information about the Transfer Statistics Indicator parameters.

Parameters

statsPeriod	• Period between transfer statistics reports. – 0 - Do not report – Other - Period between reports (seconds)
statsMask	• Requested statistic bit 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 • Each bit set causes the corresponding optional information to be sent in SLQSWdsEventReport-CallBack. • All unlisted bits are reserved for future use and must be set to zero.

8.1390.2 Field Documentation

8.1390.2.1 uint32_t wds_TrStatInd::statsMask

8.1390.2.2 uint8_t wds_TrStatInd::statsPeriod

8.1391 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

8.1391.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

Parameters

trafficClass	• 0x00 - Subscribed • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 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 • 0x00 - Subscribe • 0x01 - Delivery order on • 0x02 - Delivery order off
maxSDUSize	• Maximum SDU size

sduErrorRatio	- SDU error ratio • Target value for fraction of SDUs lost or detected as erroneous. • 0x00 - Subscribe • 0x01 - 1∗10^(-2) • 0x02 - 7∗10^(-3) • 0x03 - 1∗10^(-3) • 0x04 - 1∗10^(-4) • 0x05 - 1∗10^(-5) • 0x06 - 1∗10^(-6) • 0x07 - 1∗10^(-1)
resBerRatio	- Residual bit error ratio • Target value for undetected bit error ratio in in the delivered SDUs. • 0x00 - Subscribe • 0x01 - 5∗10^(-2) • 0x02 - 1∗10^(-2) • 0x03 - 5∗10^(-3) • 0x04 - 4∗10^(-3) • 0x05 - 1∗10^(-3) • 0x06 - 1∗10^(-4) • 0x07 - 1∗10^(-5) • 0x08 - 1∗10^(-6) • 0x09 - 1∗10^(-8)
deliveryErrSDU	- delivery of erroneous SDUs • Indicates whether SDUs detected as erroneous shall be delivered or not. • 0x00 - Subscribe • 0x01 - 5∗10^(-2) • 0x02 - 1∗10^(-2) • 0x03 - 5∗10^(-3) • 0x04 - 4∗10^(-3) • 0x05 - 1∗10^(-3) • 0x06 - 1∗10^(-4) • 0x07 - 1∗10^(-5) • 0x08 - 1∗10^(-6) • 0x09 - 1∗10^(-8)
transferDelay	- Transfer delay (ms) • Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds.
trafficPriority	- Transfer handling priority • Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.

8.1391.2 Field Documentation

8.1391.2.1 uint8_t wds_UMTSMinQoS::deliveryErrSDU

8.1391.2.2 uint32_t wds_UMTSMinQoS::grntDownlinkBitrate

8.1391.2.3 uint32_t wds_UMTSMinQoS::grntUplinkBitrate

8.1391.2.4 uint32_t wds_UMTSMinQoS::maxDownlinkBitrate

8.1391.2.5 uint32_t wds_UMTSMinQoS::maxSDUSize

8.1391.2.6 uint32_t wds_UMTSMinQoS::maxUplinkBitrate

8.1391.2.7 uint8_t wds_UMTSMinQoS::qosDeliveryOrder

8.1391.2.8 uint8_t wds_UMTSMinQoS::resBerRatio

8.1391.2.9 uint8_t wds_UMTSMinQoS::sduErrorRatio

8.1391.2.10 uint8_t wds_UMTSMinQoS::trafficClass

8.1391.2.11 uint32_t wds_UMTSMinQoS::trafficPriority

8.1391.2.12 uint32_t wds_UMTSMinQoS::transferDelay

8.1392 WdsByteTotals Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsByteTotalsElmnts ByteTotalsElmntsV4
- struct WdsByteTotalsElmnts ByteTotalsElmntsV6

8.1392.1 Detailed Description

WDS ByteTotals request data structure

Parameters

pV4sessionId	• The v4 session ID for which the byte totals are to be retrieved • provide a NULL pointer if not applicable
pV6sessionId	• The v6 session ID for which the byte totals are to be retrieved • 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.

8.1392.2 Field Documentation

8.1392.2.1 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV4

8.1392.2.2 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV6

8.1392.2.3 ULONG* WdsByteTotals::pV4sessionId

8.1392.2.4 ULONG* WdsByteTotals::pV6sessionId

8.1393 WdsByteTotalsElmnts Struct Reference

Data Fields

- ULONGLONG * pTXTotalBytes
- ULONGLONG * pRXTotalBytes

8.1393.1 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.

8.1393.2 Field Documentation

8.1393.2.1 ULONGLONG* WdsByteTotalsElmnts::pRXTotalBytes

8.1393.2.2 ULONGLONG* WdsByteTotalsElmnts::pTXTotalBytes

8.1394 WdsClientLeaseChange Struct Reference

Data Fields

- BYTE * pEnableNotification

8.1394.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

Parameters

pEnable-Notification	[IN] • Enable Notification or not

8.1394.2 Field Documentation

8.1394.2.1 BYTE* WdsClientLeaseChange::pEnableNotification

8.1395 WdsConnectionRate Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsConnectionRateElmnts ConnRateElmntsV4
- struct WdsConnectionRateElmnts ConnRateElmntsV6

8.1395.1 Detailed Description

WDS ConnectionRate request data structure

Parameters

pV4sessionId	• The v4 session ID for which the connection rate are to be retrieved • provide a NULL pointer if not applicable
pV6sessionId	• The v6 session ID for which the connection rate are to be retrieved • 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.

8.1395.2 Field Documentation

8.1395.2.1 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV4

8.1395.2.2 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV6

8.1395.2.3 ULONG* WdsConnectionRate::pV4sessionId

8.1395.2.4 ULONG* WdsConnectionRate::pV6sessionId

8.1396 WdsConnectionRateElmnts Struct Reference

Data Fields

- ULONG * pCurrentChannelTXRate
- ULONG * pCurrentChannelRXRate
- ULONG * pMaxChannelTXRate
- ULONG * pMaxChannelRXRate

8.1396.1 Detailed Description

WDS Connection rates request data structure for individual session

Parameters

pCurrentChannelTXRate[OUT]	• Instantaneous channel Tx rate in bits per second.
pCurrentChannelRXRate[OUT]	• Instantaneous channel Rx rate in bits per second.
pMaxChannelTXRate[OUT]	• Maximum Tx rate that can be assigned to the device by the • serving system in bits per second
pMaxChannelRXRate[OUT]	• Maximum Rx rate that can be assigned to the device by the • serving system in bits per second

8.1396.2 Field Documentation

8.1396.2.1 ULONG* WdsConnectionRateElmnts::pCurrentChannelRXRate

8.1396.2.2 ULONG* WdsConnectionRateElmnts::pCurrentChannelTXRate

8.1396.2.3 ULONG* WdsConnectionRateElmnts::pMaxChannelRXRate

8.1396.2.4 ULONG* WdsConnectionRateElmnts::pMaxChannelTXRate

8.1397 WdsDHCPv4ClientLeaseInd Struct Reference

Data Fields

- WdsDHCPv4ProfileId * pProfileId
- BYTE * pLeaseState
- ULONG * pIPv4Addr
- DHCPOptionList * pOptList

8.1397.1 Detailed Description

This structure contains DHCPv4 client lease status

Parameters

pProfileId	• Profile Type and Id
pLeaseState	• Values – 0 - active, newly acquired – 1 - active, renewed – 2 - active, renewing – 3 - active, rebinding – 4 - inactive, expired – 5 - inactive, renew refused – 6 - inactive, rebind refused – 7 - inactive, other
pIPv4Addr	• Values – IPv4 Address
pOptList	• Option list

8.1397.2 Field Documentation

8.1397.2.1 ULONG* WdsDHCPv4ClientLeaseInd::pIPv4Addr

8.1397.2.2 BYTE* WdsDHCPv4ClientLeaseInd::pLeaseState

8.1397.2.3 DHCPOptionList* WdsDHCPv4ClientLeaseInd::pOptList

8.1397.2.4 WdsDHCPv4ProfileId* WdsDHCPv4ClientLeaseInd::pProfileId

8.1398 WdsDHCPv4Config Struct Reference

Data Fields

- WdsDHCPv4ProfileId * pProfileId
- WdsDHCPv4HWConfig * pHwConfig
- WdsDHCPv4OptionList * pRequestOptionList

8.1398.1 Detailed Description

WDS SWI DHCPv4 Config Structure

Parameters

pProfileId	[IN] • pointer to Profile Id structure
pHWConfig	[IN/OUT] • pointer to HW Config structure

pRequestOption-List	[IN/OUT] • pointer to Option List structure to be sent in DHCP request

8.1398.2 Field Documentation

8.1398.2.1 WdsDHCPv4HWConfig* WdsDHCPv4Config::pHwConfig

8.1398.2.2 WdsDHCPv4ProfileId* WdsDHCPv4Config::pProfileId

8.1398.2.3 WdsDHCPv4OptionList* WdsDHCPv4Config::pRequestOptionList

8.1399 wdsDhcpv4HwConfig Struct Reference

Data Fields

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

8.1399.1 Detailed Description

Parameters

hwType	DHCP HW Type, examples: • 0 - Ethernet • 20 - Serial
chaddrLen	Length of chaddr field, examples: • 6 for Ethernet MAC address
chaddr	Client hardware address

8.1399.2 Field Documentation

8.1399.2.1 uint8_t wdsDhcpv4HwConfig::chaddr[16]

8.1399.2.2 uint8_t wdsDhcpv4HwConfig::chaddrLen

8.1399.2.3 uint8_t wdsDhcpv4HwConfig::hwType

8.1400 WdsDHCPv4HWConfig Struct Reference

Data Fields

- BYTE hwType
- BYTE chaddrLen
- BYTE chaddr [16]

8.1400.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

Parameters

hwType	• HW Type 1 - Ethernet 20 - Serial
chaddrlen	• chaddrlen
chaddr	• chaddr. Max size 16 bytes

8.1400.2 Field Documentation

8.1400.2.1 BYTE WdsDHCPv4HWConfig::chaddr[16]

8.1400.2.2 BYTE WdsDHCPv4HWConfig::chaddrLen

8.1400.2.3 BYTE WdsDHCPv4HWConfig::hwType

8.1401 wdsDhcpv4Option Struct Reference

Data Fields

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

8.1401.1 Detailed Description

Parameters

optCode	Option code • 0 - 255
optValLen	Option value length • 0 - 255
optVal	Option Value

8.1401.2 Field Documentation

8.1401.2.1 uint8_t wdsDhcpv4Option::optCode

8.1401.2.2 uint8_t wdsDhcpv4Option::optVal[255]

8.1401.2.3 uint8_t wdsDhcpv4Option::optValLen

8.1402 WdsDHCPv4Option Struct Reference

Data Fields

- BYTE optCode
- BYTE optValLen
- BYTE optVal [255]

8.1402.1 Detailed Description

WDS SWI DHCPv4 Option Structure

Parameters

optCode	• Option code – 0 - 255
optValLen	• Option value length – 0 - 255
optVal	• Option value

8.1402.2 Field Documentation

8.1402.2.1 BYTE WdsDHCPv4Option::optCode

8.1402.2.2 BYTE WdsDHCPv4Option::optVal[255]

8.1402.2.3 BYTE WdsDHCPv4Option::optValLen

8.1403 WdsDHCPv4OptionList Struct Reference

Data Fields

- BYTE numOpt
- WdsDHCPv4Option * pOptList

8.1403.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

Parameters

numOpt	• number of options – 0 - 255
pOptList	• pointer to list of DHCP Options

8.1403.2 Field Documentation

8.1403.2.1 BYTE WdsDHCPv4OptionList::numOpt

8.1403.2.2 WdsDHCPv4Option* WdsDHCPv4OptionList::pOptList

8.1404 wdsDhcpv4OptionList Struct Reference

Data Fields

- uint8_t numOpt
- wdsDhcpv4Option * pOptList

8.1404.1 Detailed Description

Parameters

numOpt	number of options • 0 - 255
pOptList	pointer to list of DHCP Options

8.1404.2 Field Documentation

8.1404.2.1 uint8_t wdsDhcpv4OptionList::numOpt

8.1404.2.2 wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList

8.1405 WdsDHCPv4ProfileId Struct Reference

Data Fields

- BYTE profileType
- BYTE profileId

8.1405.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

Parameters

profileType	• 0 for 3GPP
profileId	• 1 to 24 for 3GPP profile

8.1405.2 Field Documentation

8.1405.2.1 BYTE WdsDHCPv4ProfileId::profileId

8.1405.2.2 BYTE WdsDHCPv4ProfileId::profileType

8.1406 wdsDhcpv4ProfileId Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileId

8.1406.1 Detailed Description

Parameters

profileType	profile type • 0 - 3GPP
profileId	profile index • index identifiying the profile 1-24 valid for 3GPP profile type (EM74xx and onwards)

8.1406.2 Field Documentation

8.1406.2.1 uint8_t wdsDhcpv4ProfileId::profileId

8.1406.2.2 uint8_t wdsDhcpv4ProfileId::profileType

8.1407 WDSGetLoopbackData Struct Reference

Data Fields

- BYTE ByteLoopbackMode
- BYTE ByteLoopbackMultiplier

8.1407.1 Detailed Description

This API to Queries Enable/disable Data Loopback Mode and set the value of loopback multiplier.

Parameters

pReq	[IN] • 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. – 0 - Disable – 1 - Enable
ByteLoopback-Multiplier	• Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.1407.2 Field Documentation

8.1407.2.1 BYTE WDSGetLoopbackData::ByteLoopbackMode

8.1407.2.2 BYTE WDSGetLoopbackData::ByteLoopbackMultiplier

8.1408 WdsIpAddressInfoReq Struct Reference

Data Fields

- ULONG * pv4sessionId
- ULONG * pv6sessionId
- QmiWdsIpAddressInfo ip

8.1408.1 Field Documentation

8.1408.1.1 QmiWdsIpAddressInfo WdsIpAddressInfoReq::ip

8.1408.1.2 ULONG* WdsIpAddressInfoReq::pv4sessionId

8.1408.1.3 ULONG* WdsIpAddressInfoReq::pv6sessionId

8.1409 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

8.1409.1 Detailed Description

WDS Pkt Statistics request data structure for individual session

Parameters

pTXPacket-Successes	• No of transmitted Packets without error.
pRXPacket-Successes	• No 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	• No of bytes transmitted without error.
pRXOkBytes-Count	• No of bytes received without error.
pTXOKBytes-LastCall	• 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
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.

8.1409.2 Field Documentation

8.1409.2.1 ULONG* WdsPktStatisticsElmnts::pRXDroppedCount

8.1409.2.2 ULONGLONG* WdsPktStatisticsElmnts::pRXOkBytesCount

8.1409.2.3 ULONGLONG* WdsPktStatisticsElmnts::pRXOKBytesLastCall

8.1409.2.4 ULONG* WdsPktStatisticsElmnts::pRXPacketErrors

8.1409.2.5 ULONG* WdsPktStatisticsElmnts::pRXPacketOverflows

8.1409.2.6 ULONG* WdsPktStatisticsElmnts::pRXPacketSuccesses

8.1409.2.7 ULONG* WdsPktStatisticsElmnts::pTXDroppedCount

8.1409.2.8 ULONGLONG* WdsPktStatisticsElmnts::pTXOkBytesCount

8.1409.2.9 ULONGLONG* WdsPktStatisticsElmnts::pTXOKBytesLastCall

8.1409.2.10 ULONG* WdsPktStatisticsElmnts::pTXPacketErrors

8.1409.2.11 ULONG* WdsPktStatisticsElmnts::pTXPacketOverflows

8.1409.2.12 ULONG* WdsPktStatisticsElmnts::pTXPacketSuccesses

8.1410 WdsPktStatisticsReq Struct Reference

Data Fields

- ULONG * pStatMask

8.1410.1 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

8.1410.2 Field Documentation

8.1410.2.1 ULONG* WdsPktStatisticsReq::pStatMask

8.1411 WdsPktStatisticsResp Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsPktStatisticsElmnts PktStatElmntsV4
- struct WdsPktStatisticsElmnts PktStatElmntsV6

8.1411.1 Detailed Description

WDS PktStatistics response data structure

Parameters

pV4sessionId	• The v4 session ID for which the byte totals are to be retrieved • provide a NULL pointer if not applicable
pV6sessionId	• The v6 session ID for which the byte totals are to be retrieved • 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.

8.1411.2 Field Documentation

8.1411.2.1 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV4

8.1411.2.2 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV6

8.1411.2.3 ULONG* WdsPktStatisticsResp::pV4sessionId

8.1411.2.4 ULONG* WdsPktStatisticsResp::pV6sessionId

8.1412 WdsProfileParam Union Reference

Data Fields

- struct Profile3GPP SlqsProfile3GPP
- struct Profile3GPP2 SlqsProfile3GPP2

8.1412.1 Detailed Description

This union WdsProfileParam consist of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

8.1412.2 Field Documentation

8.1412.2.1 struct Profile3GPP WdsProfileParam::SlqsProfile3GPP

8.1412.2.2 struct Profile3GPP2 WdsProfileParam::SlqsProfile3GPP2

8.1413 WdsRunTimeSettings Struct Reference

Data Fields

- ULONG * v4sessionId

- ULONG * v6sessionId
- struct qmiWdsRunTimeSettings rts

8.1413.1 Detailed Description

WDS runtime settings request data structure

Parameters

v4sessionId	• The v4 session ID for which the runtime settings are to be retrieved • provide a NULL pointer if not applicable
v6sessionId	• The v6 session ID for which the runtime settings are to be retrieved • 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.

8.1413.2 Field Documentation

8.1413.2.1 struct qmiWdsRunTimeSettings WdsRunTimeSettings::rts

8.1413.2.2 ULONG* WdsRunTimeSettings::v4sessionId

8.1413.2.3 ULONG* WdsRunTimeSettings::v6sessionId

8.1414 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

8.1414.1 Detailed Description

This structure contains the information about the Set Event Report Request parameters.

Parameters

pCurrChannelRateInd	(optional) • Current Channel Rate Indicator. – 0 - Do not report – 1 - Report channel rate when it changes
pTransferStatInd	(optional) • See TrStatInd for more information.
pDataBearerTechInd	(optional) • Data Bearer Technology Indicator. – 0 - Do not report – 1 - Report radio interface used for data transfer when it changes
pDormancyStatusInd	(optional) • Dormancy Status indicator. – 0 - Do not report – 1 - Report traffic channel state of interface used for data connection
pMIPStatusInd	(optional) • MIP Status Indicator. – 0 - Do not report – 1 - Report MIP status
pCurrDataBearerTechInd	(optional) • Current Data Bearer Technology Indicator. – 0 - Do not report – 1 - Report current data bearer technology when it changes
pDataCallStatusChangeInd	(optional) • Data Call Status Change Indicator. – 0 - Do not report – 1 - Report data call status change when it changes
pCurrPrefDataSysInd	(optional) • Current Preferred Data System Indicator. – 0 - Do not report – 1 - Report preferred data system when it changes
pEVDOPageMonPerChangeInd	(optional) • EV-DO Page Monitor Period Change Indicator. – 0 - Do not report – 1 - Report EV-DO page monitor period change event
pDataSystemStatusChangeInd	(optional) • Data System Status Change Indicator. – 0 - Do not report – 1 - Report data system status change event

Note

At least one parameter should be present.

8.1414.2 Field Documentation

8.1414.2.1 BYTE* wdsSetEventReportReq::pCurrChannelRateInd

8.1414.2.2 BYTE* wdsSetEventReportReq::pCurrDataBearerTechInd

8.1414.2.3 BYTE* wdsSetEventReportReq::pCurrPrefDataSysInd

8.1414.2.4 BYTE* wdsSetEventReportReq::pDataBearerTechInd

8.1414.2.5 BYTE* wdsSetEventReportReq::pDataCallStatusChangeInd

8.1414.2.6 BYTE* wdsSetEventReportReq::pDataSystemStatusChangeInd

8.1414.2.7 BYTE* wdsSetEventReportReq::pDormancyStatusInd

8.1414.2.8 BYTE* wdsSetEventReportReq::pEVDOPageMonPerChangeInd

8.1414.2.9 BYTE* wdsSetEventReportReq::pMIPStatusInd

8.1414.2.10 TrStatInd* wdsSetEventReportReq::pTransferStatInd

8.1415 WDSSetLoopbackData Struct Reference

Data Fields

- BYTE * pLoopbackMode
- BYTE * pLoopbackMultiplier

8.1415.1 Detailed Description

WDS SWI Set Loopback Structure of Set Loopback Information.

Parameters

pLoopbackMode	• Loopback Mode. – 0 - Disable – 1 - Enable
pLoopback-Multiplier	• Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.1415.2 Field Documentation

8.1415.2.1 BYTE* WDSSetLoopbackData::pLoopbackMode

8.1415.2.2 BYTE* WDSSetLoopbackData::pLoopbackMultiplier

8.1416 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

8.1416.1 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.

8.1416.2 Field Documentation

8.1416.2.1 unsigned long WDSSWICurrentChannelRates::current_channel_rx_rate

8.1416.2.2 unsigned long WDSSWICurrentChannelRates::current_channel_tx_rate

8.1416.2.3 unsigned long WDSSWICurrentChannelRates::max_channel_rx_rate

8.1416.2.4 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)

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,
  eTMD =0x18,
  eRMS =225,
  eSWIOMA =240,
  eSWIM2MCMD =243,
  eSWIOMAEXT =244,
  eSWILOC =246 }
- enum msgtype {
  eREQ =0,
  eRSP =2,
  eIND =4 }

Functions

- uint16_t helper_get_xid (uint8_t *qmi_resp)
- const char * helper_get_resp_ctx (uint8_t svc, uint8_t *pbuf, uint32_t len, unpack_qmi_t *pCtx)
- unsigned unpack_result_code_only (uint8_t *pMdmResp)
- int helper_set_log_func (logger func)
- void 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 ()

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 SDU_HDR_LEN (3)

9.2.1.8 #define UNUSEDPARAM(*x*) (void)x

9.2.2 Typedef Documentation

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

9.2.3 Enumeration Type Documentation

9.2.3.1 enum eLOG_LEVEL

log levels

Enumerator

eLOG_INFO

eLOG_DEBUG

eLOG_WARN

eLOG_FATAL

9.2.3.2 enum eQMI_SVC

qmi service

Enumerator

eCTL

eWDS

eDMS

eNAS

eQOS

eSMS

ePDS

eVOICE

eCAT

eUIM

eLOC

eSAR

eTMD

eRMS

eSWIOMA

eSWIM2MCMD

eSWIOMAEXT

eSWILOC

9.2.3.3 enum eTimeout

eTimeout

Enumerator

eTIMEOUT_2_S

eTIMEOUT_5_S

eTIMEOUT_8_S

eTIMEOUT_10_S

eTIMEOUT_20_S

eTIMEOUT_30_S

eTIMEOUT_60_S

eTIMEOUT_300_S

eTIMEOUT_DEFAULT

9.2.3.4 enum msgtype

qmi message type

Enumerator

eREQ

eRSP

eIND

9.2.4 Function Documentation

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

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

9.2.4.3 char* get_version ( )

Returns

version string

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

extract msgid/xid/type from modem reply

Parameters

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

Returns

qmi message string

9.2.4.5 uint16_t helper_get_xid (uint8_t * *qmi_resp*)

9.2.4.6 int helper_set_log_func (logger *func*)

set log function

9.2.4.7 int helper_set_log_lvl (uint8_t *lvl*)

set log level

9.2.4.8 char* liteqmi_GetVersion ( )

Returns

version string

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

9.2.4.10 unsigned unpack_result_code_only (uint8_t * *pMdmResp*)

common handler for unpacking response with TLV type 0x02 only

9.2.5 Variable Documentation

9.2.5.1 logger glog

9.2.5.2 uint8_t gloglvl

9.3 dms.h File Reference

Data Structures

- struct unpack_dms_GetModelID_t
- struct unpack_dms_GetIMSI_t
- struct unpack_dms_GetFirmwareInfo_t
- struct unpack_dms_GetPower_t
- struct unpack_dms_GetSerialNumbers_t
- struct unpack_dms_GetHardwareRevision_t
- struct unpack_dms_SLQSGetBandCapability_t
- struct unpack_dms_GetDeviceCapabilities_t
- struct unpack_dms_GetFirmwareRevisions_t
- struct unpack_dms_GetFirmwareRevision_t
- struct unpack_dms_GetDeviceSerialNumbers_t
- struct unpack_dms_GetPRLVersion_t
- struct unpack_dms_GetNetworkTime_t
- struct unpack_dms_GetVoiceNumber_t
- struct unpack_dms_GetDeviceHardwareRev_t
- struct unpack_dms_GetFSN_t
- struct unpack_dms_GetDeviceCap_t
- struct pack_dms_SetPower_t
- struct unpack_dms_SetPower_t
- struct unpack_dms_GetBandCapability_t
- struct unpack_dms_GetUSBComp_t
- struct pack_dms_SetUSBComp_t
- struct unpack_dms_SetUSBComp_t
- struct pack_dms_SetCustFeature_t
- struct unpack_dms_SetCustFeature_t
- struct unpack_dms_GetCustFeature_t
- struct unpack_dms_SetFirmwarePreference_t
- struct unpack_dms_GetCrashAction_t
- struct pack_dms_SetCrashAction_t
- struct unpack_dms_SetCrashAction_t
- struct unpack_dms_GetDeviceMfr_t
- struct pack_dms_SetEventReport_t
- struct unpack_dms_SetEventReport_t
- struct dms_OperatingModeTlv
- struct dms_ActivationStatusTlv
- struct unpack_dms_SetEventReport_ind_t
- struct pack_dms_UIMGetICCID_t
- struct unpack_dms_UIMGetICCID_t

- struct pack_dms_SetCustFeaturesV2_t
- struct unpack_dms_SetCustFeaturesV2_t
- struct pack_dms_GetCustFeaturesV2_t
- struct DMSgetCustomInput
- struct DMScustSettingInfo
- struct DMScustSettingList
- struct DMSgetCustomFeatureV2
- struct unpack_dms_GetCustFeaturesV2_t
- struct unpack_dms_GetActivationState_t
- struct image_info_t
- struct unpack_dms_SLQSSwiGetFirmwareCurr_t
- struct pack_dms_SLQSSwiSetDyingGaspCfg_t
- struct unpack_dms_SLQSSwiSetDyingGaspCfg_t
- struct unpack_dms_SLQSSwiClearDyingGaspStatistics_t
- struct packgetDyingGaspStatistics
- struct unpack_dms_SLQSSwiGetDyingGaspStatistics_t
- struct packgetDyingGaspCfg
- struct unpack_dms_SLQSSwiGetDyingGaspCfg_t
- struct unpack_dms_SLQSDmsSwiGetResetInfo_t
- struct unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t
- struct pack_dms_SLQSDmsSwiIndicationRegister_t
- struct unpack_dms_SLQSDmsSwiIndicationRegister_t
- struct unpack_dms_SLQSSwiGetFwUpdateStatus_t
- struct unpack_dms_GetManufacturer_t
- struct unpack_dms_GetOfflineReason_t
- struct pack_dms_SetActivationStatusCallback_t
- struct unpack_dms_SetActivationStatusCallback_t
- struct pack_dms_UIMSetPINProtection_t
- struct unpack_dms_UIMSetPINProtection_t
- struct pack_dms_UIMUnblockPIN_t
- struct pack_dms_UIMVerifyPIN_t
- struct pack_dms_UIMChangePIN_t
- struct pack_dms_UIMGetControlKeyStatus_t
- struct unpack_dms_UIMGetControlKeyStatus_t
- struct unpack_dms_UIMGetPINStatus_t
- struct pack_dms_UIMSetControlKeyProtection_t
- struct unpack_dms_UIMSetControlKeyProtection_t
- struct pack_dms_UIMUnblockControlKey_t
- struct unpack_dms_UIMUnblockControlKey_t
- struct pack_dms_ResetToFactoryDefaults_t
- struct unpack_dms_ResetToFactoryDefaults_t
- struct pack_dms_ActivateAutomatic_t
- struct eriDataparams
- struct unpack_dms_SLQSGetERIFile_t
- struct unpack_dms_SLQSUIMGetState_t
- struct pack_dms_SLQSSwiGetCrashInfo_t
- struct crashInformation
- struct crashInfoParams
- struct unpack_dms_SLQSSwiGetCrashInfo_t
- struct unpack_dms_SLQSSwiGetHostDevInfo_t
- struct pack_dms_SLQSSwiSetHostDevInfo_t
- struct unpack_dms_SLQSSwiSetHostDevInfo_t
- struct unpack_dms_SLQSSwiGetOSInfo_t
- struct pack_dms_SLQSSwiSetOSInfo_t
- struct unpack_dms_SLQSSwiSetOSInfo_t
- struct unpack_dms_SLQSSwiGetSerialNoExt_t

Macros

- #define DMS_UINT8_MAX_STRING_SZ 255
- #define DMS_MAX_CUST_ID_LEN 64
- #define DMS_MAX_CUST_VALUE_LEN 8
- #define DMS_IMGDETAILS_LEN 16
- #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255
- #define DMS_MAX_FWUPDATE_REF_STR_SZ 15
- #define DMS_SLQSFWINFO_MODELID_SZ 20
- #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85
- #define DMS_SLQSFWINFO_APPVERSION_SZ 85
- #define DMS_SLQSFWINFO_SKU_SZ 15
- #define DMS_SLQSFWINFO_PACKAGEID_SZ 85
- #define DMS_SLQSFWINFO_CARRIER_SZ 20
- #define DMS_SLQSFWINFO_PRIVERSION_SZ 16
- #define DMS_SLQSFWINFO_CUR_CARR_NAME 17
- #define DMS_SLQSFWINFO_CUR_CARR_REV 13
- #define MAX_BUILD_ID_LEN 255
- #define UNIQUE_ID_LEN 16
- #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160
- #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20
- #define SPC_SIZE 6
- #define CK_MAX_SIZE 8
- #define ACT_CODE_MAX_SIZE 81
- #define ERI_DATA_MAX_SIZE 1024
- #define MEID_MAX_SIZE 8
- #define DMS_PM_ONLINE 0x00 /∗ Online ∗/
- #define DMS_PM_LOW 0x01 /∗ Low Power ∗/
- #define DMS_PM_FACTORY 0x02 /∗ Factory Test Mode ∗/
- #define DMS_PM_OFFLINE 0x03 /∗ Offline ∗/
- #define DMS_PM_RESET 0x04 /∗ Reset ∗/
- #define DMS_PM_SHUT_DOWN 0x05 /∗ Shut Down ∗/
- #define DMS_PM_PERSISTENT_LOW 0x06 /∗ Persistent Low Power ∗/
- #define DMS_SET_REPORT_ENABLE 1
- #define DMS_SET_REPORT_DISABLE 0
- #define DMS_SWI_SET_IND_ENABLE 1
- #define DMS_SWI_SET_IND_DISABLE 0

Functions

- int pack_dms_GetIMSI (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetIMSI (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetIMSI_t ∗pOutput)
- int pack_dms_GetModelID (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetModelID (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetModelID_t ∗pOutput)
- int pack_dms_GetFirmwareInfo (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetFirmwareInfo (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetFirmwareInfo_t ∗p-Output)
- 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)

9.3.1 Macro Definition Documentation

9.3.1.1 #define ACT_CODE_MAX_SIZE 81

9.3.1.2 #define CK_MAX_SIZE 8

9.3.1.3 #define DMS_IMGDETAILS_LEN 16

9.3.1.4 #define DMS_MAX_CUST_ID_LEN 64

9.3.1.5 #define DMS_MAX_CUST_VALUE_LEN 8

9.3.1.6 #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255

9.3.1.7 #define DMS_MAX_FWUPDATE_REF_STR_SZ 15

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

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

9.3.1.10 #define DMS_PM_OFFLINE 0x03 /* Offline */

9.3.1.11 #define DMS_PM_ONLINE 0x00 /* Online */

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

9.3.1.13 #define DMS_PM_RESET 0x04 /* Reset */

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

9.3.1.15 #define DMS_SET_REPORT_DISABLE 0

9.3.1.16 #define DMS_SET_REPORT_ENABLE 1

9.3.1.17 #define DMS_SLQSFWINFO_APPVERSION_SZ 85

9.3.1.18 #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85

9.3.1.19 #define DMS_SLQSFWINFO_CARRIER_SZ 20

9.3.1.20 #define DMS_SLQSFWINFO_CUR_CARR_NAME 17

9.3.1.21 #define DMS_SLQSFWINFO_CUR_CARR_REV 13

9.3.1.22 #define DMS_SLQSFWINFO_MODELID_SZ 20

9.3.1.23 #define DMS_SLQSFWINFO_PACKAGEID_SZ 85

9.3.1.24 #define DMS_SLQSFWINFO_PRIVERSION_SZ 16

9.3.1.25 #define DMS_SLQSFWINFO_SKU_SZ 15

9.3.1.26 #define DMS_SWI_SET_IND_DISABLE 0

9.3.1.27 #define DMS_SWI_SET_IND_ENABLE 1

9.3.1.28 #define DMS_UINT8_MAX_STRING_SZ 255

9.3.1.29 #define ERI_DATA_MAX_SIZE 1024

9.3.1.30 #define MAX_BUILD_ID_LEN 255

9.3.1.31 #define MEID_MAX_SIZE 8

9.3.1.32 #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160

9.3.1.33 #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20

9.3.1.34 #define SPC_SIZE 6

9.3.1.35 #define UNIQUE_ID_LEN 16

9.3.2 Function Documentation

9.3.2.1 int pack_dms_ActivateAutomatic (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_ActivateAutomatic_t * *pReq*)

To set UIM Unblock Control Key pack

Parameters

in,out	*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

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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

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

Get Crash Action pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.3.2.5 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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

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

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

Get Device Capabilities pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.3.2.8 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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

Get Hardware Revision pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

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

Get Manufacture pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

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

get Device Serial Number pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.3.2.12 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

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

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

Get FSN pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

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

get hardware revision 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

9.3.2.17 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

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

Get Network Time pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.3.2.21 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 power 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

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

Get PRL Versions pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.3.2.24 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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

Get USB Comp pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

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

Get Voice Number pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.3.2.27 int pack_dms_ResetToFactoryDefaults (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen,* pack_dms_ResetToFactoryDefaults_t * *pReq*)

To set UIM Unblock Control Key pack

Parameters

in,out	*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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

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

Set Power pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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_SetUSBComp (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetUSBComp_t * *reqArg*)

Set USB Comp pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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_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

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

Set the registration state for different indication pack

Parameters

in,out	*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

Note

support EM/MC74xx onwards

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

get band capability pack

Parameters

in,out	*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

9.3.2.39 int pack_dms_SLQSGetERIFile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To set UIM Get ERI file pack

Parameters

in,out	*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

9.3.2.40 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

9.3.2.41 int pack_dms_SLQSSwiGetCrashInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SLQSSwiGetCrashInfo_t * *pReq*)

To set Get Crash Info pack

Parameters

in,out	*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

9.3.2.42 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

9.3.2.43 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

9.3.2.44 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

9.3.2.45 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

9.3.2.46 int pack_dms_SLQSSwiGetHostDevInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To set Host Dev 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

9.3.2.47 int pack_dms_SLQSSwiGetOSInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To set 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

9.3.2.48 int pack_dms_SLQSSwiGetSerialNoExt (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To set 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

9.3.2.49 int pack_dms_SLQSSwiSetDyingGaspCfg (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SLQSSwiSetDyingGaspCfg_t * *reqArg*)

Set Dying GASP Config pack

Parameters

in,out	*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_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

9.3.2.51 int pack_dms_SLQSSwiSetOSInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SLQSSwiSetOSInfo_t * *pReq*)

To set Host OS Info pack

Parameters

in,out	*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

9.3.2.52 int pack_dms_SLQSUIMGetState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To set UIM Get State 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

9.3.2.53 int pack_dms_UIMChangePIN (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_UIMChangePIN_t * *pReq*)

To set UIM change PIN pack

Parameters

in,out	*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

9.3.2.54 int pack_dms_UIMGetControlKeyStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_UIMGetControlKeyStatus_t * *pReq*)

To set UIM Get Control Key Status pack

Parameters

in,out	*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_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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.3.2.56 int pack_dms_UIMGetPINStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To set UIM Get PIN Status pack

Parameters

in,out	*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_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

9.3.2.58 int pack_dms_UIMSetPINProtection (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_UIMSetPINProtection_t * *pReq*)

To set UIM PIN protection pack

Parameters

in,out	*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_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

9.3.2.60 int pack_dms_UIMUnblockPIN (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_UIMUnblockPIN_t * *pReq*)

To set UIM unblock PIN pack

Parameters

in,out	*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

9.3.2.61 int pack_dms_UIMVerifyPIN (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_UIMVerifyPIN_t * *pReq*)

To set UIM verify PIN pack

Parameters

in,out	*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

9.3.2.62 int pack_dms_ValidateSPC (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_ResetToFactoryDefaults_t * *pReq*)

To set UIM Unblock Control Key pack

Parameters

in,out	*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 unpack_dms_ActivateAutomatic (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_ResetToFactoryDefaults_t * *pOutput*)

To get UIM Unblock Control Key unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.64 int unpack_dms_GetActivationState (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetActivationState_t * *pOutput*)

Get Activation State unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.65 int unpack_dms_GetBandCapability (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetBandCapability_t * *pOutput*)

Get Band Capabilities unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.66 int unpack_dms_GetCrashAction (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetCrashAction_t * *pOutput*)

Get Crash Action unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.67 int unpack_dms_GetCustFeature (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetCustFeature_t * *pOutput*)

Get Custom Feature unpack. This API is deprecated for EM74xx/MC74xx, please use unpack_dms_GetCustFeaturesV2() instead for EM74xx/MC74xx.

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

9.3.2.69 int unpack_dms_GetDeviceCap (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceCap_t * *pOutput*)

Get Device Capabilities unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.70 int unpack_dms_GetDeviceCapabilities (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceCapabilities_t * *pOutput*)

get device capability unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.71 int unpack_dms_GetDeviceHardwareRev (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceHardwareRev_t * *pOutput*)

Get Hardware Revision unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.72 int unpack_dms_GetDeviceMfr (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceMfr_t * *pOutput*)

Get Manufacture unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.73 int unpack_dms_GetDeviceSerialNumbers (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceSerialNumbers_t * *pOutput*)

get Device Serial Number unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.74 int unpack_dms_GetFirmwareInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFirmwareInfo_t * *pOutput*)

get firmware info unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.75 int unpack_dms_GetFirmwareRevision (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFirmware-Revision_t * *pOutput*)

get Firmware Revision unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.76 int unpack_dms_GetFirmwareRevisions (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFirmware-Revisions_t * *pOutput*)

get Firmware Revisions unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.77 int unpack_dms_GetFSN (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFSN_t * *pOutput*)

Get FSN unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.78 int unpack_dms_GetHardwareRevision (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetHardwareRevision_t * *pOutput*)

get hardware revision unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.79 int unpack_dms_GetIMSI (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetIMSI_t * *pOutput*)

get IMSI unpack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx-_xx_xx and all EM74xx firmware versions. Please use unpack_uim_ReadTransparent()(EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

To get device manufacturer information unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.81 int unpack_dms_GetModelID (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetModelID_t * *pOutput*)

get model id unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.82 int unpack_dms_GetNetworkTime (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetNetworkTime_t * *pOutput*)

Get Network Time unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

To get operating mode offline reason unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.84 int unpack_dms_GetPower (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetPower_t * *pOutput*)

get power unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.85 int unpack_dms_GetPRLVersion (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetPRLVersion_t * *pOutput*)

Get PRL Versions unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.86 int unpack_dms_GetSerialNumbers (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetSerialNumbers_t * *pOutput*)

get serial numbers unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.87 int unpack_dms_GetUSBComp (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetUSBComp_t * *pOutput*)

Get USB Comp unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.88 int unpack_dms_GetVoiceNumber (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetVoiceNumber_t * *pOutput*)

Get Voice Number unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.89 int unpack_dms_ResetToFactoryDefaults (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_ResetToFactoryDefaults_t * *pOutput*)

To get UIM Unblock Control Key unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.90 int unpack_dms_SetActivationStatusCallback (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetActivationStatusCallback_t * *pOutput*)

Set Activation status unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.91 int unpack_dms_SetCrashAction (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetCrashAction_t * *pOutput*)

Set Crash Action unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	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

9.3.2.92 int unpack_dms_SetCustFeature (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetCustFeature_t * *pOutput*)

Set Custom Feature unpack. This API is deprecated for EM74xx/MC74xx, please use unpack_dms_SetCustFeaturesV2() instead for EM74xx/MC74xx.

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.93 int unpack_dms_SetCustFeaturesV2 (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetCustFeaturesV2_t * *pOutput*)

Set Cust features unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.94 int unpack_dms_SetEventReport (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetEventReport_t * *pOutput*)

Set Event Report unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.95 int unpack_dms_SetEventReport_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetEventReport_ind_t * *pOutput*)

Event Report Indication unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.96 int unpack_dms_SetFirmwarePreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetFirmware-Preference_t * *pOutput*)

Set Firmware Preference unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.97 int unpack_dms_SetPower (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetPower_t * *pOutput*)

Set Power unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.98 int unpack_dms_SetUSBComp (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetUSBComp_t * *pOutput*)

Set USB Comp unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.99 int unpack_dms_SLQSDmsSwiGetResetInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSDmsSwiGetResetInfo_t * *pOutput*)

To get reset info unpack

Parameters

in,out	*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

9.3.2.100 int unpack_dms_SLQSDmsSwiGetResetInfo_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t * *pOutput*)

DMS reset info Indication unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	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

9.3.2.101 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	*pespLen*	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

9.3.2.102 int unpack_dms_SLQSGetBandCapability (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSGetBandCapability_t * *pOutput*)

get band capability unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.103 int unpack_dms_SLQSGetERIFile (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSGetERIFile_t * *pOutput*)

To get UIM Get ERI file unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.104 int unpack_dms_SLQSSwiClearDyingGaspStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiClearDyingGaspStatistics_t * *pOutput*)

Clear Dying GASP Statistics unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.105 int unpack_dms_SLQSSwiGetCrashInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetCrashInfo_t * *pOutput*)

To get Crash Info unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.106 int unpack_dms_SLQSSwiGetDyingGaspCfg (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetDyingGaspCfg_t * *pOutput*)

Get Dying GASP Config unpack

Parameters

in,out	*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

9.3.2.107 int unpack_dms_SLQSSwiGetDyingGaspStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetDyingGaspStatistics_t * *pOutput*)

Get Dying GASP Statistics unpack

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

9.3.2.108 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.109 int unpack_dms_SLQSSwiGetFwUpdateStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetFwUpdateStatus_t * *pOutput*)

To get Firmware Update status unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.110 int unpack_dms_SLQSSwiGetHostDevInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetHostDevInfo_t * *pOutput*)

To get Host Dev Info unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.111 int unpack_dms_SLQSSwiGetOSInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetOSInfo_t * *pOutput*)

To get Host OS Info unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.112 int unpack_dms_SLQSSwiGetSerialNoExt (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetSerialNoExt_t * *pOutput*)

To get serial number extension unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.113 int unpack_dms_SLQSSwiSetDyingGaspCfg (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiSetDyingGaspCfg_t * *pOutput*)

Set Dying GASP Config unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.114 int unpack_dms_SLQSSwiSetHostDevInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiSetHostDevInfo_t * *pOutput*)

To get Host Dev Info unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.115 int unpack_dms_SLQSSwiSetOSInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiSetOSInfo_t * *pOutput*)

To get Host OS Info unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	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_SLQSUIMGetState (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSUIMGetState_t * *pOutput*)

To get UIM Get State unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.117 int unpack_dms_UIMChangePIN (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMSetPINProtection_t * *pOutput*)

To get UIM change PIN unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.118 int unpack_dms_UIMGetControlKeyStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMGetControlKeyStatus_t * *pOutput*)

To get UIM Get ControlKeyStatus unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

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

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.120 int unpack_dms_UIMGetPINStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMGetPINStatus_t * *pOutput*)

To get UIM Get PIN Status unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.121 int unpack_dms_UIMSetControlKeyProtection (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMSetControlKeyProtection_t * *pOutput*)

To get UIM Set Control Key Protection unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.122 int unpack_dms_UIMSetPINProtection (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMSetPIN-Protection_t * *pOutput*)

To get UIM Set PIN protection unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.123 int unpack_dms_UIMUnblockControlKey (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMUnblock-ControlKey_t * *pOutput*)

To get UIM Unblock Control Key unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.124 int unpack_dms_UIMUnblockPIN (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMSetPINProtection_t * *pOutput*)

To get UIM unblock PIN unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.125 int unpack_dms_UIMVerifyPIN (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMSetPINProtection_t * *pOutput*)

To get UIM verify PIN unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.126 int unpack_dms_ValidateSPC (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_ResetToFactoryDefaults_t * *pOutput*)

To get UIM Unblock Control Key unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	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 • See FMSImageList
in,out	*pValid-CombinationSize*	• number of combination passed in and returned
out	*pValid-Combinations*	• valid combinations returned • See CarrierImage_t

Returns

- eQCWWAN_ERR_INVALID_ARG - Invalid parameters
- eQCWWAN_ERR_BUFFER_SZ - No enough element to store combinatons returned

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.4.2.2 int pack_fms_GetImagesPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_GetImagesPreference_t * *reqArg*)

Get Images Preference pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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	*pespLen*	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	*pespLen*	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	*pespLen*	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

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

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)

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_SENSOR_DATA_LEN 64

9.5.1.29 #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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.2 int pack_loc_EventRegister (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen,* pack_loc_EventRegister_t * *reqArg*)

Event Register pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.3 int pack_loc_SetExtPowerState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SetExtPowerState_t * *reqArg*)

Set Ext Power State pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

Set Operation Mode pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.5 int pack_loc_SLQSLOCGetBestAvailPos (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCGetBestAvailPos_t * *reqArg*)

Get Best Avail position pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.6 int pack_loc_SLQSLOCInjectPosition (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectPosition_t * *reqArg*)

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

9.5.3.7 int pack_loc_SLQSLOCInjectSensorData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectSensorData_t * *reqArg*)

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_SLQSLOCInjectUTCTime (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCInjectUTCTime_t * reqArg)

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_SLQSLOCSetCradleMountConfig (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCSetCradleMountConfig_t * reqArg)

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_Start (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Start_t * reqArg)

LOC Start pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.11 int pack_loc_Stop (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen,* pack_loc_Stop_t * *reqArg*)

Loc Stop pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.12 int unpack_loc_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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.13 int unpack_loc_DeleteAssistData (uint8_t * *pResp,* uint16_t *respLen,* unpack_loc_Delete_Assist_Data_t * *pOutput*)

Delete Assistant Data unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.14 int unpack_loc_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	*pespLen*	qmi response length
out	*pOutput*	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_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	*pespLen*	qmi response length
out	*pOutput*	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_EventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EventRegister_t * *pOutput*)

Event Register unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	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_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	*pespLen*	qmi response length
out	*pOutput*	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_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	*pespLen*	qmi response length
out	*pOutput*	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_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	*pespLen*	qmi response length
out	*pOutput*	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_SetExtPowerState (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerState_t * *pOutput*)

Set Ext Power State unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	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_SetOperationMode (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetOperationMode_t * *pOutput*)

Set Operation Mode unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	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_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	*pespLen*	qmi response length
out	*pOutput*	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_SLQSLOCGetBestAvailPos (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCGetBestAvailPos_t * *pOutput*)

Get Best Avail position unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	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_SLQSLOCInjectPosition (uint8_t * *pResp*, uint16_t *respLen*)

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

9.5.3.25 int unpack_loc_SLQSLOCInjectSensorData (uint8_t * *pResp*, uint16_t *respLen*)

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

9.5.3.26 int unpack_loc_SLQSLOCInjectUTCTime (uint8_t * *pResp*, uint16_t *respLen*)

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

9.5.3.27 int unpack_loc_SLQSLOCSetCradleMountConfig (uint8_t * *pResp*, uint16_t *respLen*)

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

9.5.3.28 int unpack_loc_Start (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Start_t * *pOutput*)

Loc Start unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	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_Stop (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Stop_t * *pOutput*)

Loc Stop unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6 nas.h File Reference

Data Structures

- struct unpack_nas_GetSignalStrengths_t
- struct unpack_nas_SLQSGetSysSelectionPref_t
- struct nas_netSelectionPref
- struct nas_acqOrderPref
- struct nas_CSGID
- struct pack_nas_SLQSSetSysSelectionPref_t
- struct pack_nas_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

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_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_nas_SLQSNasSigInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSigInfoCallback_ind_t *pOutput)
- int unpack_nas_GetHomeNetwork (uint8_t *pResp, uint16_t respLen, unpack_nas_GetHomeNetwork_t *pOutput)
- int pack_nas_GetHomeNetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int pack_nas_SLQSGetSysInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetSysInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfo_t *pOutput)
- int unpack_nas_SLQSNasSysInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSysInfoCallback_ind_t *pOutput)
- int pack_nas_GetServingNetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetwork (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t *pOutput)
- int pack_nas_GetServingNetworkCapabilities (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetworkCapabilities (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetworkCapabilities_t *pOutput)
- int pack_nas_PerformNetworkScan (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_PerformNetworkScan (uint8_t *pResp, uint16_t respLen, unpack_nas_PerformNetworkScan_t *pOutput)
- int pack_nas_SLQSSwiGetLteCQI (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetLteCQI (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteCQI_t *pOutput)
- int pack_nas_SLQSNasSwiModemStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasSwiModemStatus (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiModemStatus_t *pOutput)
- int pack_nas_SLQSGetServingSystem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetServingSystem (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetServingSystem_t *pOutput)
- int pack_nas_SLQSGetSignalStrength (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint16_t reqMask)
- int unpack_nas_SLQSGetSignalStrength (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSignalStrength_t *pOutput)
- int pack_nas_SLQSSetSignalStrengthsCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSignalStrengthsCallback_t *pReqParam)
- int unpack_nas_SLQSSetSignalStrengthsCallback (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SetRFInfoCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetRFInfoCallback (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SetLURejectCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetLURejectCallback (uint8_t *pResp, uint16_t respLen)

- int unpack_nas_SetEventReportInd (uint8_t *pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t *pOutput)
- int pack_nas_GetCDMANetworkParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetCDMANetworkParameters (uint8_t *pResp, uint16_t respLen, unpack_nas_GetCDMANetworkParameters_t *pOutput)
- int pack_nas_GetANAAAAuthenticationStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetANAAAAuthenticationStatus (uint8_t *pResp, uint16_t respLen, uint32_t *pAuthStatus)
- int pack_nas_GetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetACCOLC (uint8_t *pResp, uint16_t respLen, uint8_t *pAccolc)
- int pack_nas_SetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetACCOLC_t reqParam)
- int unpack_nas_SetACCOLC (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasConfigSigInfo2_t *pReqParam)
- int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t *pResp, uint16_t respLen)
- int unpack_nas_SetDataCapabilitiesCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetDataCapabilitiesCallback_ind_t *pOutput)
- int pack_nas_GetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_GetNetworkPreference_t *pOutput)
- int pack_nas_SetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetNetworkPreference_t *reqArg)
- int unpack_nas_SetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t *pOutput)
- int unpack_nas_SetRoamingIndicatorCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetRoamingIndicatorCallback_ind_t *pOutput)
- int unpack_nas_SetServingSystemCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t *pOutput)
- int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SlqsGetLTECphyCAInfo_t *pOutput)
- int unpack_nas_SLQSSetSysSelectionPrefCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t *pOutput)
- int unpack_nas_SLQSNasSwiOTAMessageCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t *pOutput)
- int pack_nas_SLQSInitiateNetworkRegistration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSInitiateNetworkRegistration_t *pReqParam)
- int unpack_nas_SLQSInitiateNetworkRegistration (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSNasSwiIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasSwiIndicationRegister_t *pReqParam)
- int unpack_nas_SLQSNasSwiIndicationRegister (uint8_t *pResp, uint16_t respLen)
- int pack_nas_SLQSGetPLMNName (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSGetPLMNName_t *reqArg)
- int unpack_nas_SLQSGetPLMNName (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetPLMNName_t *pOutput)
- int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetCellLocationInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetCellLocationInfo_t *pOutput)
- int pack_nas_SLQSGetNetworkTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetNetworkTime (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetNetworkTime_t *pOutput)
- int unpack_nas_SLQSNasNetworkTimeCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t *pOutput)
- int unpack_nas_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)

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*)

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

9.6.3.2 int pack_nas_GetANAAAAuthenticationStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.3 int pack_nas_GetCDMANetworkParameters (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.4 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	*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

9.6.3.5 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	*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

9.6.3.6 int pack_nas_GetNetworkPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

9.6.3.7 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	*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

9.6.3.8 int pack_nas_GetServingNetwork (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.9 int pack_nas_GetServingNetworkCapabilities (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.10 int pack_nas_GetSignalStrengths (pack_qmi_t * *pCtx*, uint8_t * *pReq*, uint16_t * *pLen*)

get signal strengths 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

9.6.3.11 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	*pReq*	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

9.6.3.12 int pack_nas_PerformNetworkScan (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.13 int pack_nas_SetACCOLC (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetACCOLC_t *reqParam*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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	*pReq*	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

9.6.3.15 int pack_nas_SetLURejectCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pBenable*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
in	*bEnable*	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*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
in	*pack*	default prototype

9.6.3.17 int pack_nas_SetRFInfoCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
in	*bEnable*	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	*pReq*	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	*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

9.6.3.20 int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

9.6.3.21 int pack_nas_SLQSGetNetworkTime (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

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	*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

9.6.3.23 int pack_nas_SLQSGetPLMNName (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSGetPLMNName_t * reqArg)

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*)

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

9.6.3.25 int pack_nas_SLQSGetSignalStrength (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint16_t *reqMask*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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*)

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

9.6.3.27 int pack_nas_SLQSGetSysSelectionPref (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.28 int pack_nas_SLQSInitiateNetworkRegistration (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSInitiateNetworkRegistration_t * *pReqParam*)

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

9.6.3.29 int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasConfigSigInfo2_t * *pReqParam*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.6.3.30 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*)

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_SLQSNasGetHDRColorCode (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

HDR Color Code Parameters 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

9.6.3.33 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	*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

9.6.3.34 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

9.6.3.35 int pack_nas_SLQSNasIndicationRegisterExt (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasIndicationRegisterExt_t * *pReqParam*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.6.3.36 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	*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

9.6.3.37 int pack_nas_SLQSNasSwiIndicationRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasSwiIndicationRegister_t * pReqParam)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.6.3.38 int pack_nas_SLQSNasSwiModemStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.39 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	*pReq*	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

9.6.3.40 int pack_nas_SLQSSetBandPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetBandPreference_t * pReqParam)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.6.3.41 int pack_nas_SLQSSetSignalStrengthsCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSignalStrengthsCallback_t * pReqParam)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.6.3.42 int pack_nas_SLQSSetSysSelectionPref (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSysSelectionPref_t * pReqParam)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.6.3.43 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	*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

9.6.3.44 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	*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

9.6.3.45 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	*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

9.6.3.46 int pack_nas_SLQSSwiGetLteCQI (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.47 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

9.6.3.48 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	*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

9.6.3.49 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	*pReq*	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

9.6.3.50 int unpack_nas_GetACCOLC (uint8_t * *pResp*, uint16_t *respLen*, uint8_t * *pAccolc*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pAccolc*	accolc

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.51 int unpack_nas_GetANAAAAuthenticationStatus (uint8_t * *pResp*, uint16_t *respLen*, uint32_t * *pAuthStatus*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*authStatus*	auth status

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.52 int unpack_nas_GetCDMANetworkParameters (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetCDMANetworkParameters_t * *pOutput*)

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

9.6.3.53 int unpack_nas_GetHomeNetwork (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetHomeNetwork_t * *pOutput*)

get home network unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.54 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.55 int unpack_nas_GetNetworkPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetNetwork-Preference_t * *pOutput*)

9.6.3.56 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.57 int unpack_nas_GetServingNetwork (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetServingNetwork_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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_GetServingNetworkCapabilities (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetServingNetworkCapabilities_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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_GetSignalStrengths (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetSignalStrengths_t * *pOutput*)

get signal strengths unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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_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

9.6.3.61 int unpack_nas_PerformNetworkScan (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_PerformNetworkScan_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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_SetACCOLC (uint8_t * *pResp*, uint16_t *respLen*)

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

9.6.3.63 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

9.6.3.64 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	*pOuput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.65 int unpack_nas_SetEventReportInd (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetEventReportInd_t * *pOutput*)

9.6.3.66 int unpack_nas_SetLURejectCallback (uint8_t * *pResp*, uint16_t *respLen*)

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_SetNasLTECphyCaIndCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNasLTECphyCaIndCallback_ind_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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

9.6.3.68 int unpack_nas_SetNetworkPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNetwork-Preference_t * *pOutput*)

9.6.3.69 int unpack_nas_SetRFInfoCallback (uint8_t * *pResp*, uint16_t *respLen*)

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

9.6.3.70 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	*pOuput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.71 int unpack_nas_SetServingSystemCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetServingSystemCallback_ind_t * *pOutput*)

9.6.3.72 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

9.6.3.73 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.74 int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SlqsGetLTECphyCAInfo_t * *pOutput*)

9.6.3.75 int unpack_nas_SLQSGetNetworkTime (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetNetworkTime_t * *pOutput*)

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

9.6.3.76 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.77 int unpack_nas_SLQSGetPLMNName (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetPLMNName_t * *pOutput*)

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

9.6.3.78 int unpack_nas_SLQSGetServingSystem (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetServingSystem_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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_SLQSGetSignalStrength (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSignalStrength_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.80 int unpack_nas_SLQSGetSysInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysInfo_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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_SLQSGetSysSelectionPref (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysSelectionPref_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.82 int unpack_nas_SLQSInitiateNetworkRegistration (uint8_t * *pResp*, uint16_t *respLen*)

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

9.6.3.83 int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t * *pResp*, uint16_t *respLen*)

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

9.6.3.84 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

9.6.3.85 int unpack_nas_SLQSNasGetCellLocationInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetCellLocationInfo_t * *pOutput*)

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

9.6.3.86 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.87 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.88 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

9.6.3.89 int unpack_nas_SLQSNasIndicationRegisterExt (uint8_t * *pResp*, uint16_t *respLen*)

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

9.6.3.90 int unpack_nas_SLQSNasNetworkTimeCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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

9.6.3.91 int unpack_nas_SLQSNasSigInfoCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasSigInfoCallback_ind_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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

9.6.3.92 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	*pOuput*	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_SLQSNasSwiIndicationRegister (uint8_t * *pResp*, uint16_t *respLen*)

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

9.6.3.94 int unpack_nas_SLQSNasSwiModemStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasSwiModemStatus_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.95 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	*pOuput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.96 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

9.6.3.97 int unpack_nas_SLQSNasSysInfoCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSysInfoCallback_ind_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.98 int unpack_nas_SLQSNasTimerCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasTimerCallback_ind_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	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

9.6.3.99 int unpack_nas_SLQSSetBandPreference (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

9.6.3.100 int unpack_nas_SLQSSetSignalStrengthsCallback (uint8_t * *pResp*, uint16_t *respLen*)

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_SLQSSetSysSelectionPref (uint8_t * *pResp*, uint16_t *respLen*)

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

9.6.3.102 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	*pOuput*	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_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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.104 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	*pOuput*	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_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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.106 int unpack_nas_SLQSSwiGetLteCQI (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetLteCQI_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.107 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.108 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.109 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

9.7 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

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)

9.7.1 Function Documentation

9.7.1.1 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	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.2 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	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.3 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	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.4 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	*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

9.7.1.5 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	*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

9.7.1.6 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	*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

9.7.1.7 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.8 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.9 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.10 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

9.7.1.11 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

9.7.1.12 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

9.8 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 QmiCbkCatEventStatusReportInd *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

pGetAudioVolTL-BCfgReq[IN]	• See GetAudioVolTLBConfigReq for more information
pGetAudioVolTL-BCfgResp[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 SLQSSetAudioVolTLBConfig (SetAudioVolTLBConfigReq * *pSetAudioVolTLBCfgReq,* SetAudioVolTLBConfigResp * *pSetAudioVolTLBCfgResp*)

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 – 0x01 - Menu Selection – 0x02 - Event DL User activity – 0x03 - Event DL Idle Screen Available – 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 [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 [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

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 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_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 ∗p-RoamingInfo)
- typedef void(∗ tFNBandPreference)(ULONGLONG band_pref)
- typedef void(∗ tFNUSSDRelease)(void)
- typedef void(∗ tFNUSSDNotification)(ULONG type, BYTE ∗pNetworkInfo)
- typedef void(∗ tFNSLQSSignalStrengths)(struct SLQSSignalStrengthsInformation sSLQSSignalStrengths-Info)
- 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 *pQmiCbkUimSlotStatusChangeInd)
- 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)

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)

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_SENSOR_STREAM 0x00000400

9.12.2.20 #define LOC_EVENT_MASK_TIME_SYNC 0x00000800

9.12.2.21 #define LOC_EVENT_POSITION_REPORT 0x00000001

9.12.2.22 #define MAX_MITIGATION_DEV_ID_LEN 255

9.12.2.23 #define MAX_NO_OF_APPLICATIONS 10

9.12.2.24 #define MAX_NO_OF_CALLS 20

9.12.2.25 #define MAX_NO_OF_FILES 255

9.12.2.26 #define MAX_NO_OF_SLOTS 5

9.12.2.27 #define MAX_NO_OF_UUSINFO 20

9.12.2.28 #define MAX_PATH_LENGTH 255

9.12.2.29 #define MAX_RADIO_INTERFACE_LIST 255

9.12.2.30 #define MAXUSSDLENGTH 182

9.12.2.31 #define NAS_SRV 0x02

9.12.2.32 #define NUM_OF_SET 0xFF

9.12.2.33 #define PDS_SRV 0x04

9.12.2.34 #define QMI_ETWS_MAX_PAYLOAD_LENGTH 1254 /∗ Qualcomm defined max ∗/

9.12.2.35 #define QMI_MAX_VOICE_NUMBER_LENGTH 81

9.12.2.36 #define QMI_WMS_MAX_PAYLOAD_LENGTH 256

9.12.2.37 #define REGISTER_EVENT 0x01

9.12.2.38 #define REGISTER_SRV 0x01

9.12.2.39 #define SECOND_INSTANCE 0x01

9.12.2.40 #define SIGSTRENGTH_THRESHOLD_ARR_SZ 5

9.12.2.41 #define THIRD_INSTANCE 0x02

9.12.2.42 #define USSD_DCS_8BIT 0x02 /∗ 8-bit coding scheme ∗/

9.12.2.43 #define USSD_DCS_ASCII 0x01 /∗ ASCII coding scheme ∗/

9.12.2.44 #define USSD_DCS_UCS2 0x03 /∗ UCS2 coding scheme ∗/

9.12.2.45 #define VOICE_SRV 0x08

9.12.2.46 #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. • Values • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
samplesPer-Batch	• number of samples per batch the GNSS location engine is to receive. • samplingFrequency = samplesPerBatch ∗ batchesPerSecond • samplesPerBatch must be a nonzero positive value.
batchPerSec	• LTE NAS version minor • Number of sensor-data batches the GNSS location engine is to receive per second. • 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. • Values • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
samplesPer-Batch	• number of samples per batch the GNSS location engine is to receive. • samplingFrequency = samplesPerBatch ∗ batchesPerSecond • samplesPerBatch must be a nonzero positive value.
batchPerSec	• LTE NAS version minor • Number of sensor-data batches the GNSS location engine is to receive per second. • 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. • Units - Weeks
gpsTimeOf-WeekMs	• Amount of time into the current GPS week. • 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. • Values • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
samplesPer-Batch	• number of samples per batch the GNSS location engine is to receive. • samplingFrequency = samplesPerBatch ∗ batchesPerSecond • samplesPerBatch must be a nonzero positive value.
batchPerSec	• LTE NAS version minor • Number of sensor-data batches the GNSS location engine is to receive per second. • 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. • Values • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
samplesPer-Batch	• number of samples per batch the GNSS location engine is to receive. • samplingFrequency = samplesPerBatch ∗ batchesPerSecond • samplesPerBatch must be a nonzero positive value.
batchPerSec	• LTE NAS version minor • Number of sensor-data batches the GNSS location engine is to receive per second. • 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 • Values: – 0 - unknown – 1 - normal – 2 - high(warning) – 3 - high(critical) – 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 – 1 - Disconnected – 2 - Connected – 3 - Suspended – 4 - Authenticating
reconfigReqd	• Indicates if the network interface on the host needs to be reconfigured – 0 - No need to reconfigure – 1 - Reconfiguration required
sessionEnd-Reason	• See qaGobiApiTableCallEndReasons.h for Call End Reason, 0xFFFF means invalid value

verboseSessnEndReasonType	• Call End Reason Type – 0 - Unspecified – 1 - Mobile IP – 2 - Internal – 3 - Call Manager defined – 6 - 3GPP Specification defined – 7 - PPP – 8 - EHRPD – 9 - IPv6 – 0xFFFF - invalid value
verboseSessnEndReason	• 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 – 4 - IPv4 – 6 - IPv6 – 0xFF - invalid value
techName	• Technology name of the packet data connection. – 32767 - CDMA – 32764 - UMTS – 30592 - EPC – 30590 - EMBMS – 30584 - Modem Link Local – 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. • Range - 1 (highest accuracy) to 50 (lowest accuracy) • PDOP = square root of (Square of HDOP + Square of VDOP2)

HDOP	• Horizontal dilution of precision. • Range - 1 (highest accuracy) to 50 (lowest accuracy)
VDOP	• Vertical dilution of precision. • 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: – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down
source	• entity which initiated the reset or power down, possible values listed below: – 0 - unknown – 1 - user requested – 2 - hardware switch – 3 - temperature critical – 4 - voltage critical – 5 - configuration update – 6 - LWM2M – 7 - OMA-DM – 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 • Values: – QMI_ERR_NONE – No error in the request – QMI_ERR_CAUSE_CODE - SMS cause code – QMI_ERR_MESSAGE_DELIVERY_FAILURE - Message could not be delivered – 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 • Values: – 0x00 - ERROR_CLASS_TEMPORARY – 0x01 - ERROR_CLASS_PERMANENT

RPCause	• GW RP cause
TPCause	• GW TP Cause
msgDelFailure-Type	• Message delivery failure type • Values: – 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY – 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT
msgDelFailure-Cause	• Message delivery failure cause • Values: – 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 NULL, if this event is not present in the smsEventType parameter
pTransferRoute-MTMessageInfo	• pointer to the SMSTransferRouteMTMessageInfo structure . NULL, if this event is not present in the smsEventType parameter
pMessageMode-Info	• pointer to the SMSMessageModeInfo structure NULL, if this event is not present in the smsEventType parameter
pEtwsMessage-Info	• pointer to the SMSEtwsMessageInfo structure NULL, if this event is not present in the smsEventType parameter
pEtwsPlmnInfo	• pointer to the SMSEtwsPlmnInfo structure NULL, if this event is not present in the smsEventType parameter
pSMSCAddress-Info	• pointer to the SMSCAddressInfo structure NULL, if this event is not present in the smsEventType parameter

pSMSOnIMSInfo	• pointer to the SMSOnIMSInfo structure 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. • Following information is associated with each SV ID: – GPS - 1 to 32 – SBAS - 33 to 64 – GLONASS - 65 to 96 – QZSS - 193 to 197 – 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 – 0 - LTE ESM uplink – 1 - LTE ESM downlink – 2 - LTE EMM uplink – 3 - LTE EMM downlink – 4 - GSM/UMTS uplink – 5 - GSM/UMTS downlink
data_len	• OTA Message Content Length

data	• OTA Message Content
pLteNasRelInfo	• LTE NAS Release Info • 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 – 0 - Service not activated – 1 - Service activated – 2 - Activation connecting – 3 - Activation connected – 4 - OTASP security authenticated – 5 - OTASP NAM downloaded – 6 - OTASP MDN downloaded – 7 - OTASP IMSI downloaded – 8 - OTASP PRL downloaded – 9 - OTASP SPC downloaded – 10 - OTASP settings committed

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

pVoiceSetAll-CallStatusCbk-Info	• Call back will populated memory pointed by this parameter when a call is originated, connected, or ended. 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: • 0 - BC0_A - Band Class 0, A-System • 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band • 2 - BC1 - Band Class 1, all blocks • 3 - BC2 - Band Class 2 place holder • 4 - BC3 - Band Class 3, A-System • 5 - BC4 - Band Class 4, all blocks • 6 - BC5 - Band Class 5, all blocks • 7 - GSM_DCS_1800 - GSM DCS band • 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band • 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band • 10 - BC6 - Band Class 6 • 11 - BC7 - Band Class 7 • 12 - BC8 - Band Class 8 • 13 - BC9 - Band Class 9 • 14 - BC10 - Band Class 10 • 15 - BC11 - Band Class 11 • 16 - GSM_450 - GSM 450 band • 17 - GSM_480 - GSM 480 band • 18 - GSM_750 - GSM 750 band • 19 - GSM_850 - GSM 850 band • 20 - GSM_RGSM_900 - GSM Railways GSM Band • 21 - GSM_PCS_1900 - GSM PCS band • 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band • 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band • 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band • 25 - WCDMA_IV_1700 - WCDMA US 1700 band • 26 - WCDMA_V_850 - WCDMA US 850 band • 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band • 28 - BC12 - Band Class 12 • 29 - BC14 - Band Class 14 • 30 - RESERVED_2 - Reserved 2 • 31 - BC15 - Band Class 15 • 32 - 47 - Reserved • 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band • 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band • 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band • 51 to 55 - Reserved • 56 - BBC16 - Band Class 16 • 57 - BC17 - Band Class 17 • 58 - BC18 - Band Class 18 • 59 - BC19 - Band Class 19 • 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 – 16 - Display Text – 17 - Get In-Key – 18 - Get Input – 19 - Setup Menu – 20 - Select Item – 21 - Send SMS - Alpha Identifier – 22 - Setup Event List – 23 - Setup Idle Mode Text – 24 - Language Notification – 25 - Refresh – 26 - End Proactive Session
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. – 1 - GPRS – 2 - EDGE – 3 - HSDPA – 4 - HSUPA – 5 - WCDMA – 6 - CDMA 1xRTT – 7 - CDMA 1xEV-DO Rev 0 – 8 - CDMA 1xEV-DO Rev. A – 9 - GSM – 10 - EVDO Rev. B – 11 - LTE – 12 - HSDPA Plus – 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(∗ tFNEventPosition)(QmiCbkLocPositionReportInd ∗pLocPositionReport)

Position report callback function.

Parameters

pLocPosition-Report[OUT]	• Events related to LOC, see QmiCbkLocPositionReportInd for details

9.12.3.45 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: – eQCWWAN_ERR_NONE - indicates firmware download/switching is successful – eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED - indicates no actual download takes place, this is the case of image switching stored on device – eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE - indicates modem enters firmware download mode, firmware flashing is going to be started. – 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

9.12.3.46 typedef void(∗ tFNGnssSvInfo)(gnssSvInfoNotification ∗pGnssSvInfoNotification)

GNSS SVN Information Notification callback.

Parameters

pGnssSvInfo-Notification	• See gnssSvInfoNotification for more information.

9.12.3.47 typedef void(∗ tFNHDRPersonaity)(HDRPersonalityInd ∗pHDRPers)

HDR Personality indication callback.

Parameters

pHDRPers	• See HDRPersonalityInd for more information.

Note

Technology Supported: CDMA

9.12.3.48 typedef void(∗ tFNImsaPdpStatus)(imsaPdpStatusInfo ∗pImsaPdpStatusInfo)

IMSA PDP status indication callback.

Parameters

pImsaPdpStatusInfo	• See imsaPdpStatusInfo for more information.

9.12.3.49 typedef void(∗ tFNImsaRatStatus)(imsaRatStatusInfo ∗pImsaRatStatusInfo)

IMSA RAT handover status indication callback.

Parameters

pImsaRatStatusInfo	• See imsaRatStatusInfo for more information.

9.12.3.50 typedef void(∗ tFNImsaRegStatus)(imsaRegStatusInfo ∗pImsaRegStatusInfo)

IMSA Registration Status indication callback.

Parameters

pImsaRegStatusInfo	• See imsaRegStatusInfo for more information.

9.12.3.51 typedef void(∗ tFNImsaSvcStatus)(imsaSvcStatusInfo ∗pImsaSvcStatusInfo)

IMSA Service Status indication callback.

Parameters

pImsaSvcStatusInfo	• See imsaSvcStatusInfo for more information.

9.12.3.52 typedef void(∗ tFNImsRegMgrConfig)(imsRegMgrConfigInfo ∗pImsRegMgrConfigInfo)

IMS Reg Mgr Config indication callback.

Parameters

pImsRegMgrConfigInfo	• See imsRegMgrConfigInfo for more information.

9.12.3.53 typedef void(∗ tFNImsSIPConfig)(imsSIPConfigInfo ∗pImsSIPConfigInfo)

IMS SIP Config indication callback.

Parameters

pImsSIPConfig-Info	• See imsSIPConfigInfo for more information.

9.12.3.54 typedef void(∗ tFNImsSMSConfig)(imsSMSConfigInfo ∗pImsSMSConfigInfo)

IMS SMS Config indication callback.

Parameters

pImsSMSConfig-Info	• See imsSMSConfigInfo for more information.

9.12.3.55 typedef void(∗ tFNImsUserConfig)(imsUserConfigInfo ∗pImsUserConfigInfo)

IMS User Config indication callback.

Parameters

pImsUserConfig-Info	• See imsUserConfigInfo for more information.

9.12.3.56 typedef void(∗ tFNImsVoIPConfig)(imsVoIPConfigInfo ∗pImsVoIPConfigInfo)

IMS VoIP Config indication callback.

Parameters

pImsVoIPConfig-Info	• See imsVoIPConfigInfo for more information.

9.12.3.57 typedef void(∗ tFNInfoRec)(voiceInfoRec ∗pVoiceInfoRec)

Voice Information Record callback.

Parameters

pVoiceInfoRec	• See voiceInfoRec for more information.

Note

Technology Supported: CDMA
Device Supported: MC7750

9.12.3.58 typedef void(∗ tFNInjectPosition)(QmiCbkLocInjectPositionInd ∗pInjectPositionNotification)

Inject Position Notification callback.

Parameters

pInjectPosition-Notification	• See QmiCbkLocInjectPositionInd for more information.

9.12.3.59 typedef void(∗ tFNInjectSensorData)(QmiCbkLocInjectSensorDataInd ∗pLocInjectSensorData)

LOC Inject Sensor Data indication callback.

Parameters

pLocInject-SensorData	• See QmiCbkLocInjectSensorDataInd for more information.

9.12.3.60 typedef void(∗ tFNInjectTimeStatus)(QmiCbkLocInjectTimeInd ∗pLocInjectTime)

LOC Inject time status indication callback.

Parameters

pLocInjectTime	• See QmiCbkLocInjectTimeInd for more information.

9.12.3.61 typedef void(∗ tFNInjectUTCTime)(QmiCbkLocInjectUTCTimeInd ∗pInjectUTCTimeNotification)

Inject UTC Time Notification callback.

Parameters

pInjectUTCTime-Notification	• See QmiCbkLocInjectUTCTimeInd for more information.

9.12.3.62 typedef void(∗ tFNLUReject)(ULONG serviceDomain, ULONG rejectCause)

LU reject callback function.

Parameters

serviceDomain	• Service domain – 1 - Circuit Switched – 2 - Packet Switched – 3 - Circuit and Packet Switched

rejectCause	• Reject cause • Valid Values – 2 - IMSI unknown in HLR – 3 - Illegal MS – 4 - IMSI unknown in VLR – 5 - IMEI not accepted – 6 - Illegal ME – 7 - EPS services not allowed – 8 - EPS services and non-EPS services not allowed – 9 - UE identity cannot be derived by the network – 10 - Implicitly detached – 11 - PLMN not allowed – 12 - Location Area not allowed – 13 - Roaming not allowed in this location area – 14 - EPS services not allowed in this PLMN – 15 - No Suitable Cells In Location Area – 16 - MSC temporarily not reachable – 17 - Network failure – 18 - CS domain not available – 19 - ESM failure – 20 - MAC failure – 21 - Synch failure – 22 - Congestion – 23 - GSM authentication unacceptable – 24 - Security mode rejected, unspecified – 25 - Not authorized for this CSG – 26 - Non-EPS authentication unacceptable – 32 - Service option not supported – 33 - Requested service option not subscribed – 34 - Service option temporarily out of order – 38 - Call cannot be identified – 48 to 63 - retry upon entry into a new cell – 95 - Semantically incorrect message – 96 - Invalid mandatory information – 97 - Message type non-existent or not implemented – 98 - Message type not compatible with the protocol state – 99 - Information element non-existent or not implemented – 100 - Conditional IE error – 101 - Message not compatible with the protocol state – 111 - Protocol error, unspecified – Note - Any other value received by the mobile station shall be treated as 34, 'Service option temporarily out of order'. * Any other value received by the network shall be treated as 111, 'Protocol error, unspecified'. 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

9.12.3.63 typedef void(∗ tFNMemoryFull)(SMSMemoryInfo ∗pSMSMemoryFullInfo)

SMS Memory related callback function.

Parameters

pSMSMemory-FullInfo[OUT]	• pointer to SMSMemoryInfo. • see SMSMemoryInfo for details.

9.12.3.64 typedef void(∗ tFNMessageWaiting)(msgWaitingInfo ∗pSMSMessageWaitingInfo)

SMS Memory related callback function.

Parameters

pSMSMessage-WaitingInfo[OUT]	• pointer to msgWaitingInfo. • see msgWaitingInfo for details.

9.12.3.65 typedef void(∗ tFNMitiLvlRpt)(QmiCbkTmdMitiLvlRptInd ∗pSetLocCradleMount)

Set Loc cradle mount Indication callback.

Parameters

pSetLocCradle-Mount	• See QmiCbkTmdMitiLvlRptInd for more information.

9.12.3.66 typedef void(∗ tFNMobileIPStatus)(ULONG mipStatus)

Mobile IP status callback function.

Parameters

mipStatus	• Mobile IP Status – 0 - success – All others error codes as defined in RFC 2002 See qaGobiApiTableCallEndReasons.h for mobile IP error codes

9.12.3.67 typedef void(∗ tFNModemTempInfo)(modemTempNotification ∗pModemTempNotification)

Modem Temperature Information callback.

Parameters

pModemTemp-Notification	• See modemTempNotification for more information.

9.12.3.68 typedef void(∗ tFNNasTimer)(nasTimers ∗timers)

Network timer Indication callback.

Parameters

timers	• See nasTimers for more information.

9.12.3.69 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 • 1: throttle
instanceId	• qmi instance id

Note

Does not require communication with the device

9.12.3.70 typedef void(∗ tFNNetworkReject)(nasNetworkReject ∗pNetworkReject)

Network Reject Cause callback.

Parameters

pNasNetwork-Time	• See nasNetworkTime for more information.

9.12.3.71 typedef void(∗ tFNNetworkTime)(nasNetworkTime ∗pNasNetworkTime)

Network Time indication callback.

Parameters

pNasNetwork-Time	• See nasNetworkTime for more information.

9.12.3.72 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 – 0 - Success – 1 - In progress – 2 - General failure – 3 - Timeout – 4 - User ended the session – 5 - Bad parameter – 6 - Phone is offline – 7 - Engine is locked – 8 - E911 session in progress
pos_src[IN]	• position source • Bitmasks – 0x01 - GPS – 0x02 - Cell ID – 0x04 - GLONASS – 0x08 - Network – 0x10 - External positino injection – Others - unknown

9.12.3.73 typedef void(∗ tFNNewNMEA)(LPCSTR pNMEA)

New NMEA sentence callback function.

Parameters

pNMEA	• NULL-terminated string containing the position data in NMEA sentence format

9.12.3.74 typedef void(∗ tFNNewRMTransferStatistics)(QmiCbkWdsStatisticsIndState ∗pMsg)

PDS session state callback function.

Parameters

enabledStatus	• GPS enabled status – 0 - Disable – 1 - Enable
trackingStatus	• GPS tracking status – 0 - Unknown – 1 - Inactive – 2 - Active RM Transfer Statistics message callback function.
pMsg[OUT]	• Events related to NAS, see QmiCbkWdsStatisticsIndState for details

9.12.3.75 typedef void(∗ tFNNewSMS)(ULONG storageType, ULONG messageIndex)

New SMS message callback function.

Parameters

storageType	• SMS message storage type for the new message 0 - UIM 1 - NV
messageIndex	• Index of the new message

9.12.3.76 typedef void(∗ tFNOMADMState)(ULONG sessionState, ULONG failureReason)

OMA-DM state callback function

Parameters

sessionState	• Session state – 0x00 - Complete, information was updated – 0x01 - Complete, update information is unavailable – 0x02 - Failed – 0x03 - Retrying – 0x04 - Connecting – 0x05 - Connected – 0x06 - Authenticated – 0x07 - Mobile Directory Number (MDN) downloaded – 0x08 - Mobile Station Identifier (MSID) downloaded – 0x09 - PRL downloaded – 0x0A - Mobile IP (MIP) profile downloaded
failureReason	• Session failure reason (when state indicates failure) – 0x00 - Unknown – 0x01 - Network is unavailable – 0x02 - Server is unavailable – 0x03 - Authentication failed – 0x04 - Maximum retry exceeded – 0x05 - Session is cancelled

Note

Technology Supported: CDMA

9.12.3.77 typedef void(∗ tFNOpMode)(ULONG mode)

Operation mode callback function.

Parameters

mode[OUT]	• Status of the Set Operation Mode request. • Valid values – eQMI_LOC_SUCCESS - 0 - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE - 1 - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED - 2 - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER -3 - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY - 4 - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE - 5 -Request failed because the phone is offline – eQMI_LOC_TIMEOUT - 6 - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED -7 - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY - 8 - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED - 9 - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE - 10 - Location service failed because of an XTRA version-based file format check failure

9.12.3.78 typedef void(∗ tFNOTASPStatus)(voiceOTASPStatusInfo ∗pVoiceOTASPStatusInfo)

OTASP or OTAPA event Indication Callback function

Parameters

pVoiceOTASP-StatusInfo	• OTASP Status Information. • See voiceOTASPStatusInfo for more information

Note

Technology Supported: CDMA

9.12.3.79 typedef void(∗ tFNPacketSrvState)(packetSrvStatus ∗pPacketSrvStatus)

Packet Service state callback function.

Parameters

pPacketSrv-Status	• See packetSrvStatus for more details

9.12.3.80 typedef void(∗ tFNPDSState)(ULONG enabledStatus, ULONG trackingStatus)

PDS session state callback function.

Parameters

enabledStatus	• GPS enabled status – 0 - Disable – 1 - Enable
trackingStatus	• GPS tracking status – 0 - Unknown – 1 - Inactive – 2 - Active

9.12.3.81 typedef void(∗ tFNPower)(ULONG operatingMode)

Power operating mode callback function.

Parameters

operatingMode	• Service Operating mode See Tables for Operating Modes

Note

Technology Supported: UMTS/CDMA
Device Supported: MC83x5, MC7700/50

9.12.3.82 typedef void(∗ tFNPrivacyChange)(voicePrivacyInfo ∗pVoicePrivacyInfo)

Preferred voice privacy indication callback.

Parameters

pVoicePrivacy-Info	• See voicePrivacyInfo for more information.

Note

Technology Supported: CDMA

9.12.3.83 typedef void(∗ tFNQosNWStatus)(BYTE status)

QOS Network status callback function.

Parameters

status	Network QoS support status • 0x00 – Current network does not support QoS • 0x01 – Current network supports QoS

9.12.3.84 typedef void(∗ tFNQosPriEvent)(WORD event)

QOS primary flow callback function.

Parameters

event	Event which causes this indication: • 0x0001 – Primary flow QoS modify operation success • 0x0002 – Primary flow QoS modify operation failure

9.12.3.85 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: • 0x01 – QMI_QOS_STATUS_ACTIVATED • 0x02 – QMI_QOS_STATUS_SUSPENDED • 0x03 – QMI_QOS_STATUS_GONE
event	• 0x01 – QMI_QOS_ACTIVATED_EV • 0x02 – QMI_QOS_SUSPENDED_EV • 0x03 – QMI_QOS_GONE_EV • 0x04 – QMI_QOS_MODIFY_ACCEPTED_EV • 0x05 – QMI_QOS_MODIFY_REJECTED_EV • 0x06 – QMI_QOS_INFO_CODE_UPDATED_EV

reason	• 0x01 - QMI_QOS_INVALID_PARAMS • 0x02 - QMI_QOS_INTERNAL_CALL_ENDED • 0x03 - QMI_QOS_INTERNAL_ERROR • 0x04 - QMI_QOS_INSUFFICIENT_LOCAL_Resources • 0x05 - QMI_QOS_TIMED_OUT_OPERATION • 0x06 - QMI_QOS_INTERNAL_UNKNOWN_CAUSE_CODE • 0x07 - QMI_QOS_INTERNAL_MODIFY_IN_PROGRESS • 0x08 - QMI_QOS_NOT_SUPPORTED • 0x09 - QMI_QOS_NOT_AVAILABLE • 0x0A - QMI_QOS_NOT_GUARANTEED • 0x0B - QMI_QOS_INSUFFICIENT_NETWORK_RESOURCES • 0x0C - QMI_QOS_AWARE_SYSTEM • 0x0D - QMI_QOS_UNAWARE_SYSTEM • 0x0E - QOS_REJECTED_OPERATION • 0x0F - QMI_QOS_WILL_GRANT_WHEN_QOS_RESUMED • 0x10 - QMI_QOS_NETWORK_CALL_ENDED • 0x11 - QMI_QOS_NETWORK_SERVICE_NOT_AVAILABLE • 0x12 - QMI_QOS_NETWORK_L2_LINK_RELEASED • 0x13 - QMI_QOS_NETWORK_L2_LINK_REESTAB_REJ • 0x14 - QMI_QOS_NETWORK_L2_LINK_REESTAB_IND • 0x15 - QMI_QOS_NETWORK_UNKNOWN_CAUSE_CODE • 0x16 - QMI_NETWORK_BUSY

9.12.3.86 typedef void(∗ tFNRankIndicator)(RankIndicatorInd ∗pRankIndicatorInd)

Rank Indicator callback function.

Parameters

pRankIndicator-Ind[OUT]	• Events related to NAS, see RankIndicatorInd for details.

9.12.3.87 typedef void(∗ tFNResetInfo)(ResetInfoNotification ∗pResetInfoNotification)

Get Reset Info Indication callback.

Parameters

pResetInfo-Notification	• See ResetInfoNotification for more information.

9.12.3.88 typedef void(∗ tFNRFInfo)(ULONG radioInterface, ULONG activeBandClass, ULONG activeChannel)

RF information callback function.

Parameters

radioInterface	• Radio interface technology of the signal being measured See Tables for Radio Interface
activeBandClass	• Active band class See Tables for Active Band Class
activeChannel	• Active channel – 0 - Channel is not relevant to the reported technology

9.12.3.89 typedef void(∗ tFNRoamingIndicator)(ULONG roaming)

Roaming indicator callback function.

Parameters

roaming	• Roaming Indication – 0 - Roaming – 1 - Home – 2 - Roaming partner – >2 - Operator defined values

9.12.3.90 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

9.12.3.91 typedef void(∗ tFNSensorStreaming)(QmiCbkLocSensorStreamingInd ∗pLocSensorStream)

LOC Sensor Streaming indication callback.

Parameters

pLocSensor-Stream	• See QmiCbkLocSensorStreamingInd for more information.

9.12.3.92 typedef void(∗ tFNServingSystem)(struct ServingSystemInfo ∗pServingSystem,struct RoamingInfo ∗pRoamingInfo)

Serving System callback function

Parameters

pServingSystem	• ServingSystemInfo structure

9.12.3.93 typedef void(∗ tFNSetCradleMount)(QmiCbkLocCradleMountInd ∗pSetLocCradleMount)

Set Cradle Mount indication callback.

Parameters

pSetLocCradle-Mount	• See QmiCbkLocCradleMountInd for more information.

9.12.3.94 typedef void(∗ tFNSetEngineState)(QmiCbkLocEngineStateInd ∗pSetLocEngineState)

Set Engine State Indication callback.

Parameters

pSetLocEngine-State	• See QmiCbkLocEngineStateInd for more information.

9.12.3.95 typedef void(∗ tFNSetEventTimeSync)(QmiCbkLocEventTimeSyncInd ∗pSetLocEventTimeSync)

LOC event time sync indication callback.

Parameters

pSetLocEvent-TimeSync	• See QmiCbkLocEventTimeSyncInd for more information.

9.12.3.96 typedef void(∗ tFNSetExtPowerConfig)(QmiCbkLocSetExtPowerConfigInd ∗pSetExtConfigIndStatus)

Set Ext Power Config Indication callback.

Parameters

pSetExtConfig-IndStatus	• See QmiCbkLocSetExtPowerConfigInd for more information.

9.12.3.97 typedef void(∗ tFNSigInfo)(nasSigInfo ∗pNasSigInfo)

Signal Strength Information indication callback.

Parameters

pNasSigInfo	• See nasSigInfo for more information.

9.12.3.98 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 See Tables for Radio Interface

9.12.3.99 typedef void(∗ tFNSLQSOMADMAlert)(ULONG eventType, BYTE ∗pEventFields)

SWIOMA-DM network-initiated alert callback function

Parameters

eventType	• 0x00 - SWIOMA-DM FOTA • 0x01 - SWIOMA-DM Config • 0x02 - SWIOMA-DM Notification
pEventFields	• Pointer to structure containing info for that session type • See sessionInfo for more details

9.12.3.100 typedef void(∗ tFNSLQSQOSEvent)(BYTE instance, QosFlowInfo ∗pFlowInfo)

QOS Event callback function.

Parameters

instance	• QMI instance
pFlowInfo	• See QosFlowInfo for more information

9.12.3.101 typedef void(* tFNSLQSSessionState)(slqsSessionStateInfo *pSessionStateInfo)

Session state callback function.

Parameters

pSessionStateInfo	• See slqsSessionStateInfo for more details

9.12.3.102 typedef void(* tFNSLQSSignalStrengths)(struct SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo)

Received Signal Strength Information callback function.

Parameters

sSLQSSignalStrengthsInfo	• See SLQSSignalStrengthsInformation for more information.

9.12.3.103 typedef void(* tFNSLQSWDSEvent)(slqsWdsEventInfo *pWdsEventInfo)

WDS Event callback function.

Parameters

pWdsEventInfo	• See slqsWdsEventInfo for more details

9.12.3.104 typedef void(* tFNSMSEvents)(SMSEventInfo *pSMSEventInfo)

SMS event related callback function.

Parameters

pSMSEventInfo[-OUT]	• Events related to SMS, see SMSEventInfo for details

9.12.3.105 typedef void(* tFNSUPSInfo)(voiceSUPSInfo *pVoiceSUPSInfo)

Preferred SUPS indication callback.

Parameters

pVoiceSUPSInfo	• See voiceSUPSInfo for more information.

Note

Technology Supported: GSM

9.12.3.106 typedef void(* tFNSUPSNotification)(voiceSUPSNotification *pVoiceSUPSNotification)

Supplementary service notification callback.

Parameters

pVoiceSUPS-Notification	• See voiceSUPSNotification for more information.

9.12.3.107 typedef void(* tFNSwiAvmEventReport)(SwiAvmsEventInd *pEvent)

AVMS Event Report indication callback.

Parameters

pEvent	• See SwiAvmsEventInd for more information.

9.12.3.108 typedef void(* tFNSysInfo)(nasSysInfo *pNasSysInfo)

System Information indication callback.

Parameters

pNasSysInfo	• See nasSysInfo for more information.

9.12.3.109 typedef void(* tFNSysSelectionPref)(sysSelectPrefInfo *pSysSelectPrefInfo)

System Selection Preference Callback function

Parameters

pSysSelectPref-Info	• Current System Selection preferences for the device. • See sysSelectPrefInfo for more information

9.12.3.110 typedef void(* tFNtransLayerInfo)(transLayerNotification *pTransLayerNotification)

Transport Layer Information callback.

Parameters

transLayer-Notification	• See transLayerNotification for more information.

9.12.3.111 typedef void(∗ tFNtransNWRegInfo)(transNWRegInfoNotification ∗pTransNWRegInfoNotification)

Transport Network Registration Information callback.

Parameters

pTransNWRegInfoNotification	• See transNWRegInfoNotification for more information.

9.12.3.112 typedef void(∗ tFNUIMRefresh)(UIMRefreshEvent ∗pUIMRefreshEvent)

UIM Refresh Callback function

Parameters

pUIMRefreshEvent	• Pointer to Refresh Event structure. • See UIMRefreshEvent for more information

9.12.3.113 typedef void(∗ tFNUIMStatusChangeInfo)(UIMStatusChangeInfo ∗pUIMStatusChangeInfo)

UIM Status Change Callback function

Parameters

pUIMStatusChangeInfo	• Pointer to UIM status change structure. • See UIMStatusChangeInfo for more information

9.12.3.114 typedef void(∗ tFNUSSDNotification)(ULONG type, BYTE ∗pNetworkInfo)

SetUSSDNotificationCallback function prototype

Parameters

type	- Notification type • 0x01 - No action required • 0x02 - Action required
pNetworkInfo	• USS information from the network (0 indicates that no info was received) – See USSInfo for more details

Note

Technology Supported: UMTS

9.12.3.115 typedef void(* tFNUSSDNoWaitIndication)(USSDNoWaitIndicationInfo *pNetworkInfo)

USSD No Wait indication callback.

Parameters

pNetworkInfo	• See USSDNoWaitIndicationInfo for more information.

9.12.3.116 typedef void(* tFNUSSDRelease)(void)

USSD releasecallback function prototype

Note

Technology Supported: UMTS

9.12.3.117 typedef struct _transLayerInfoNotification transLayerNotification

Contains the parameters passed for SLQSSetTransLayerInfoCallback by the device.

Parameters

regInd	• Indicates whether the transport layer is registered or not • Values: – 0x00 - Transport layer is not registered – 0x01 - Transport layer is registered
pTransLayerInfo	• Optional parameter • See transLayerInfo for more information

Note

None

9.12.3.118 typedef struct _transNWRegInfoNotification transNWRegInfoNotification

Contains the parameters passed for SLQSSetTransNWRegInfoCallback by the device.

Parameters

NWRegStat	• provides the transport network registration information • Values: – 0x00 - No Service – 0x01 - In Progress – 0x02 - Failed – 0x03 - Limited Service – 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 – 0x00000001 - Display Text – 0x00000002 - Get In-Key – 0x00000004 - Get Input – 0x00000008 - Setup Menu – 0x00000010 - Select Item – 0x00000020 - Send SMS - Alpha Identifier – 0x00000040 - Setup Event: User Activity – 0x00000080 - Setup Event: Idle Screen Notify – 0x00000100 - Setup Event: Language Sel Notify – 0x00000200 - Setup Idle Mode Text – 0x00000400 - Language Notification – 0x00000800 - Refresh – 0x00001000 - End Proactive Session
pErrorMask[OUT]	• error bitmask. Each bit set indicates the proactive command that caused the error – 0x00000001 - Display Text – 0x00000002 - Get In-Key – 0x00000004 - Get Input – 0x00000008 - Setup Menu – 0x00000010 - Select Item – 0x00000020 - Send SMS - Alpha Identifier – 0x00000040 - Setup Event: User Activity – 0x00000080 - Setup Event: Idle Screen Notify – 0x00000100 - Setup Event: Language Sel Notify – 0x00000200 - Setup Idle Mode Text – 0x00000400 - Language Notification – 0x00000800 - Refresh – 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 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

9.12.5.13 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 • 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

9.12.5.14 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 • 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

9.12.5.15 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

9.12.5.16 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

9.12.5.17 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

9.12.5.18 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)

9.12.5.19 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

9.12.5.20 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

9.12.5.21 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

9.12.5.22 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

9.12.5.23 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

9.12.5.24 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

9.12.5.25 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)

9.12.5.26 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

9.12.5.27 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

9.12.5.28 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

9.12.5.29 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

9.12.5.30 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

9.12.5.31 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 • 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

9.12.5.32 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

9.12.5.33 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

9.12.5.34 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 • NULL to disable OMADMState notification

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

Technology Supported: CDMA
Timeout: 2 seconds

9.12.5.35 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

9.12.5.36 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

9.12.5.37 ULONG SetRankIndicatorCallback (tFNRankIndicator *pCallback*)

9.12.5.38 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

9.12.5.39 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

9.12.5.40 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

9.12.5.41 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; • This parameter is not used when disabling the callback.

pThresholds[IN]	• Signal threshold array for each entry (in dBm). • 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.

9.12.5.42 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 • NULL to disable SLQSOMADMAlert notification

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

9.12.5.43 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 • 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

9.12.5.44 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

9.12.5.45 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 • 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

9.12.5.46 ULONG SetUSSDNoWaitIndicationCallback (tFNUSSDNoWaitIndication *pCallback*)

SetUSSDNoWaitIndicationCallback

Parameters

pNetworkInfo	• Data from the network. • See USSDNoWaitIndicationInfo for more details.

Note

Technology Supported: UMTS
Device Supported: MC83x5

9.12.5.47 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 • 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

9.12.5.48 ULONG SLQSNasNetworkRejectCallback (tFNNetworkReject *pCallback*)

9.12.5.49 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.

9.12.5.50 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 • 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

9.12.5.51 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 • 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

9.12.5.52 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

9.12.5.53 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.

9.12.5.54 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

9.12.5.55 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 • 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

9.12.5.56 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

9.12.5.57 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

9.12.5.58 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) • 0 - Do not report • 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

9.12.5.59 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

9.12.5.60 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

9.12.5.61 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

9.12.5.62 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

9.12.5.63 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

9.12.5.64 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

9.12.5.65 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

9.12.5.66 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)

9.12.5.67 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)

9.12.5.68 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

9.12.5.69 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

9.12.5.70 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)

9.12.5.71 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

9.12.5.72 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

9.12.5.73 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

9.12.5.74 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

9.12.5.75 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 • 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
```

9.12.5.76 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 • NULL to disable ServingSystem notification

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

9.12.5.77 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

9.12.5.78 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 • 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.

9.12.5.79 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

9.12.5.80 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

9.12.5.81 ULONG SLQSSetSwiGetResetInfoCallback (tFNResetInfo *pCallback*)

Reset Info callback.

Parameters

pCallback	• See tFNResetInfo for more information.

9.12.5.82 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

9.12.5.83 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 • 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

9.12.5.84 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

9.12.5.85 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

9.12.5.86 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. • ignored when disabling, should be non-zero when enabling • 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 • 6 for an IPv6 data session • 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.

9.12.5.87 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. • ignored when disabling, should be non-zero when enabling • 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 • 6 for an IPv6 data session • 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.

9.12.5.88 ULONG SLQSSwiAVMSEventReportCallback (tFNSwiAvmEventReport *pCallback*)

9.12.5.89 ULONG SLQSTmdMitigationLvlRptCallback (TmdMitigationLvlIndReq * *req,* tFNMitiLvlRpt *pCallback*)

Thermal Mitigation callback.

Parameters

req	• See TmdMitigationLvlIndReq for more information.
pCallback	• See tFNMitiLvlRpt for more information.

9.12.5.90 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 • 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.

9.12.5.91 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 • NULL to disable Band Preference notification

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

Timeout: 2 seconds

9.12.5.92 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

9.12.5.93 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

9.12.5.94 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.

9.12.5.95 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 • 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.

9.12.5.96 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.

9.12.5.97 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.

9.12.5.98 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

9.12.5.99 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

9.12.5.100 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

9.12.5.101 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)

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] • Name of AT command port
NmeaPort	[OUT] • Name of NMEA port
DmPort	[OUT] • 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 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

9.13.3.2 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. • Upon successful output, actual number of elements in the device array.
pDevices[IN/OUT]	• Device array; array elements are structures with the following elements: CHAR deviceID[256] - Device path (e.g. /dev/qcqmi0) 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

9.13.3.3 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

9.13.3.4 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

9.13.3.5 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

9.13.3.6 ULONG QCWWANEnumerateDevices (BYTE * *pDevicesSize,* BYTE * *pDevices*)

Enumerates the QC WWAN devices currently attached to the host. This API is deprecated; use QCWWAN2k-EnumerateDevices() instead.

Parameters

pDeviceSize[IN/-OUT]	• Upon input, maximum number of elements that the device array can contain. • Upon successful output, actual number of elements in the device array.
pDevices[IN/OU-T]	• Device array; array elements are structures with the following elements: CHAR deviceID[256] - Device path (e.g. /dev/qcqmi0) 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.

9.13.3.7 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

9.13.3.8 ULONG SLQSGetDeviceMode (BYTE * *pDeviceMode*)

Returns the Device Mode

Parameters

pDeviceMode[O-UT]	• 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

9.13.3.9 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

9.13.3.10 ULONG SLQSGetUsbPortNames (struct DcsUsbPortNames * *pUsbPortNames*)

Returns the Usb Port Names currently in use.

Parameters

pUsbPortNames[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

9.13.3.11 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.

9.13.3.12 ULONG SLQSSetLoggingMask (BYTE *mask*)

Limit Syslog messages according to the Mask provided by user

Parameters

mask	• Mask 0x01: disable all log • 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

9.13.3.13 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 • 1: second modem detected • 2: third modem detected • ... • 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

9.13.3.14 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

9.13.3.15 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 – Bit0 - WDS – Bit1 - NAS – Bit2 - PDS – 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

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)

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 • values: – 0x00 - GPS is disabled – 0x01 - GPS is enabled • function SLQSGetCustFeatures() returns a default value FFFFFFFF if no value is returned by the modem
pDisableIMSI[OUT]	• optional 1 byte parameter • describes if IMSI display is enabled or disabled • values: – 0x00 - Allow display of IMSI – 0x01 - Do not display IMSI • function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem
pIPFamSupport[OUT]	• optional 2 byte BitMask • bitmask representing the IP families supported • values: – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - IPv4v6 • function SLQSGetCustFeatures() returns a default value FFFF if no value is returned by the modem
pRMAutoConnect[OUT]	• optional 1 byte parameter • QMI Mode RM Net Auto Connect Support • values: – 0x00 - Not Supported – 0x01 - Supported • function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem
pGPSSel[OUT]	• optional 1 byte parameter • GPS Antenna Select • values: – 0x00 - Dedicated GPS Port – 0x01 - GPS Rx over AUX Port – 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied • function SLQSGetCustFeatures() returns a default value FF if no value is returned by the modem

pSMSSupport[-OUT]	• optional 1 byte parameter • SMS support • values: – 0x00 - Not supported – 0x01 - supported • Used to determine whether or not to hide SMS from user • 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 • Voice support • values: – 0x00 - Enable voice on both AT and QMI interface (default) – 0x01 - Reserved – 0x02 - Disable voice on both AT and QMI interface
pDHCPRelay-Enabled[OUT]	• optional 1 byte parameter • DHCP Relay support • values: – 0x00 - Disable DHCP relay – 0x01 - Enable DHCP relay
pGPSLPM[OUT]	• optional 1 byte parameter • GPSLPM support • values: – 0x00 - Enable GPS in Low Power Mode – 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 • GPS Antenna Select • values: – 0x00 - Dedicated GPS Port – 0x01 - GPS Rx over AUX Port – 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied
pGPSEnable	• optional 4 byte parameter • GPS Enable/Disable • values: The value of 7 least significant bits: – 0 - Disabled – 1 - MT & MO enabled – 2 - MO enabled – 3 - MT enabled – 4 - MT & MO enabled if GPS_DISABLE pin is not asserted – 5 - MO GPS enabled if GPS_DISABLE pin is not asserted – 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 • voice enabled/disabled • values: – 0 - Enable voice on both AT and QMI interface (default) – 1 - Reserved – 2 - Disable voice on both AT and QMI interface
pDHCPRelay-Enabled	• optional 1 byte parameter • DHCPRELAYENABLE support • values: – 0 - Disable DHCP relay – 1 - Enable DHCP relay
pGPSLPM	• optional 1 byte parameter • GPSLPM support • values: – 0 - Enable GPS in Low Power Mode – 1 - Disable GPS in Low Power Mode

9.14.3.3 typedef struct dmsCurrentPRLInfo dmsCurrentPRLInfo

This structure contains GetCurrentPRLInfo response parameter

Parameters

pPRLVersion[OUT]	- Optional • PRL version of device.
pPRLPreference	[OUT]- Optional • PRL Preference – 0 - Unset – 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. • 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) • Null terminated ASCII String
bModelSize[IN/OUT]	• Host Device Model String Size
pModelString[OUT]	• Host Device Model String(Optional parameter) • Null terminated ASCII string.
bSWVerSize[IN/OUT]	• Host Device Software Version String Size
pSWVerString[OUT]	• Host Device Software Version String(Optional parameter) • Null terminated ASCII string
bPlasmaIDSize[IN/OUT]	• Host Device Plasma ID String Size
pPlasmaIDString[OUT]	• Host Device Plasma ID String(Optional parameter) • 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[O-UT]	• Operating System Name(Optional parameter) • Null terminated ASCII string
bVersionSize[I-N/OUT]	• Operating System Version Size
pVersionString[-OUT]	• Operating System Version String(Optional parameter) • Null terminated ASCII string.

9.14.3.8 typedef struct _SLQSSwiGetSerialNoExtParams SLQSSwiGetSerialNoExtParams

This structure is used to store MEID Information

Parameters

meidLength[OU-T]	• String length of the of MEID received
pMeidString[OU-T]	• Optional parameter • 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) • Null terminated ASCII String
bModelSize[IN]	• Host Device Model String Size
pModelString[IN]	• Host Device Model String(Optional parameter) • Null terminated ASCII string.
bSWVerSize[IN]	• Host Device Software Version String Size

pSWVerString[IN]	• Host Device Software Version String(Optional parameter) • Null terminated ASCII string
bPlasmaIDSize[IN]	• Host Device Plasma ID String Size
pPlasmaIDString[IN]	• Host Device Plasma ID String(Optional parameter) • 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) • Null terminated ASCII string
bVersionSize[IN]	• Operating System Version Size
pVersionString[IN]	• Operating System Version String(Optional parameter) • 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

pActivationCode[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.

Parameters

pActivation-State[OUT]	• Service Activation Code 0 - Service not activated 1 - Service activated 2 - Activation connecting 3 - Activation connected 4 - OTASP security authenticated 5 - OTASP NAM downloaded 6 - OTASP MDN downloaded 7 - OTASP IMSI downloaded 8 - OTASP PRL downloaded 9 - OTASP SPC downloaded 10 - OTASP settings committed

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[O-UT]	• Maximum transmission rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies
pMaxRX-ChannelRate[O-UT]	• Maximum reception rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies

pDataService-Capability[OUT]	• CS/PS data service capability – 0 - No data services supported – 1 - Only Circuit Switched (CS) services supported – 2 - Only Packet Switched (PS) services supported – 3 - Simultaneous CS and PS – 4 - Non-simultaneous CS and PS
pSimCapability[-OUT]	• Device SIM capability – 0 - SIM not supported – 1 - SIM supported
pRadioIfaces-Size[IN/OUT]	• Upon input, the maximum number of elements that the radio interface array can contain • 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 – 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 modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx 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 0 - 32 kHz device clock 1 - CDMA network 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 • Bitmask of offline reasons – 0x00000001 - Host image configuration issue – 0x00000002 - PRI image configuration issue – 0x00000004 - PRI version incompatible – 0x00000008 - PRI copy issue – All others - Reserved
pbPlatform[OUT]	• Optional parameter • Is the device offline due to a platform restriction? – 0 - No – 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 – 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 • 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 – 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 SLQSDmsSwiGetResetInfo (dmsSwiGetResetInfo * *pGetResetInfoResp*)

This function is used to get reset info

Parameters

pGetResetInfo-Resp	• See dmsSwiGetResetInfo for more information of the input structure

9.14.4.19 ULONG SLQSDmsSwiIndicationRegister (dmsIndicationRegisterReq * *pIndicationRegisterReq*)

This function used to set Swi Indication Register

Parameters

pConfig	• See dmsIndicationRegisterReq for more information of the inpurt structure

9.14.4.20 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

9.14.4.21 ULONG SLQSGetBandCapability (ULONGLONG * *pBandCapability*)

Returns the band capability of the device.

Parameters

pBand-Capability[OUT]	Bitmask of bands supported by the device • Bit 0 - Band class 0, A-system • Bit 1 - Band class 0, B-system • Bit 2 - Band class 1, all blocks • Bit 3 - Band class 2 • Bit 4 - Band class 3, A-system • Bit 5 - Band class 4, all blocks • Bit 6 - Band class 5, all blocks • Bit 7 - GSM DCS band (1800) • Bit 8 - GSM Extended GSM (E-GSM) band (900) • Bit 9 - GSM Primary GSM (P-GSM) band (900) • Bit 10 - Band class 6 • Bit 11 - Band class 7 • Bit 12 - Band class 8 • Bit 13 - Band class 9 • Bit 14 - Band class 10 • Bit 15 - Band class 11 • Bit 16 - GSM 450 band • Bit 17 - GSM 480 band • Bit 18 - GSM 750 band • Bit 19 - GSM 850 band • Bit 20 - GSM railways GSM band (900) • Bit 21 - GSM PCS band (1900) • Bit 22 - WCDMA (Europe, Japan, and China) 2100 band • Bit 23 - WCDMA US PCS 1900 band • Bit 24 - WCDMA (Europe and China) DCS 1800 band • Bit 25 - WCDMA US 1700 band • Bit 26 - WCDMA US 850 band • Bit 27 - WCDMA Japan 800 band • Bit 28 - Band class 12 • Bit 29 - Band class 14 • Bit 30 - Reserved • Bit 31 - Band class 15 • Bits 32 through 47 - Reserved • Bit 48 - WCDMA Europe 2600 band • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved • Bit 56 - Band class 16 • Bit 57 - Band class 17 • Bit 58 - Band class 18 • Bit 59 - Band class 19

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 SLQSGetCurrentPRLInfo (dmsCurrentPRLInfo * *pCurrentPRLInfo*)

This API get the currently active PRL information of the device.

Parameters

pCurrentPRLInfo	• Pointer to structure dmsCurrentPRLInfo • 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

9.14.4.23 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. • 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

9.14.4.24 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

9.14.4.25 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 • 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

9.14.4.26 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 SLQSSwiGetSerialNoExt() to get MEID of CDMA modems. Timeout: 2 seconds

9.14.4.27 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. • 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

9.14.4.28 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
- GPSSEL
- IMSWITCHHIDE
- IPV6ENABLE
- WAKEHOSTEN

Parameters

pSetCustSetting	• Optional parameter • See setCustomSettingV2 for more information

9.14.4.29 ULONG SLQSSwiClearDyingGaspStatistics ( )

This function Clear Dying GASP Statistics.

9.14.4.30 ULONG SLQSSwiGetCrashAction (BYTE * *pDevCrashState*)

This API queries the Crash State from the device.

Parameters

pDevCrash-State[OUT]	• Device Crash State • Values: – 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. – 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 – 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

9.14.4.31 ULONG SLQSSwiGetCrashInfo (BYTE * *pClear*, CrashInfoParams * *pCrashInfoParams*)

This API queries the Crash Information from the device.

Parameters

pClear[IN]	• request parameter Clear(Optional parameter) • Values: 0 - Do not clear crash data after response 1 - Clear crash data after response

pCrashInfo-Params[Out]	• Pointer to structure CrashInfoParams • 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

9.14.4.32 ULONG SLQSSwiGetDyingGaspCfg (getDyingGaspCfg * *pConfig*)

This function queries Dying GASP Config.

Parameters

pGetCustom-FeatureV2	• See getDyingGaspCfg for more information of the inpurt structure *

9.14.4.33 ULONG SLQSSwiGetDyingGaspStatistics (getDyingGaspStatistics * *pStatistics*)

This function queries Dying GASP Statistics.

Parameters

pStatistics	• See getDyingGaspStatistics for more information of the inpurt structure *

9.14.4.34 ULONG SLQSSwiGetFSN (FactorySequenceNumber * *pFSNumber*)

This API get the Factory Sequence Number of the device.

Parameters

pFSNumber	• Pointer to structure FactorySequenceNumber • 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

9.14.4.35 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 • 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

9.14.4.36 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

9.14.4.37 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

9.14.4.38 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

9.14.4.39 ULONG SLQSSwiGetUSBComp (USBCompParams * *pUSBCompParams*)

This API queries the modem's USB interface configuration and supported configuration parameters.

Parameters

pUSBComp-Params	• Pointer to structure USBCompParams • 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

9.14.4.40 ULONG SLQSSwiSetCrashAction (BYTE *crashActionParams*)

This API set the Crash Action to the device.

Parameters

crashAction-Params[IN]	• Crash Action • Values: – 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. – 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 – 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

9.14.4.41 ULONG SLQSSwiSetDyingGaspCfg (setDyingGaspCfg * *pConfig*)

This function set Dying GASP Config.

Parameters

pConfig	• See setDyingGaspCfg for more information of the inpurt structure

9.14.4.42 ULONG SLQSSwiSetHostDevInfo (SLQSSwiSetHostDevInfoParams * *pSetHostDevInfoParams*)

This API Sets the host device info configured on the modem for OMA-DM reporting

Parameters

pSetHostDevInfoParams	• 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

9.14.4.43 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

9.14.4.44 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 • 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

9.14.4.45 ULONG SLQSUIMGetState (ULONG * *pUIMState*)

Returns the UIM state. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMGetCardStatus() for new firmware versions and new modules

Parameters

pUIMState[OUT]	• UIM state: – 0x00 - UIM initialization completed – 0x01 - UIM locked or failed – 0x02 - UIM not present – 0x03 - 0xFE - Reserved – 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

9.14.4.46 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 modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMChangePin() for new firmware versions and new modules

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 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 • Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
pUnblock-RetriesLeft[OUT]	• Optional Parameter • Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked; i.e. UIM is unusable. – 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

9.14.4.47 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 modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Pleaes use SLQSUIMGet-Configuration instead for new firmware versions and new modules.

Parameters

id[IN]	• Facility ID – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)
pStatus[OUT]	• Control key status – 0 - Deactivated – 1 - Activated – 2 - Blocked
pVerifyRetries-Left[OUT]	• The number of retries left, after which the control key will be blocked – 0xFFFFFFFF - Unknown
pUnblock-RetriesLeft[OUT]	• The number of unblock retries left, after which the control key will be permanently blocked – 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

9.14.4.48 ULONG UIMGetICCID (BYTE *stringSize*, CHAR * *pString*)

Returns the UIM ICCID. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15-C_05_xx_xx_xx and all EM74xx 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

9.14.4.49 ULONG UIMGetPINStatus (ULONG *id*, ULONG * *pStatus*, ULONG * *pVerifyRetriesLeft*, ULONG * *pUnblockRetriesLeft*)

Gets the status of the SIM PINs. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMGetCardStatus() for new firmware versions and new modules

Parameters

id	• PIN ID – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
pStatus[OUT]	• PIN status(0xFFFFFFFF - Unknown) – 0 - PIN not initialized – 1 - PIN enabled, not verified – 2 - PIN enabled, verified – 3 - PIN disabled – 4 - PIN blocked – 5 - PIN permanently blocked
pVerifyRetries-Left[OUT]	• Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
pUnblock-RetriesLeft[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. – 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 UIMSetControlKeyProtection (ULONG *id*, ULONG *status*, CHAR * *pValue*, ULONG * *pVerifyRetriesLeft*)

Changes the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMDepersonalization() for new firmware versions and new modules

Parameters

id[IN]	• Facility ID – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)
status[IN]	• Control key status – 0 - Deactivated
pValue[IN]	• Control key de-personalization string (maximum length of 8 characters)
pVerifyRetries-Left[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left, after which the control key will be blocked – 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 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 modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMSet-PinProtection() for new firmware versions and new modules

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 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 • Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
pUnblock-RetriesLeft[OUT]	• Optional parameter • Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. – 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

9.14.4.52 ULONG UIMUnblockControlKey (ULONG *id,* CHAR * *pValue,* ULONG * *pUnblockRetriesLeft*)

Unblocks the specified UIM facility control key. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMDepersonalization() for new firmware versions and new modules

Parameters

id[IN]	• Facility ID – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)
pValue[IN]	• Control key de-personalization string (maximum length of 8 characters)

pUnblock-RetriesLeft[OUT]	• Optional parameter • Upon operational failure, this will indicate number of unblock retries left, after which the control key will be blocked – 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 UIMUnblockPIN (ULONG *id*, CHAR * *pPUKValue*, CHAR * *pNewValue*, ULONG * *pVerifyRetriesLeft*, ULONG * *pUnblockRetriesLeft*)

Unblocks a blocked SIM. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_-05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMUnblockPin() for new firmware versions and new modules

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 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 • Upon operational failure, this will indicate number of retries left, after which the PIN will be blocked. – 0xFFFFFFFF - Unknown
pUnblock-RetriesLeft[OUT]	• Optional Parameter • Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked; i.e. UIM is unusable – 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 UIMVerifyPIN (ULONG *id*, CHAR * *pValue*, ULONG * *pVerifyRetriesLeft*, ULONG * *pUnblockRetriesLeft*)

Verifies a SIM PIN. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API SLQSUIMVerifyPin()() for new firmware versions and new modules

Parameters

id[IN]	• PIN ID – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
pValue[IN]	• Value of PIN to verify
pVerifyRetriesLeft[OUT]	• Optional Parameter • Upon operational failure, this will indicate number of retries left, after which the PIN will be blocked. – 0xFFFFFFFF - Unknown
pUnblockRetriesLeft[OUT]	• Optional Parameter • Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked; i.e. UIM is unusable – 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 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 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)

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. • 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 • 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 • 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 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 – 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 • 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

9.15.4.8 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

9.15.4.9 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

9.15.4.10 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.

9.15.4.11 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 • 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

9.15.4.12 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) • should use a "/" at the end of the path.
imgType[IN]	• 2 - Firmware Image(.cwe extension) • 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

9.15.4.13 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 • 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

9.15.4.14 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 • 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

9.15.4.15 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

9.15.4.16 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

9.15.4.17 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) • 1 - ignore crash state checking

Note

Device Supported: EM73xx/MC73xx, EM74xx/MC74xx

9.15.4.18 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

9.15.4.19 void SLQSSetSpkgFormatRequired (BYTE *isneeded*)

Set if SPKG format download is required for SL9090/MC9090

Parameters

isneeded	[INPUT] user inputs the firmware download method preference • 1 - Gobi3K download method, use mbn files. This is default value • 2 - SPKG download method, use cwe file

Returns

None

Note

Device Supported: MC9090/SL9090

9.15.4.20 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.

9.15.4.21 ULONG SLQSSwiGetFirmwareCurr (CurrentImgList * *pCurrentImgList*)

This API gets the currently active images on the device.

Parameters

pCurrentImgList	• Pointer to structure CurrentImgList • 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

9.15.4.22 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

9.15.4.23 ULONG upgrade_mc77xx_fw (LPCSTR *path*)

9.15.4.24 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 *pSetSIPConfig-Resp)
- ULONG SLQSSetRegMgrConfig (SetRegMgrConfigReq *pSetRegMgrConfigReq, SetRegMgrConfigResp *pSetRegMgrConfigResp)
- ULONG SLQSSetIMSSMSConfig (SetIMSSMSConfigReq *pSetIMSSMSConfigReq, SetIMSSMSConfig-Resp *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

pGetIMSSMS-ConfigParams[I-N/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

pGetIMSUser-ConfigParams[I-N/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

GetIMSVoIP-ConfigResp[OU-T]	• 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

pGetRegMgr-ConfigParams[I-N/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

pGetSIPConfig-Resp[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. – 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

pSetIMSUser-ConfigReq[IN]	• See SetIMSUserConfigReq for more information
pSetIMSUser-ConfigResp[OU-T]	• 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

pSetIMSVoIP-ConfigReq[IN]	• See SetIMSVoIPConfigReq for more information
pSetIMSVoIP-ConfigResp[OU-T]	• 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

pSetRegMgr-ConfigReq[IN]	• See SetRegMgrConfigReq for more information
pSetRegMgr-ConfigResp[OU-T]	• 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

pSetSIPConfig-Req[IN]	• See SetSIPConfigReq for more information
pSetSIPConfig-Resp[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

pIMSARegistrationStatus[OUT]	• Structure containing response parameters for registration status. – 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

pIMSAServiceStatus[OUT]	• Structure containing response parameters for service status. – 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[OUT]	• Structure containing Supported Fields Response. – 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. – 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. – 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

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)

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. • 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 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

9.18.3.5 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

9.18.3.6 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

9.18.3.7 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

9.18.3.8 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

9.18.3.9 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: – eQMI_LOC_OPER_MODE_DEFAULT (1) - Use the default engine mode – eQMI_LOC_OPER_MODE_MSB (2) - Use the MS-based mode – eQMI_LOC_OPER_MODE_MSA (3) - Use the MS-assisted mode – eQMI_LOC_OPER_MODE_STANDALONE (4) - Use Standalone mode – eQMI_LOC_OPER_MODE_CELL_ID (5) - Use cell ID; this mode is only valid for GSM/UMTS networks – 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

9.18.3.10 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

9.18.3.11 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

9.18.3.12 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

9.18.3.13 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 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 _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

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 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 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)

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 – 0x04 - GERAN – 0x05 - UMTS – 0x08 - LTE – 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. • Upon successful output, the actual number of elements in the network info instance array.
pNetworkInfo[O-UT]	• Network info instance array – See SlqsNas3GppNetworkInfo for more information
pRATInstances[-IN/OUT]	• Upon input, maximum number of elements that the RAT info instance array can contain. • Upon successful output, the actual number of elements in the RAT info instance array.

pRATInfo[OUT]	• RAT info instance array – See SlqsNas3GppNetworkRAT for more information
pPCSDigitInstances[IN/OUT]	• Upon input, maximum number of elements that the PCS Digit info instance array can contain. • Upon successful output, the actual number of elements in the PCS Digit info instance array.
pPCSDigitInfo[OUT]	• PCS Digit info instance array – See SlqsNasPcsDigit for more information
pScanResult[OUT]	• status of network scan • 0x00 - scan successful • 0x01 - scan was aborted • 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 • Values: – 0 - OFF (normal) – 1 - ON (Emergency) • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.

pBandPref	• Optional parameter • Bit mask representing the band preference • Bit values: – Bit 0 - Band Class 0, A-System – Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band – Bit 2 - Band Class 1, all blocks – Bit 3 - Band Class 2 place holder – Bit 4 - Band Class 3, A-System – Bit 5 - Band Class 4, all blocks – Bit 6 - Band Class 5, all blocks – Bit 7 - GSM_DCS_1800 band – Bit 8 - GSM Extended GSM (E-GSM) 900 band – Bit 9 - GSM Primary GSM (P-GSM) 900 band – Bit 10 - Band Class 6 – Bit 11 - Band Class 7 – Bit 12 - Band Class 8 – Bit 13 - Band Class 9 – Bit 14 - Band Class 10 – Bit 15 - Band Class 11 – Bit 16 - GSM 450 band – Bit 17 - GSM 480 band – Bit 18 - GSM 750 band – Bit 19 - GSM 850 band – Bit 20 - GSM Railways GSM 900 Band – Bit 21 - GSM PCS 1900 band – Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band – Bit 23 - WCDMA U.S. PCS 1900 band – Bit 24 - WCDMA Europe and China DCS 1800 band – Bit 25 - WCDMA U.S. 1700 band – Bit 26 - WCDMA U.S. 850 band – Bit 27 - WCDMA Japan 800 band – Bit 28 - Band Class 12 – Bit 29 - Band Class 14 – Bit 30 - Reserved – Bit 31 - Band Class 15 – Bit 32 to 47 - Reserved – Bit 48 - WCDMA Europe 2600 band – Bit 49 - WCDMA Europe and Japan 900 band – Bit 50 - WCDMA Japan 1700 band – Bit 51 to 55 - Reserved – Bit 56 - Band Class 16 – Bit 57 - Band Class 17 – Bit 58 - Band Class 18 – Bit 59 - Band Class 19 – Bit 60 to 64 - Reserved • function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.

pPRLPref	• Optional parameter indicating the CDMA PRL Preference • Values: – 0x0001 - Acquire available system only on the A side – 0x0002 - Acquire available system only on the B side – 0x3FFF - Acquire any available systems • function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device.
pRoamPref	• Optional parameter indicating the roaming Preference • Values: – 0x01 - Acquire only systems for which the roaming indicator is off – 0x02 - Acquire a system as long as its roaming indicator is not off – 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only – 0xFF - Acquire systems, regardless of their roaming indicator • function SLQSGetSysSelectionPref() returns a default value FFFF if no value is returned by the device.

pLTEBandPref	• Optional parameter • Bit mask representing the LTE band preference • Bit Values – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – All other bits are reserved • function SLQSGetSysSelectionPref() returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.

pNetSelPref	• Optional parameter indicating network selection preference • Values: – 0x00 - Automatic network selection – 0x01 - Manual network selection • function SLQSGetSysSelectionPref() returns a default value FF if no value is returned by the device.
pSrvDomainPref	• Optional parameter indicating Service domain preference • Values: – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach • 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 • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM • 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 • Values: – 0 - OFF (normal) – 1 - ON (Emergency)
pModePref	• Optional parameter • Bit Mask indicating the radio technology mode preference • Bit values: – Bit 0 - cdma2000 1x – Bit 1 - cdma2000 HRPD(1xEV-DO) – Bit 2 - GSM – Bit 3 - UMTS – Bit 4 - LTE

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

pPRLPref	• Optional parameter indicating the CDMA PRL Preference • Values: – 0x0001 - Acquire available system only on the A side – 0x0002 - Acquire available system only on the B side – 0x3FFF - Acquire any available systems
pRoamPref	• Optional parameter indicating the roaming Preference • Values: – 0x01 - Acquire only systems for which the roaming indicator is off – 0x02 - Acquire a system as long as its roaming indicator is not off – 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only – 0xFF - Acquire systems, regardless of their roaming indicator

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

pNetSelPref	- netSelectionPref • Optional parameter for specifying Network Selection Preference • Modem selects networks based on this parameter(if present). • Either of pNetSelPref or pCSGID can be set. • see netSelectionPref for more information
pChgDuration	• Optional parameter specifying the duration of the change • At least one system selection setting to be set if pChgDuration is populated. • Values: – 0x00 - Power cycle - Remains active until the next device power cycle – 0x01 - Permanent - Remains active through power cycles until changed by client – Device will use "0x01 - permanent" as default if this parameter is omitted
pMNCIncPCS-DigStat	• Optional parameter indicating if MNC includes PCS digit • pNetSelPref is expected if MNC includes PCS digit is set to 1. • Values: – TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
pSrvDomainPref	• Optional parameter indicating Service domain preference • Values: – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach
pGWAcqOrder-Pref	• Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM
pTdscdmaBand-Pref	• Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set. • Values: – 0x01 - TD-SCDMA Band A – 0x02 - TD-SCDMA Band B – 0x04 - TD-SCDMA Band C – 0x08 - TD-SCDMA Band D – 0x10 - TD-SCDMA Band E – 0x20 - TD-SCDMA Band F – All other bits are reserved

pAcqOrderPref	- acqOrderPref • Optional parameter for specifying Acquisition Order Preference • see acqOrderPref for more information
pSrvReg-Restriction	• Optional parameter indicating Network Selection Registration Restriction Preference • Values: – 0x00 - Device follows the normal registration process – 0x01 - Device camps on the network according to its provisioning, but does not register – 0x02 - Device selects the network for limited service – All other values are reserved.
pCSGID	- CSGID • Optional parameter for specifying CSG ID • Either of pNetSelPref or pCSGID can be set. • see CSGID for more information
pRAT	• Optional parameter Radio Access Technology order Preference • Values: – 0x04 - GSM – 0x05 - UMTS – 0x08 - LTE – 0x09 - TDSCDMA

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 – 0 - Failure – 1 - Success – 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 – 0xFF-Unknown
pSCM[OUT]	• Station class mark – 0xFF-Unknown
pRegHomeSID[OUT]	• Register on home SID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pRegForeignSID[OUT]	• Register on foreign SID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pRegForeignNID[OUT]	• Register on foreign NID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pForceRev0[OUT]	• Force CDMA 1x-EV-DO Rev. 0 mode – 0 - Disabled – 1 - Enabled – 0xFF - Unknown

pCustomSCP[OUT]	• Use a custom config for CDMA 1x-EV-DO SCP – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pProtocol[OUT]	• Protocol mask for custom SCP config – 0x00000001 - Subtype 2 Physical Layer – 0x00000002 - Enhanced CCMAC – 0x00000004 - Enhanced ACMAC – 0x00000008 - Enhanced FTCMAC – 0x00000010 - Subtype 3 RTCMAC – 0x00000020 - Subsystem 1 RTCMAC – 0x00000040 - Enhanced Idle – 0x00000080 - Generic Multimode Capable Disc Port – 0xFFFFFFFF - Unknown
pBroadcast[OUT]	• Broadcast mask for custom SCP config – 0x00000001 - Generic broadcast enabled – 0xFFFFFFFF - Unknown
pApplication[OUT]	• Application mask for custom SCP config – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application – 0xFFFFFFFF - Unknown
pRoaming[OUT]	• Roaming preference – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming – 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 – 0xFFFF - Unknown. – Only applies to cdma2000
pNID[OUT]	• Home network ID – 0xFFFF - Unknown. – 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 – 0xFFFF - Unknown. – Only applies to cdma2000
pNID[OUT]	• Home network ID – 0xFFFF - Unknown. – Only applies to cdma2000
pNw2MCC[OUT]	• Mobile country code (3GPP2 only). • Range : 0 to 999
pNw2MNC[OUT]	• Mobile network code (3GPP2 only). • Range : 0 to 999
pNw2DescDisp[OUT]	• Network Name Display (3GPP2 only). -Valid Value – 0x00 - Do not display – 0x01 - Display – 0xFF - Unknown
pNw2DescDisp[OUT]	• Encoding of the network description (3GPP2 only). • Valid Value – 0x00 - Octet, unspecified – 0x02 - 7-bit ASCII – 0x04 - Unicode – 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. • No bits set indicates to the device to automatically determine the technology to use • Values: – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved
pDuration[OUT]	• Duration of active preference – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call
pPersistent-TechnologyPref[-OUT]	• Bit field representing persistent radio technology preference – 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. • Upon successful output, actual number of elements in RF info instances array.
pInstances[OUT]	• RF info instances array – 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: – 0 - Not registered – 1 - Registered – 2 - Searching/Not Registered – 3 - Registration Denied – 4 - Unknown
pCSDomain[OUT]	• Circuit switch domain status: – 0 - Unknown/Not Applicable – 1 - Attached – 2 - Detached

pPSDomain[OUT]	• Packet switch domain status – 0 - Unknown/Not Applicable – 1 - Attached – 2 - Detached
pRAN[OUT]	• Type of radio access network on which mobile is registered: – 0 - Unknown – 1 - cdma2000 network – 2 - UMTS network
pRadioIfacesSize[IN/OUT]	• Upon input, maximum number of elements that the radio interface array contain. • Upon successful output, actual number of elements in the radio interface array.
pRadioIfaces[OUT]	• An array of Radio Interface Technology – 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. • Upon output, the actual number of elements in the data capabilities array.
pDataCaps[OUT]	• Data capabilities array of unsigned long type – 1 - GPRS – 2 - EDGE – 3 - HSDPA – 4 - HSUPA – 5 - WCDMA – 6 - CDMA 1xRTT – 7 - CDMA 1xEV-DO Rev 0 – 8 - CDMA 1xEV-DO Rev. A – 9 - GSM – 10 - EVDO Rev. B – 11 - LTE – 12 - HSDPA Plus – 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.

Parameters

pArraySizes[IN/OUT]	• Upon input maximum number of elements that each array can contain. • 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 – 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 1 - Attach 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 – 1 - Automatic – 2 - Manual
mcc	• Mobile country code

mnc	• Mobile network code
rat	• Radio access technology – 4 - GSM – 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. • Upon successful output, the actual number of elements in the network info instance array.
pInstances[OUT]	• Network info instance array – 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 – 0 - Disabled – 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 – 0 - Disabled – 1 - Enabled Note: Enabled can only be specified if pForceRev0 is set to Disabled

pProtocol[IN]	• Protocol mask for custom SCP config – 0x00000001 - Subtype 2 Physical Layer – 0x00000002 - Enhanced CCMAC – 0x00000004 - Enhanced ACMAC – 0x00000008 - Enhanced FTCMAC – 0x00000010 - Subtype 3 RTCMAC – 0x00000020 - Subsystem 1 RTCMAC – 0x00000040 - Enhanced Idle – 0x00000080 - Generic Multimode Capable Disc Port – 0xFFFFFFFF - Unknown
pBroadcast[IN]	• Broadcast mask for custom SCP config – 0x00000001 - Generic broadcast enabled – 0xFFFFFFFF - Unknown
pApplication[IN]	• Application mask for custom SCP config – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application – 0xFFFFFFFF - Unknown
pRoaming[IN]	• Roaming preference – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming – 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 – No bits set indicates device to automatically determine the technology to use. – Type of technology * Bit 0 - Technology is 3GPP2 * Bit 1 - Technology is 3GPP – Technology-specific protocol bitmask * Bit 2 - Analog · AMPS if 3GPP2, GSM if 3GPP * Bit 3 - Digital · CDMA if 3GPP2, WCDMA if 3GPP * Bit 4 - HDR * Bit 5 - LTE * All other bits are reserved.
duration[IN]	• Duration of active preference – 0 - Persistent – 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

pNasInitNetRegistrationReq[IN]	• Pointer to structure nasInitNetworkReq – 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

pSetSignalStrengthInfo[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 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

9.19.5.30 ULONG SLQSNASGetLTECPHYCaInfo (nasGetLTECphyCa * *pLTECPhyCa*)

This API Get LTE CPHY Carrier Info

Parameters

pLTECPhyCa[I-N]	• 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.

9.19.5.31 ULONG SLQSNasGetSigInfo (nasGetSigInfoResp * *pGetSigInfoResp*)

This API queries information regarding the signal strength.

Parameters

pGetSigInfo-Resp[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

9.19.5.32 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.

9.19.5.33 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.

9.19.5.34 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. tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem – 0x00 - for disable – 0x01 - for enable – 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. – 0x00 - for disable – 0x01 - for enable – 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. tFNBandPreference – 0x00 - for disable – 0x01 - for enable – 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

9.19.5.35 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.

9.19.5.36 ULONG SLQSNasIndicationRegisterLTECphyCa (BYTE * *bStatus*)

This API Registers/De-registers for NAS CPHY Carrier Info.

Parameters

bStatus[IN]	• Values – 0 - De-register. – 1 - Register.

Returns

eQCWWAN_ERR_sNONE on success, eQCWWAN_xxx error value otherwise.

See Also

See qmerrno.h for eQCWWAN_xxx error values.

9.19.5.37 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.

9.19.5.38 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

9.19.5.39 ULONG SLQSNasSwiModemStatus (swiModemStatusResp * *pModemStatusResp*)

This API requests the device to return the current status of modem.

Parameters

pModemStatusResp[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

9.19.5.40 ULONG SLQSNASSwiSetChannelLock (nasSwiSetChannelLockReq * *pNasSwiSetChannelLockReq*)

This API allows the host to lock the UE to a specific channel or cell.

Parameters

pNasSwiSetChannelLockReq[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.

9.19.5.41 ULONG SLQSPerformNetworkScan (slqsNetworkScanInfo * *pNetworkInfo*)

Performs scan for available networks and scans for RAT info as well.

Parameters

pNetworkInfo[IN/OUT]	• See slqsNetworkScanInfo for more information • Valid pointers to the following structure members are mandatory – pNetworkInfoInstances – 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

9.19.5.42 ULONG SLQSSetBandPreference (ULONGLONG *bandpreference*)

Provides information about the band preference.

Parameters

bandpreference[-IN]	• Bit mask representing the band preference to be set. • Bit position meanings: – 0 - BC0_A - Band Class 0, A-System – 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band – 2 - BC1 - Band Class 1, all blocks – 3 - BC2 - Band Class 2 place holder – 4 - BC3 - Band Class 3, A-System – 5 - BC4 - Band Class 4, all blocks – 6 - BC5 - Band Class 5, all blocks – 7 - GSM_DCS_1800 - GSM DCS band – 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band – 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band – 10 - BC6 - Band Class 6 – 11 - BC7 - Band Class 7 – 12 - BC8 - Band Class 8 – 13 - BC9 - Band Class 9 – 14 - BC10 - Band Class 10 – 15 - BC11 - Band Class 11 – 16 - GSM_450 - GSM 450 band – 17 - GSM_480 - GSM 480 band – 18 - GSM_750 - GSM 750 band – 19 - GSM_850 - GSM 850 band – 20 - GSM_RGSM_900 - GSM Railways GSM Band – 21 - GSM_PCS_1900 - GSM PCS band – 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band – 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band – 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band – 25 - WCDMA_IV_1700 - WCDMA US 1700 band – 26 - WCDMA_V_850 - WCDMA US 850 band – 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band – 28 - BC12 - Band Class 12 – 29 - BC14 - Band Class 14 – 30 - RESERVED_2 - Reserved 2 – 31 - BC15 - Band Class 15 – 32 - 47 - Reserved – 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band – 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band – 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band – 51 to 55 - Reserved – 56 - BBC16 - Band Class 16 – 57 - BC17 - Band Class 17 – 58 - BC18 - Band Class 18 – 59 - BC19 - Band Class 19 – 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

9.19.5.43 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

9.19.5.44 ULONG SLQSSwiGetHDRPersonality (HDRPersonalityResp * *pHDRPersonalityResp*)

This API retrieves HDR Personality related information

Parameters

pHDRPersonalityResp[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

9.19.5.45 ULONG SLQSSwiGetHDRProtSubtype (HDRProtSubtypResp * *pHDRProtSubtypResp*)

This API retrieves HDR Prototype Subtype related information

Parameters

pHDRProtSubtypResp[OUT]	• 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

9.19.5.46 ULONG SLQSSwiGetHRPDStats (GetHRPDStatsResp * *pGetHRPDStatsResp*)

This API retrieves currently acquired HRPD system statistics

Parameters

pGetHRPDStatsResp[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

9.19.5.47 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

9.19.5.48 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

9.19.5.49 ULONG SLQSSwiNetworkDebug (NetworkDebugResp * *pNetworkDebugResp*)

This API retrieves device and network status details

Parameters

pNetworkDebugResp[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

9.19.5.50 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 – 0x04 - Network-initiated PRL update – 0x05 - Network-initiated device configure
SessionID[OUT]	• Session Id – 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 – 0x00 - Complete, information was updated – 0x01 - Complete, update information is unavailable – 0x02 - Failed – 0x03 - Retrying – 0x04 - Connecting – 0x05 - Connected – 0x06 - Authenticated – 0x07 - Mobile Directory Number (MDN) downloaded – 0x08 - Mobile Station Identifier (MSID) downloaded – 0x09 - PRL downloaded – 0x0A - Mobile IP (MIP) profile downloaded
sessionType[OUT]	• Session State – 0x00 - Client-initiated device configure – 0x01 - Client-initiated PRL update – 0x02 - Client-initiated hands-free activation – 0x03 - Device-initiated hands-free activation – 0x04 - Network-initiated PRL update – 0x05 - Network-initiated device configure
FailureReason[OUT]	• Session failure reason (when state indicates failure) – 0x00 - Unknown – 0x01 - Network is unavailable – 0x02 - Server is unavailable – 0x03 - Authentication failed – 0x04 - Maximum retry exceeded – 0x05 - Session is cancelled
RetryCount[OUT]	• Session retry count (when state indicates retrying)
SessionPause[OUT]	• Session pause timer (in seconds , when state indicates retrying)
TimeRemaining[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 – 0x00 - Client-initiated device configure – 0x01 - Client-initiated PRL update – 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 – 0 - Standalone – 1 - MS based – 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 – 0 - disable – 1 - enable
pTracking-Status[OUT]	• Current PDS tracking session state • Values: – 0x00 - Unknown – 0x01 - Inactive – 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 – 0x00 - Disabled – 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 – 0x00 - Disabled – 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 – 0 - Disabled – 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 – 0x00 - None (any available network) – 0x01 - Home-only, only when on home systems – 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)
system-Discontinuities	• 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) – 0x00000001 - EPH – 0x00000002 - ALM – 0x00000004 - POS – 0x00000008 - TIME – 0x00000010 - IONO – 0x00000020 - UTC – 0x00000040 - HEALTH – 0x00000080 - SVDIR – 0x00000100 - SVSTEER – 0x00000200 - SADATA – 0x00000400 - RTI – 0x00000800 - ALM_CORR – 0x00001000 - FREQ_BIAS_EST

pCellDataMask[-IN]	• Bitmask of cell data to clear (optional) – 0x00000001 - POS – 0x00000002 - LATEST_GPS_POS – 0x00000004 - OTA_POS – 0x00000008 - EXT_REF_POS – 0x00000010 - TIMETAG – 0x00000020 - CELLID – 0x00000040 - CACHED_CELLID – 0x00000080 - LAST_SRV_CELL – 0x00000100 - CUR_SRV_CELL – 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 – 0 - Standalone – 1 - MS based – 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 – Zero - disable – 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 – 0x00 - Disabled – 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 StartPDSTrackingSessionExt 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 – 0x00 - Disabled – 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 – 0 - Disabled – 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 – 0x00 - None (any available network) – 0x01 - Home-only, only when on home systems – 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

pServerAddress[OUT]	• IPv4 address of AGPS server. "0" if not set
pServerPort[OUT]	• Port number of AGPS server. "0" if not set
pServerURL[OUT]	• URL of the AGPS server. "0" if not set
pServerURLLength[OUT]	• URL length of AGPS server. "0" if not set

pNetworkMode[-IN]	• Network Mode of AGPS Server [optional - should be present in Multimode Systems] – 0x00 - UMTS – 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 • 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 – 0x00 - GPS (midnight, Jan 6, 1980) – 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] – 0x00 - UMTS – 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: – 0x0 - Manual
sessionType[IN]	• Type: – 0x0 - New
sessionOperation[IN]	• Operating mode: – 0x00 - Standalone – 0x01 - MS-based
sessionServerOption[IN]	• Location server option: – 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)

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 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: • 0x01 – QMI_QOS_STATUS_ACTIVATED • 0x02 – QMI_QOS_STATUS_SUSPENDED • 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

9.22.3.2 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 • Index identifying the QoS flow that has been negotiated
in	*pGranted[OUT]*	• Tx/Rx Qos granted flow • 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

9.22.3.3 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 • 0 – No QoS support in network • 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

9.22.3.4 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

9.22.3.5 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

9.22.3.6 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

9.22.3.7 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

9.22.3.8 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

9.22.3.9 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

9.22.3.10 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

9.22.3.11 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

9.22.3.12 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 – 0 - Disabled – 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 – Zero - Disable – 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 • 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

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 – 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

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 – 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

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 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 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	- • Optional 1 BYTE parameter indicating registration status of transport layer information events • Values: – 0x00 - Disabled – 0x01 - Enabled • 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	- • Optional 1 BYTEparameter indicating registration status of transport network registration information events • Values: – 0x00 - Disabled – 0x01 - Enabled • 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	- • Optional 1 BYTE parameter indicating registration status of call status information events • Values: – 0x00 - Disabled – 0x01 - Enabled • 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	- • Optional parameter indicating if transport layer is registered • Values: – 0x00 - Transport layer is not registered – 0x01 - Transport layer is registered • function SLQSGetTransLayerInfo() returns a default value 0xFF if no response is received from the device.
pTransLayerInfo	• Pointer to structure of transLayerInfo. – Optional parameter – See transLayerInfo for more information • 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	- • Optional 1 BYTE parameter indicating transport layer network registration status • Values: – 0x00 - No service – 0x01 - In progress – 0x02 - Failed – 0x03 - Limited Service – 0x04 - Full Service • 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 – 0x00 - Deactivated – 0x01 - Activated
num_instances	• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. – serviceCategory – language – selected
broadcastConfig	• A CDMABroadcastConfig structure array. • 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 – 0x00 - Deactivated – 0x01 - Activated

num_instances	• Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table. – fromServiceId – toServiceId – selected
broadcastConfig	• A BroadcastConfig structure array. • Further defined by the structure BroadcastConfig

9.25.3.6 typedef struct _setIndicationRegReq setIndicationRegReq

This structure contains Indication Register request parameters

Parameters

pRegTrans-LayerInfoEvt	- • Optional 1 BYTE parameter indicating registration status of transport layer information events • Values: – 0x00 - Disabled – 0x01 - Enabled – 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	- • Optional 1 BYTE parameter indicating registration status of transport network registration information events • Values: – 0x00 - Disabled – 0x01 - Enabled – 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	- • Optional 1 BYTE parameter indicating registration status of call status information events • Values: – 0x00 - Disabled – 0x01 - Enabled – 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 – 0x00 - IMS
TransCap	• Transport Capability • Values: – 0x00 - CDMA – 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 – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
messageFormat[IN]	• Message format – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 6 - GSM/WCDMA PP
messageSize	• The length of the message contents in bytes
pMessage[IN]	• The message contents
pMessageIndex[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 – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 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 • The message is to be sent on IMS. – 0x00 Message is not to be sent on IMS. – 0x01 Message is to be sent on IMS. – 0x02 to 0xFF Reserved.
pMessage-FailureCode[OUT]	• (Optional) Message Failure Code • 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 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

9.25.4.7 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 – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
pMessage-Index[IN]	• (Optional) message index
pMessageTag[I-N]	• (Optional) message tag – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent
pMessage-Mode[IN]	• (Optional) messgae mode • 0x00 - CDMA, LTE (if network type is CDMA) • 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

9.25.4.8 ULONG SLQSGetIndicationRegister (getIndicationRegResp * *pGetIndicationRegInfo*)

This API provides registration state of different WMS indications.

Parameters

pGetIndication-RegInfo	[OUT] • Pointer to structure of getIndicationRegResp – 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

9.25.4.9 ULONG SLQSGetMessageWaiting (getMsgWaitingInfo * *pGetMsgWaitingInfoResp*)

This API provides information about the message waiting information.

Parameters

pGetMsg-WaitingInfoResp	[OUT] • Pointer to structure of getMsgWaitingInfoResp – 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

9.25.4.10 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 – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
messageIndex	• Message index
pMessageTag[-OUT]	• Message tag – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent

pMessage-Format[OUT]	• Message format – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 6 - GSM/WCDMA PP
pMessageSize[IN/OUT]	• 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.
pMessage[OUT]	• The message contents array
pMessage-Mode[IN]	• (Optional) Message Mode • 0x00 - CDMA, LTE (if network type is CDMA) • 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

9.25.4.11 ULONG SLQSGetSmsBroadcastConfig (BYTE *mode,* qaQmi3GPPBroadcastCfgInfo * *pBroadcastConfig,* qaQmi3GPP2BroadcastCfgInfo * *pCDMABroadcastConfig*)

Provides Information about the SMS BroadcastConfiguration

Parameters

mode[IN]	• Mode – 0x00 - CDMA, LTE (if network type is CDMA) – 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

9.25.4.12 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 – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
pRequested-Tag[IN]	• (Optional) Message tag – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 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. • 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 • 0x00 - CDMA, LTE (if network type is CDMA) • 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

9.25.4.13 ULONG SLQSGetTransLayerInfo (getTransLayerInfoResp * *pGetTransLayerInfoResp*)

This API provides information about the transport layer.

Parameters

pGetTransLayer-InfoResp	[OUT] • Pointer to structure of getTransLayerInfoResp – 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

9.25.4.14 ULONG SLQSGetTransNWRegInfo (getTransNWRegInfoResp * *pGetTransNWRegInfoResp*)

This API provides transport layer network registration info.

Parameters

pGetTransNW-RegInfoResp	[OUT] • Pointer to structure of getTransNWRegInfoResp – 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

9.25.4.15 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 – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
messageIndex[IN]	• Message index
messageTag[IN]	• Message tag – 0 - Read – 1 - Not read
pMessageMode[IN]	• (Optional) Message Mode • 0x00 - CDMA, LTE (if network type is CDMA) • 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

9.25.4.16 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

9.25.4.17 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 – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 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 – 0 - 7 bit encoding – 4 - 8 bit encoding – 8 - 16 bit UCS2 encoding – 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

9.25.4.18 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

9.25.4.19 ULONG SLQSSetIndicationRegister (setIndicationRegReq * *pSetIndicationRegReq*)

This API sets the registration state of different WMS indications.

Parameters

pSetIndication-RegReq	[IN] • Pointer to structure of indicationRegReqParams – 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

9.25.4.20 ULONG SLQSSetSmsBroadcastActivation (BYTE *mode,* BYTE *broadcastActivate*)

Enables or disables the reception of broadcast SMS messages.

Parameters

Mode[IN]	• Mode • 0x00 - CDMA, LTE (if network type is CDMA) • 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

9.25.4.21 ULONG SLQSSetSmsBroadcastConfig (BYTE *mode*, qaQmi3GPPBroadcastCfgInfo * *pBroadcastConfig*, qaQmi3GPP2BroadcastCfgInfo * *pCDMABroadcastConfig*)

Sets the information about the SMS BroadcastConfiguration

Parameters

mode[IN]	• Mode – 0x00 - CDMA, LTE (if network type is CDMA) – 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

9.25.4.22 ULONG SLQSSetSmsStorage (BYTE *smsStorage*)

Sets the SMS Storage on the device

Parameters

smsStorage[IN]	• SMS Storage – 0x01 - device's permanent memory – 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

9.25.4.23 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 – See smsMaxStorageSizeReq for more information
pMaxStorage-SizeResp[OUT]	• Response parameters for SmsSLQSGetMaxStorageSize – 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

9.25.4.24 ULONG SLQSSmsGetMessageProtocol (smsMsgprotocolResp * *pMessageProtocol*)

This API queries the message protocol currently in use for the WMS client.

Parameters

pMessage-Protocol	[OUT} • Pointer to smsMsgprotocolResp – 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

9.25.4.25 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] • Pointer to structure of smsSetRoutesReq – 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

9.25.4.26 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: – 0x01 - device's permanent memory – 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

9.25.4.27 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

9.25.4.28 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

9.25.4.29 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 • 1 - Firmware. • 2 - User App. • 3 - Legato Framework.
bStat	: State • 0x01 - No binary update available. • 0x02 - Query binary Download. • 0x03 - Binary Downloading. • 0x04 - Binary downloaded. • 0x05 -Query Binary Update. • 0x06 - Binary updating. • 0x07 - Binary updated.

bUserInput-Request	Bit mask of available user inputs. • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 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. • 0x01 - Mandatory. • 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. • 0x01 - AVMS Read Request. • 0x02 - AVMS Change Request. • 0x03 - AVMS Config Complete.
bUserInput-Request	Bit mask of available user inputs. • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 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. • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 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. • 0: Data session closed. • 1: Data session activated. • 2: Register made. -3: Data session error.
wErrorCode	LWM2M Session error code. • 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. • 100 - Continue • 101 - Switching Protocols • 200 - OK • 201 - Created • 202 - Accepted • 203 - Non-Authoritative Information • 204 - No Content • 205 - Reset Content • 206 - Partial Content • 300 - Multiple Choices • 301 - Moved Permanently • 302 - Found • 303 - See Other • 304 - Not Modified • 305 - Use Proxy • 307 - Temporary Redirect • 400 - Bad Request • 401 - Unauthorized • 402 - Payment Required • 403 - Forbidden • 404 - Not Found • 405 - Method Not Allowed • 406 - Not Acceptable • 407 - Proxy Authentication Required • 408 - Request Timeout • 409 - Conflict • 410 - Gone • 411 - Length Required • 412 - Precondition Failed • 413 - Payload Too Large • 414 - URI Too Long • 415 - Unsupported Media Type • 416 - Range Not Satisfiable • 417 - Expectation Failed • 426 - Upgrade Required • 500 - Internal Server Error • 501 - Not Implemented • 502 - Bad Gateway • 503 - Service Unavailable • 504 - Gateway Timeout • 505 - HTTP Version Not Supported
TlvPresent	Boolean indicating the presence of the TLV in the QMI response

9.28.3.6 typedef struct _QmiSwiAvmsEventReportNotificationResp QmiSwiAvmsEventReportNotificationResp

Parameters

bNotification	Notification. • 0x14 - Module starts sending data to server. • 0x15 - Authentication with the server. • 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. • 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 wasa updated • 0x7F16: Factory reset successful: Factory reset succeeded • 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. • 0: Need Bootstrap. • 1: Bootstrap made. • 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. • 0: Bootstrap session. • 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. • By default set to 0xFF for all changes. • 0x01 - device_login • 0x02 - device_MD5_key • 0x04 - server_login • 0x08 - server_MD5_key • 0x10 - server_URL • 0x20 - Nonce • 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 – 0x00 - FALSE – 0x01 - TRUE
bAutoReboot[IN]	• Automatic device reboot when the request is received from AirVantage server – 0x00 - FALSE – 0x01 - TRUE

bFw-Autodownload[I-N]	• Firmware Auto Download – 0x00 - FALSE – 0x01 - TRUE
bFwAuto-Update[IN]	• Firmware Auto Update – 0x00 - FALSE – 0x01 - TRUE
FwAutoSDM[-Optional]	- OMA Automatic UI Alert Response • 0x00 - DISABLED • 0x01 - ENABLED ACCEPT • 0x02 - ENABLED REJECT
PollingTimer[-Optional]	• 0-525600 (min) • 0:disabled
Connection-RetryTimers[-Optional]	• See QmiSwiAvmsSetSettingsConnectionRetryTimers for more information
APNInfo[-Optional]	• See QmiSwiAvmsSetSettingsAPNInfo for more information
NotifStore[-Optional]	• 0 : Disabled. • 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 SLQSAVMSSettingsConnectionRetry-Timers

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[O-UT]	• 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. – 0x01 - Accept. – 0x02 - Reject. – 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. – 0 - if modem performs the operation (download or update). – 1 - if client performs the operation (download or update).
pPackageID[IN]	• Package ID (relevant to Application binary package) – 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 – 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 – 0x01 -Available. – 0x02 -Not Available. – 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 – 0x01 - FOTA, to check availability of FW Update. – 0xFF - Cancel any active OMADM session.
pSessionRep[OUT]	• Session – 0x01 - FOTA session suspended. – 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 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)

9.29.1 Detailed Description

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

9.29.2 Typedef Documentation

9.29.2.1 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) – 0x01 - No Firmware available – 0x02 - Query Firmware Download – 0x03 - Firmware Downloading – 0x04 - Firmware Downloaded – 0x05 - Query Firmware Update – 0x06 - Firmware Updating – 0x07 - Firmware Updated
pUpdate-CompleteStatus	• 2 byte parameter indicating Update Complete Status(optional) – See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h Update Complete Status
pSeverity	• 1 byte parameter indicating severity(optional) – 0x01 - Mandatory – 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) – 0x00 - No session since boot – 0x01 - Sprint CI-DC Session – 0x02 - Sprint CI-PRL Session – 0x03 - Sprint CI-FUMO Session – 0x04 - Sprint HFA-DC Session – 0x05 - Sprint HFA-PRL Session – 0x06 - Sprint HFA-FUMO Session – 0x07 - Sprint NI Session
pSessionState	• 1 byte parameter indicating session state(optional) – 0x01 - idle – 0x02 - active – 0x03 - pending
pRetryCount	• 1 byte parameter indicating retries left count(optional) – 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.

9.29.2.2 typedef struct _SLQSOMADMSettings SLQSOMADMSettings

Structure containing the OMA DM settings retrieved from the device

Parameters

pOMADMEnabled[OUT]	• 4 byte parameter indicating OMADM service enabled – 0x00000001 - Client-initiated device configuration – 0x00000002 - Network-initiated device configuration – 0x00000010 - Client-initiated FUMO – 0x00000020 - Network-initiated FUMO • function SLQSOMADMGetSettings2() returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.

pFOTAdownload[OUT]	• 1 Byte parameter indicating support for FOTA Automatic download – 0x00 - Host permission required before downloading – 0x01 - Automatically start downloading, no host permission required – 0x02 - Automatically start downloading, while not roaming – 0x03 - Automatically reject download – 0x04 - Automatically reject download with “Enterprise Reject Policy” • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
pFOTAUpdate[OUT]	• 1 byte parameter indicating FOTA Automatic update – 0x00 - User permission required before updating firmware – 0x01 - No user permission required before updating firmware – 0x02 - User permission required, auto update on power up • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
pAutosdm[OUT]	• 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject • function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
pFwAutoCheck[OUT]	• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 0x01 - Enabled • 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

9.29.2.3 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 – 0x00 - Firmware auto download FALSE – 0x01 - Firmware auto download TRUE
FOTAUpdate	• 1 byte parameter indicating FOTA Automatic update – 0x00 - Firmware auto update FALSE – 0x01 - Firmware auto update TRUE
pAutosdm[IN]	• Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.29.2.4 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 – 0x00 - Firmware auto download FALSE – 0x01 - Firmware auto download TRUE
FOTAUpdate	• 1 byte parameter indicating FOTA Automatic update – 0x00 - Firmware auto update FALSE – 0x01 - Firmware auto update TRUE
pAutosdm[IN]	• Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject
pFwAutoCheck[IN]	• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 0x01 - Enabled

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.29.3 Function Documentation

9.29.3.1 ULONG SLQSOMADMCancelSession (ULONG *session*)

Cancels an ongoing OMA-DM session.

Parameters

session[IN]	• Session – 0x01 - FOTA, to check availability of FW Update – 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

9.29.3.2 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 – 0x01 - FOTA – 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

9.29.3.3 ULONG SLQSOMADMGetSettings (ULONG * *pbOMADMEnabled,* ULONG * *pbFOTAdownload,* ULONG * *pbFOTAUpdate*)

Returns the OMA-DM settings.

Parameters

pbOMADMEnabled[OUT]	• Device OMADM service enabled – 0x00000001 - Client-initiated device configuration – 0x00000002 - Network-initiated device configuration – 0x00000010 - Client-initiated FUMO – 0x00000020 - Network-initiated FUMO
pbFOTAdownload[OUT]	• Firmware AutoDownload – 0x00 - Firmware auto download FALSE – 0x01 - Firmware autod ownload TRUE – 0x02 - Automatically start downloading, while not roaming – 0x03 - Automatically reject download – 0x04 - Automatically reject download with “Enterprise Reject Policy”
pbFOTAUpdate[OUT]	• Firmware AutoUpdate – 0x00 - Firmware auto update FALSE – 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

9.29.3.4 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

9.29.3.5 ULONG SLQSOMADMSendSelection (ULONG *selection*)

Sends the specified OMA-DM selection for the current network initiated session.

Parameters

selection[IN]	• OMA-DM NIA Selection – 0x01 - Accept – 0x02 - Reject – 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

9.29.3.6 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 – 0x01 - Accept – 0x02 - Reject – 0x03 - Defer

pDeferTime[IN]	• Defer time in minutes. A value of 0 will cause the prompt to be resent immediately. • This TLV is mandatory if selection is set to 0x03.
pRejectReason[IN]	• Reject Reason • This TLV is processed if selection is set to 0x02. If it is not present, the reject reason 0 is used as default.

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

9.29.3.7 ULONG SLQSOMADMSetSettings (ULONG *bFOTAdownload,* ULONG *bFOTAUpdate*)

Sets the OMA-DM settings requested.

Parameters

bFOTAdownload[IN]	• Firmware Auto Download – 0x00 - Host permission required before downloading – 0x01 - Automatically start downloading, no host permission required – 0x02 - Automatically start downloading, while not roaming – 0x03 - Automatically reject download – 0x04 - Automatically reject download with “Enterprise Reject Policy”
bFOTAUpdate[IN]	• Firmware Auto Update – 0x00 - User permission required before updating firmware – 0x01 - No user permission required before updating firmware – 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

9.29.3.8 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

9.29.3.9 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

9.29.3.10 ULONG SLQSOMADMStartSession (ULONG *sessionType*)

Starts an OMA-DM session.

Parameters

sessionType[IN]	• Session type – 0x01 - FOTA, to check availability of FW Update – 0x02 - DM, to check availability of DM Update – 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

9.29.3.11 ULONG SLQSOMADMStartSession2 (ULONG *sessionType*, ULONG * *pFwAvailability*)

Starts an OMA-DM session.

Parameters

sessionType[IN]	• Session type – 0x01 - FOTA, to check availability of FW Update – 0x02 - DM, to check availability of DM Update – 0x03 - PRL, to check availability of PRL Update
pFwAvailability[-OUT]	• OMA-DM CHECK FW Available – 0x00000001 - FW Available. For CIDC and CIPRL, this value will be returned by the modem. CIDC and CIPRL are asynchronous OMADM sessions. – 0x00000002 - FW Not Available – 0x00000003 - FW Check Timed Out

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 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

9.30.1 Detailed Description

Network Access Service API Band Classes table.

9.30.2 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

9.30.2.1 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

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.31 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

9.31.1 Detailed Description

Call Control Return Reasons table.

9.31.2 Call Control Result Reasons (Value - Name - Description)

- 0x01 - QMI_VOICE_REASON_FWD_UNCONDITIONAL - Unconditional call forwarding
- 0x02 - QMI_VOICE_REASON_FWD_MOBILEBUSY - Forward when the mobile is busy
- 0x03 - QMI_VOICE_REASON_FWD_NOREPLY - Forward when there is no reply
- 0x04 - QMI_VOICE_REASON_FWD_UNREACHABLE - Forward when the call is unreachable
- 0x05 - QMI_VOICE_REASON_FWD_ALLFORWARDING - All forwarding
- 0x06 - QMI_VOICE_REASON_FWD_ALLCONDITIONAL - All conditional forwarding
- 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing
- 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal

- 0x09 - QMI_VOICE_REASON_BARR_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

9.32 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

9.32.1 Detailed Description

Wireless Data Service Call End Reasons.

9.32.2 Call end reason codes (Code - Reason)

9.32.2.1 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

9.32.2.2 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

9.32.2.3 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

9.32.2.4 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

9.32.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

9.32.2.6 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

9.32.2.7 Internal call end reasons (Type=2)

- 200 - INTERNAL MIN, internal error table offset value, no meaningful message to the error.
- 201 - INTERNAL ERROR, this error code is returned when data call is brought down due to some unspecified internal error
- 202 - CALL ENDED
- 203 - INTERNAL UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 204 - UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 205 - CLOSE IN PROGRESS
- 206 - NETWORK INITIATED TERMINATION
- 207 - APP PREEMPTED
- 208 - ERR PDN IPV4 CALL DISALLOWED, this error code is returned when V4 PDN is in throttled state due to network providing only V6 address during the previous VSNCP bring up (subs_limited_to_v6). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 209 - ERR PDN IPV4 CALL THROTTLED, this error code is returned when V4 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 210 - ERR PDN IPV6 CALL DISALLOWED, this error code is returned when V6 PDN is in throttled state due to network providing only V4 address during the previous VSNCP bring up (subs_limited_to_v4). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 211 - ERR PDN IPV6 CALL THROTTLED, this error code is returned when V6 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 212 - MODEM RESTART
- 213 - PDP PPP NOT SUPPORTED
- 214 - UNPREFERRED RAT, this error code is returned when data call is brought down since the RAT on which the data call is attempted/connected is no longer the preferred RAT
- 215 - PHYS LINK CLOSE IN PROGRESS, this error code is returned when data call bring up is rejected because physical link is in the process of cleanup
- 216 - APN PENDING HANDOVER, this error code is returned when interface bring up is attempted for an APN that is yet to be handed over to target RAT
- 217 - PROFILE BEARER INCOMPATIBLE

- 218 - MMGSDI CARD EVT, this error code is returned when data call is brought down because card got refreshed/removed
- 219 - LPM OR PWR DOWN, this error code is returned when data call is brought down because device is going into lower power mode or powering down
- 220 - APN DISABLED, this error code is returned when APN is disabled in card
- 221 - MPIT EXPIRED, this error code is returned when data call is brought down because maximum PPP inactivity timer expired
- 222 - IPV6 ADDR TRANSFER FAILED
- 223 - TRAT SWAP FAILED
- 224 - EHRPD TO HRPD FALLBACK, this error code is returned when data call is brought down because device falls back from eHRPD to HRPD (not because of OOS on eHRPD but due to operator/spec driven eHRPD to HRPD fallback requirements)
- 225 - MANDATORY APN DISABLED, this error code is returned when any mandatory APN is disabled, and MinApnList Disallow call config item is set to TRUE in device
- 226 - MIP CONFIG FAILURE, this error code is returned when UE is in MIP Only config (QCMIP=2) but MIP config fails on call bring up due to incorrect provisioning

9.32.2.8 Call Manager defined call end reasons (Type=3)

- 500 - CDMA LOCK, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to device in CDMA locked state
- 501 - INTERCEPT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an intercept order from the base station
- 502 - REORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a reorder from base station
- 503 - REL SO REJ, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with reason: SO Reject
- 504 - INCOM CALL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an incoming call from base station
- 505 - ALERT STOP, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RL/FL fade (or) receiving call release from base stations
- 506 - ACTIVATION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to channel acquisition failures. This indicates that device has failed acquiring all the channels in the PRL
- 507 - MAX ACCESS PROBE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes transmitted
- 508 - CCS NOT SUPPORTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since concurrent service is not supported by base station
- 509 - NO RESPONSE FROM BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since there is no response received from base station
- 510 - REJECTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to base station rejecting the call
- 511 - INCOMPATIBLE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since concurrent services requested were not compatible

- 512 - ALREADY IN TC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since traffic channel is already up for voice calls
- 513 - USER CAL ORIG DURING GPS
- 514 - USER CAL ORIG DURING SMS, this error code is returned when data call is brought down because traffic channel request got rejected since SMS is ongoing
- 515 - NO CDMA SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have CDMA service
- 516 - MC ABORT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since MC aborted the origination/conversation
- 517 - PSIST NG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to persistence test failure
- 518 - UIM NOT PRESENT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RUIM not present
- 519 - RETRY ORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a retry order from base station
- 520 - ACCESS BLOCK, this error code is returned when data call is brought down because traffic channel rejected/released due to Access blocked by base station
- 521 - ACCESS BLOCK ALL, this error code is returned when data call is brought down because traffic channel rejected due to Access blocked by the base station for all mobile devices
- 522 - IS707B MAX ACC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes for IS-707B call
- 523 - THERMAL EMERGENCY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) to put device in thermal emergency
- 524 - CALL ORIG THROTTLED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since call origination is throttled by DCTM module
- 525 - USER CALL ORIGINATE DURING VOICE CALL, this error code is returned when data call is brought down because traffic channel got released by CM(Call Manager) in favor of voice call or SMS when concurrent voice and data are not supported
- 1000 - CONFERENCE FAILED
- 1001 - INCOMING REJECTED
- 1002 - NO GATEWAY SERVICE
- 1003 - NO GPRS CONTEXT
- 1004 - ILLEGAL MS, This cause is sent to the MS when the network refuses service to the MS either because an identity of the MS is not acceptable to the network or because the MS does not pass the authentication check, i.e. the SRES received from the MS is different from that generated by the network
- 1005 - ILLEGAL ME, This cause is sent to the MS if the ME used is not acceptable to the network, e.g. blacklisted
- 1006 - GPRS SERVICES AND NON GPRS SERVICES NOT ALLOWED
- 1007 - GPRS SERVICES NOT ALLOWED
- 1008 - MS IDENTITY CANNOT BE DERIVED BY THE NETWORK
- 1009 - IMPLICITLY DETACHED, this error code is sent to the MS either if the network has implicitly detached the MS, e.g. some while after the Mobile reachable timer has expired, or if the GMM context data related to the subscription does not exist in the SGSN e.g. because of a SGSN restart.

- 1010 - PLMN NOT ALLOWED, this error code is sent to the MS if it requests location updating in a PLMN where the MS, by subscription or due to operator determined barring is not allowed to operate
- 1011 - LOCAL AREA NOT ALLOWED
- 1012 - GPRS SERVICES NOT ALLOWED IN THIS PLMN
- 1013 - PDP DUPLICATE
- 1014 - USER EQUIPMENT RADIO ACCESS TECHNOLOGY CHANGE
- 1015 - CONGESTION
- 1016 - NO PDP CONEXT ACTIVATED
- 1017 - ACCESS CLASS DSAC REJECTION
- 1018 - PDP ACTIVATE MAX RETRY FAILED
- 1019 - RAB FAILURE
- 1020 - EPS SERVICE NOT ALLOWED
- 1021 - TRACKING AREA NOT ALLOWED
- 1022 - ROAMING NOT ALLOWED IN THIS TRACKING AREA
- 1023 - NO SUITABLE CELLS IN TRACKING AREA
- 1024 - NOT AUTHORIZED FOR THIS CLOSED SUBSCRIBER GROUP
- 1025 - ESM UNKNOWN EPS BEARER CONTEXT
- 1026 - DRB RELEASED AT RRC
- 1027 - NAS SIG CONN RELEASED
- 1028 - EPS MOBILITY MANAGEMENT DETACHED
- 1029 - EPS MOBILITY MANAGEMENT ATTACH FAILED
- 1030 - EPS MOBILITY MANAGEMENT ATTACH STARTED
- 1031 - LTE NAS SERVICE REQ FAILED
- 1032 - ESM(EPS Session Management) ACTIVE DEDICATED BEARER REACTIVATED BY NW
- 1033 - ESM(EPS Session Management) LOWER LAYER FAILURE
- 1034 - ESM(EPS Session Management) SYNC UP WITH NW
- 1035 - ESM(EPS Session Management) NW ACTIVATED DED BEARER WITH ID OF DEF BEARER
- 1036 - ESM(EPS Session Management) BAD OTA MESSAGE
- 1037 - ESM DS REJECTED THE CALL
- 1038 - ESM(EPS Session Management) CONTEXT TRANSFERRED DUE TO IRAT
- 1039 - DS EXPLICIT DEACT
- 1040 - ESM(EPS Session Management) LOCAL CAUSE NONE
- 1041 - LTE NAS SERVICE REQ FAILED NO THROTTLE
- 1042 - ACL FAILURE, This error code should rarely triggered and reported to the application
- 1043 - LTE NAS SERVICE REQ FAILED DS DISALLOW
- 1044 - EMM(EPS Mobility Management) T3417 EXPIRED

- 1045 - EMM(EPS Mobility Management) T3417 EXT EXPIRED
- 1046 - 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

9.32.2.9 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

9.32.2.10 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

9.32.2.11 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

9.32.2.12 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

9.33 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

9.33.1 Detailed Description

Carrier Codes table.

9.33.2 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

9.34 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

Macros

- #define __GOBI_API_CODING_SCHEME_H__

9.34.1 Detailed Description

Data Coding Scheme.

9.34.2 Call Control Result Reasons (Value - Name - Description)

9.34.2.1 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

9.34.3 Coding Group Bits 7..4(0001)

9.34.3.1 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.

9.34.4 Coding Group Bits 7..4(0010)

9.34.4.1 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

9.34.5 Coding Group Bits 7..4(0011)

9.34.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS

9.34.6 Coding Group Bits 7..4(01xx)

9.34.6.1 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

9.34.7 Coding Group Bits 7..4(1001)

9.34.7.1 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

9.34.8 Coding Group Bits 7..4(1010..1101)

9.34.8.1 Reserved coding groups

9.34.9 Coding Group Bits 7..4(1110)

9.34.9.1 Defined by the WAP Forum

9.34.10 Coding Group Bits 7..4 (1111)

9.34.10.1 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

9.34.11 Macro Definition Documentation

9.34.11.1 #define __GOBI_API_CODING_SCHEME_H__

9.35 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

9.35.1 Detailed Description

Position Determination Service API GPS Capability Codes.

9.35.2 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

9.36 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

9.36.1 Detailed Description

Device Management Service API Power Modes table.

9.36.2 Power Modes (Value - Description)

- 0 - Online (default)
- 1 - Low power (airplane) mode
- 2 - Factory test mode
- 3 - Offline
- 4 - Reset
- 5 - Power off
- 6 - Persistent low power (airplane) mode
- 7 - Mode - only low power

Valid transitions for Power Modes

- Online to Low Power, Persistent low power, Factory test, Offline or Shut Down
- Low power to online, Persistent low power, Offline, or Shut Down
- Persistent low power to Online, Low power, Offline or Shut down
- Factory test to online
- Offline to Reset

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.37 qaGobiApiTableRadioInterfaces.h File Reference

Network Access Service API Radio Interfaces table.

9.37.1 Detailed Description

Network Access Service API Radio Interfaces table.

9.37.2 Radio interface

9.37.2.1 Technology (Value - Radio Interface Technology)

- 0 - No service
- 1 - CDMA 1xRTT
- 2 - CDMA 1xEV-DO
- 3 - AMPS (Unsupported)
- 4 - GSM
- 5 - UMTS
- 6 - WLAN
- 7 - GPS
- 8 - LTE

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.38 qaGobiApiTableRegionCodes.h File Reference

Region Codes table.

9.38.1 Detailed Description

Region Codes table.

9.38.2 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

9.39 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

9.39.1 Detailed Description

Voice Service Options.

9.39.2 Service Option codes (Code - Reason)

9.39.2.1 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

9.40 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

9.40.1 Detailed Description

Voice Supplementary Service Information Classes.

9.40.2 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

9.41 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

9.41.1 Detailed Description

Swi Audio related tables.

9.41.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

9.41.3 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

9.42 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

9.42.1 Detailed Description

Update Complete Status table.

9.42.2 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

9.43 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

9.43.1 Detailed Description

Voice Service Call and supplementary services end reasons.

9.43.2 Voice Call and supplementary services end reason codes (Code - Reason)

9.43.2.1 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

9.43.2.2 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

9.43.2.3 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

9.43.2.4 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

9.43.2.5 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

9.43.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

9.43.2.7 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

9.43.2.8 reject causes

- 237 - Service option is not supported

9.43.2.9 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

9.44 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)

9.44.1 Detailed Description

Thermal Mitigation Device API function prototypes.

9.44.2 Macro Definition Documentation

9.44.2.1 #define MAX_MITIGATION_DEV_ID_LEN 255

9.44.2.2 #define MAX_MITIGATION_DEV_LIST_LEN 255

9.44.3 Function Documentation

9.44.3.1 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

9.44.3.2 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

9.44.3.3 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

9.44.3.4 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

9.45 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)

9.45.1 Detailed Description

Uim Service API function prototypes.

9.45.2 Macro Definition Documentation

9.45.2.1 #define MAX_ACTIVE_PERS_FEATURES 12

9.45.2.2 #define MAX_CONTENT_LENGTH 1024

9.45.2.3 #define MAX_DESCRIPTION_LENGTH 255

9.45.2.4 #define MAX_ICCID_LENGTH 255

9.45.2.5 #define MAX_NO_OF_APPLICATIONS 10

9.45.2.6 #define MAX_NO_OF_SLOTS 5

9.45.2.7 #define MAX_PATH_LENGTH 255

9.45.2.8 #define MAX_PUK_LENGTH 8

9.45.2.9 #define MAX_SLOTS_STATUS 255

9.45.3 Function Documentation

9.45.3.1 ULONG SLQSUIMAuthenticate (UIMAuthenticateReq * *pUIMAuthenticateReq*, UIMAuthenticateResp * *pUIMAuthenticateResp*)

This API executes the authentication algorithm on the card.

Parameters

pUIM-Authenticate-Req[IN]	• See UIMAuthenticateReq for more information.
pUIM-Authenticate-Resp[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
```

9.45.3.2 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.
```

9.45.3.3 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.
```

9.45.3.4 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.
```

9.45.3.5 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.
```

9.45.3.6 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

9.45.3.7 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

pUIMGetFile-AttributesReq[IN]	• See UIMGetFileAttributesReq for more information.
pUIMGetFile-AttributesResp[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.
```

9.45.3.8 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

9.45.3.9 ULONG SLQSUIMPowerDown (UIMPowerDownReq * *pUIMPowerDownReq*)

This API powers down the SIM card.

Parameters

pUIMPower-DownReq[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.
```

9.45.3.10 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.
```

9.45.3.11 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.
```

9.45.3.12 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
```

9.45.3.13 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 )
```

9.45.3.14 ULONG SLQSUIMRefreshOK (UIMRefreshOKReq * *pUIMRefreshOKReq*)

This API Enables the client to indicate whether it is OK to start the Refresh procedure.

Parameters

pUIMRefreshOKReq[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.
```

9.45.3.15 ULONG SLQSUIMRefreshRegister (UIMRefreshRegisterReq * *pUIMRefreshRegisterReq*)

This API Registers for file change notifications triggered by the card.

Parameters

pUIMRefreshRegisterReq[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.
```

9.45.3.16 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.
```

9.45.3.17 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.
```

9.45.3.18 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.
```

9.45.3.19 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.
```

9.45.3.20 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.
```

9.46 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)

9.46.1 Detailed Description

Voice Service API function prototypes.

9.46.2 Macro Definition Documentation

9.46.2.1 #define MAX_CALL_NO_LEN 81

9.46.2.2 #define MAX_DESCRIPTION_LENGTH 255

9.46.2.3 #define MAX_NO_OF_CALLS 20

9.46.2.4 #define MAXUSSDLENGTH 182

9.46.2.5 #define PASSWORD_LENGTH 4

9.46.3 Enumeration Type Documentation

9.46.3.1 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

9.46.4 Function Documentation

9.46.4.1 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

9.46.4.2 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

9.46.4.3 ULONG OriginateUSSD (BYTE * *pInfo*)

Initiates a USSD operation.

Parameters

pInfo[IN]	• USS information • 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

9.46.4.4 ULONG SLQSOriginateUSSD (struct USSInfo * *pReq*, struct USSResp * *pResp*)

Initiates a USSD session.

Parameters

pReq	[IN] • USS information • See USSInfo for more details
pResp	[OUT] • 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

9.46.4.5 ULONG SLQSVoiceALSSelectLine (voiceALSSelectLineInfo * *pVoiceALSSelectLineInfo*)

This API allows the user to select the preferred line.

Parameters

pVoiceALS-SelectLineInfo[I-N]	• 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.

9.46.4.6 ULONG SLQSVoiceALSSetLineSwitching (voiceALSSetLineSwitchInfo * *pVoiceALSSetLineSwitchInfo*)

This API sets the line switch setting on the card.

Parameters

pVoiceALSSet-LineSwitchInfo[I-N]	• 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.

9.46.4.7 ULONG SLQSVoiceAnswerCall (voiceAnswerCall * *pVoiceAnswerCall*)

Answers an incoming voice call.

Parameters

pVoiceAnswer-Call[IN/OUT]	• Pointer to structure of voiceAnswerCall – 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.

9.46.4.8 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.

9.46.4.9 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. • 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.

9.46.4.10 ULONG SLQSVoiceDialCall (voiceCallRequestParams * *pCallRequestParams*, voiceCallResponseParams * *pCallResponseParams*)

Originates a voice call (MO call).

Parameters

pCallRequest-Params[IN]	• Pointer to structure of voiceCallRequestParams – See voiceCallRequestParams for more information
pCallResponse-Params[OUT]	• Pointer to structure of voiceCallResponseParams – 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. SLQSVoiceSetAllCallStatus-Callback() callback can be subscribed to learn if the call was successful.

9.46.4.11 ULONG SLQSVoiceEndCall (BYTE * *pCallId*)

This message ends a voice call

Parameters

pCallId	[IN/OUT] • 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.

9.46.4.12 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.

9.46.4.13 ULONG SLQSVoiceGetCallBarring (voiceGetCallBarringReq * *pVoiceGetCallBarringReq*, voiceGetCallBarringResp * *pVoiceGetCallBarringResp*)

Queries the status of Call Barring Supplementary Service (applicable only for 3GPP).

Parameters

pVoiceGetCall-BarringReq[IN]	• Pointer to structure of voiceGetCallBarringReq – See voiceGetCallBarringReq for more information
pVoiceGetCall-BarringResp[O-UT]	• Pointer to structure of voiceGetCallBarringResp – 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.

9.46.4.14 ULONG SLQSVoiceGetCallForwardingStatus (voiceGetCallFWReq * *pVoiceGetCallFWReq*, voiceGetCallFWResp * *pVoiceGetCallFWResp*)

Queries the status of Call Forwarding Supplementary Service.

Parameters

pVoiceGetCallFWReq[IN]	• Pointer to structure of voiceGetCallFWReq – See voiceGetCallFWReq for more information
pVoiceGetCallFWResp[OUT]	• Pointer to structure of voiceGetCallFWResp – 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.

9.46.4.15 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

pGetCallInfo-Req[IN]	• See voiceCallInfoReq for more information.
pGetCallInfo-Resp[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.

9.46.4.16 ULONG SLQSVoiceGetCallWaiting (voiceGetCallWaitInfo * *pVoiceGetCallWaitInfo*)

Queries the status of Call Waiting Supplementary Service (applicable only for 3GPP).

Parameters

pVoiceGetCall-WaitInfo[IN/OUT]	• Pointer to structure of voiceGetCallWaitInfo – 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.

9.46.4.17 ULONG SLQSVoiceGetCLIP (voiceGetCLIPResp * *pVoiceGetCLIPResp*)

Queries the status of the Calling Line Identification Presentation (CLIP) supplementary service (applicable only for 3GPP).

Parameters

pVoiceGetCLIP-Resp[OUT]	• Pointer to structure of voiceGetCLIPResp – 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.

9.46.4.18 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 – 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.

9.46.4.19 ULONG SLQSVoiceGetCNAP (voiceGetCNAPResp * *pVoiceGetCNAPResp*)

Queries the status of the Calling Name Presentation(CNAP) supplementary service (applicable only for 3GPP).

Parameters

pVoiceGetCNAPResp[OUT]	• Pointer to structure of voiceGetCNAPResp – 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.

9.46.4.20 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 – 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.

9.46.4.21 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 – 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.

9.46.4.22 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. – See voiceGetConfigReq for more information.
pVoiceGetConfigResp	• Structure containing Get Config response parameters. – 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.

9.46.4.23 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. – 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.
```

9.46.4.24 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

pVoiceManageCallsReq[IN]	• Request structure of to manage calls.
pVoiceManageCallsResp[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"

9.46.4.25 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

pVoiceOrigUSSDNoWaitInfo[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.

9.46.4.26 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.

9.46.4.27 ULONG SLQSVoiceSetCallBarringPassword (voiceSetCallBarringPwdInfo * *pVoiceSetCallBarringPwdInfo*, voiceSetCallBarringPwdResp * *pSetCallBarringPwdResp*)

Sets a Call Barring Password (applicable only for 3GPP).

Parameters

pVoiceSetCallBarringPwdInfo[IN]	• Pointer to structure of voiceSetCallBarringPwdInfo – See voiceSetCallBarringPwdInfo for more information
pSetCallBarringPwdResp[OUT]	• Pointer to structure of voiceSetCallBarringPwdResp – 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.

9.46.4.28 ULONG SLQSVoiceSetConfig (voiceSetConfigReq * *pVoiceSetConfigReq*, voiceSetConfigResp * *pVoiceSetConfigResp*)

This message sets various configuration parameters that control the modem behavior related to circuit-switched services.

Parameters

pVoiceSetConfigReq[IN]	• Pointer to structure of voiceSetConfigReq – See voiceSetConfigReq for more information
pVoiceSetConfigResp[OUT]	• Pointer to structure of voiceSetConfigResp – 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.

9.46.4.29 ULONG SLQSVoiceSetPreferredPrivacy (voiceSetPrefPrivacy * *pSetPrefPrivacy*)

This API sets the voice privacy preference. Applicable only for 3GPP2 devices.

Parameters

pSetPrefPrivacy[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

9.46.4.30 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

pVoiceSetSUPSServiceReq	[IN] • Pointer to structure of voiceSetSUPSServiceReq – See voiceSetSUPSServiceReq for more information
pVoiceSetSUPSServiceResp	[OUT] • Pointer to structure of voiceSetSUPSServiceResp – 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

9.46.4.31 ULONG SLQSVoiceStartContDTMF (voiceContDTMFinfo * *pContDTMFInfo*)

Starts a continuous DTMF.

Parameters

pContDTMF-Info[IN/OUT]	• Structure containing Continuous DTMF Information. – 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.

9.46.4.32 ULONG SLQSVoiceStopContDTMF (voiceStopContDTMFinfo * *pVoiceStopContDTMFinfo*)

Stops a continuous DTMF.

Parameters

pVoiceStopCont-DTMFinfo	• Structure containing Continuous Stop DTMF Information. – See voiceStopContDTMFinfo for more information. • Start continuous DTMF request is sent to the current active/alerting call when CallId is set to 0xFF. • 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.

9.47 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)

9.47.1 Detailed Description

Wireless Data Service API function prototypes.

9.47.2 Macro Definition Documentation

9.47.2.1 #define IPV6_ADDRESS_ARRAY_SIZE 8

9.47.3 Typedef Documentation

9.47.3.1 typedef struct _GetProfileSettingIn GetProfileSettingIn

This structure contains the input parameters for SLQSGetProfileSettings

Parameters

ProfileType	• Identifies the technology type of the profile – 0x00 - 3GPP – 0x01 - 3GPP2
ProfileID	• index identifying the profile

9.47.3.2 typedef struct _GetProfileSettingOut GetProfileSettingOut

This structure contains the profile settings retrieved by the API SLQSGetProfileSettings

Parameters

curProfile	• Structure containing details of the profile • See QmiProfileInfo for more details
pExtErrCode	• pointer to a 2 byte extended error code • Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device. • See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description.

9.47.3.3 typedef union WdsProfileParam QmiProfileInfo

This union WdsProfileParam consist of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

9.47.3.4 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. – NULL if the parameter is not required
pLastCallData-Bearer-Technology[OU-T]	• last call data bearer technology value. – NULL if the parameter is not required

9.47.3.5 typedef struct dataBearerTechnology QmiWDSDataBearerTechnology

Structure to hold the current data bearer technology values

Parameters

pCurrent-Network[OUT]	• current selected network – 0 - UNKNOWN – 1 - 3GPP2 – 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) – 0x8000 - NULL Bearer – 0x0000 - DO_NOT_CARE CDMA RAT mask – 0x01 - CDMA_1X – 0x02 - EVDO_REV0 – 0x04 - EVDO_REVA UMTS RAT mask – 0x01 - WCDMA – 0x02 - GPRS – 0x04 - HSDPA – 0x08 - HSUPA – 0x10 - EDGE – 0x20 - LTE – 0x40 - HSDPA+ – 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) – 0x00 - DO_NOT_CARE CDMA 1X SO mask – 0x01 - CDMA_1X_IS95 – 0x02 - CDMA_1X_IS2000 – 0x04 - CDMA_1X_IS2000_REL_A CDMA EV-DO Rev A SO mask – 0x01 - EVDO_REVA_DPA – 0x02 - EVDO_REVA_MFPA – 0x04 - EVDO_REVA_EMPA – 0x08 - EVDO_REVA_EMPA_EHRPD

9.47.3.6 typedef struct _slqs3GPPConfigItem slqs3GPPConfigItem

This structure contains the 3gpp Configuration Item information.

Parameters

pLTEAttach-Profile	• LTE Attach Profile – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) – function SLQSGet3GPPConfigItem() returns a default value 255 if no LTE Attach Profile is configured • This setting is deprecated on MC/EM74xx
pProfileList	• Profile List – an array of 4 profile configurations – Each element points to a single WORD value indicating profile – Optional parameter with possible values * 1 - 16 (MC/EM73xx and before) * 1 - 24 (MC/EM74xx and onwards) – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
pDefaultPDN-Enabled	• Always Connect Default PDN – A single BYTE value indicating the status of Always connect default PDN * 0 - disabled * 1 - enabled – Optional parameter – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
p3gppRelease	• 3gpp release – A single BYTE value indicating the 3gpp release * 0 - Release 99 * 1 - Release 5 * 2 - Release 6 * 3 - Release 7 * 4 - Release 8 – In 9x30 and onwards * 5 - Release 9 * 6 - Release 10 * 7 - Release 11 – Optional parameter – function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present

pLTEAttach-ProfileList	• pointer to WORD array indicating LTE Attach Profile List – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – 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 – valid range: 1-24 * This setting is only supported for MC/EM74xx onwards

9.47.4 Enumeration Type Documentation

9.47.4.1 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

9.47.5 Function Documentation

9.47.5.1 ULONG GetAutoconnect (ULONG * *pSetting*)

Returns the current auto connect data session setting.

Parameters

pSetting[OUT]	• NDIS auto connect setting – 0 - Disabled – 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

9.47.5.2 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

9.47.5.3 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

9.47.5.4 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 – 0x01 - CDMA2000 1x – 0x02 - CDMA 1xEV-DO Rev 0 – 0x03 - GSM – 0x04 - UMTS – 0x05 - CDMA2000 HRPD (1xEV-DO Rev A) – 0x06 - EDGE – 0x07 - HSDPA AND WCDMA – 0x08 - WCDMA AND HSUPA – 0x09 - HSDPA AND HSUPA – 0x0A - LTE – 0x0B - CDMA2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 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

9.47.5.5 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 – 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 – 0 - PDP-IP (IPv4)
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 – 0x00000001 - PAP preference * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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. • 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

9.47.5.6 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 – 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 – 0 - PDP-IP (IPv4)
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>] IPv6 address: 1234:2A01:....:5678 U0 corresponds to 1234 U1 corresponds to 2A01 – ----------— – -— – ----------— – -— U7 corresponds to 5678
pPrimaryDN-Sv6[OUT]	• Primary DNS Ipv6 address preference
pSecondaryDN-Sv6[OUT]	• Secondary DNS Ipv6 address preference
pAuthentication[-OUT]	• Bitmap that indicates authentication algorithm preference – 0x00000001 - PAP preference * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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. • 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

9.47.5.7 ULONG GetDefaultProfileNum (BYTE *profile_type*, BYTE *profile_family*, BYTE * *pProfile_no*)

This API to Get default profile number

Parameters

profile_type	[IN] • 0 - 3GPP • 1 - 3GPP2
profile_family	[IN] • 0 - Embedded • 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
```

9.47.5.8 ULONG GetDormancyState (ULONG * *pDormancyState*, BYTE *instance*)

Returns the dormancy state of the current packet data session when connected.

Parameters

pDormancy-State[OUT]	• Dormancy state of current packet data session – 1 - Traffic channel dormant – 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

9.47.5.9 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.

9.47.5.10 ULONG GetLastMobileIPError (ULONG * *pError*)

Returns the last mobile IP error.

Parameters

pError[OUT]	• Status of last MIP call (or attempt) – Zero - Success – NonZero - Error code 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

9.47.5.11 ULONG GetMobileIP (ULONG * *pMode*)

Returns the current mobile IP setting.

Parameters

mode[OUT]	• Mobile IP setting – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 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

9.47.5.12 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: – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
pAddress[OUT]	• Home IPv4 address: – 0xFFFFFFFF - Unknown

pPrimaryHA[OUT]	• Primary home agent IPv4 address – 0xFFFFFFFF - Unknown
pSecondaryHA[OUT]	• Secondary home agent IPv4 address – 0xFFFFFFFF - Unknown
pRevTunneling[OUT]	• Reverse tunneling enabled – 0 - Disabled – 1 - Enabled – 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 – 0xFFFFFFFF - Unknown
pHAState[OUT]	• Home agent key state – 0 - Unset – 1 - Set, default value – 2 - Set, modified from default – 3 - 0xFFFFFFFF - Unknown
pAAAState[OUT]	• AAA key state – 0 - Unset – 1 - Set, default value – 2 - Set, modified from default – 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

9.47.5.13 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

9.47.5.14 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]	• No. of packets transmitted without error
pRXPacket-Successes[OUT]	• No. of packets received without error
pTXPacket-Errors[OUT]	• No. of outgoing packets with framing errors
pRXPacket-Errors[OUT]	• No. 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

9.47.5.15 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

9.47.5.16 ULONG GetSessionState (ULONG * *pState,* BYTE *instance*)

Returns the state of the current packet data session.

Parameters

pState[OUT]	• Current link status – 1 - DISCONNECTED – 2 - CONNECTED – 3 - SUSPENDED (not supported) – 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

9.47.5.17 ULONG iGetByteTotals (ULONG * *pv4sessionId,* ULONG * *pv6sessionId,* struct WdsByteTotalsElmnts * *pByteTotalsElmnt*)

9.47.5.18 ULONG iGetConnectionRate (ULONG * *pv4sessionId,* ULONG * *pv6sessionId,* struct WdsConnectionRateElmnts * *pConnectionRateElmnt*)

9.47.5.19 ULONG iGetPacketStatistics (ULONG * *pV4sessionId,* ULONG * *pV6sessionId,* struct WdsPktStatisticsReq * *pStatMask,* struct WdsPktStatisticsElmnts * *pPktStatisticsElmnt*)

9.47.5.20 ULONG iSLQSMISetIPFamilyPreference (BYTE *IPFamilyPreference,* BYTE *instance*)

9.47.5.21 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
```

9.47.5.22 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

9.47.5.23 ULONG SetAutoconnect (ULONG *setting*)

Sets the auto connect data session setting.

Parameters

setting[IN]	• NDIS autoconnect setting – 0 - Disabled – 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

9.47.5.24 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 – 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) – 0 - PDP-IP (IPv4)

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) – 0x00000001 - PAP preference * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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) • 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.

9.47.5.25 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 – 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) – 0 - PDP-IP (IPv4)
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) – 0x00000001 - PAP preference * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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) • 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

9.47.5.26 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 – 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) – 0 - PDP-IP (IPv4)
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) – 0x00000001 - PAP preference * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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) • 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

9.47.5.27 ULONG SetDefaultProfileNum (BYTE *profile_type,* BYTE *profile_family,* BYTE *profile_index*)

This API to Set default profile number

Parameters

profile_type	[IN] • 0 - 3GPP • 1 - 3GPP2
profile_family	[IN] • 0 - Embedded • 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
```

9.47.5.28 ULONG SetMobileIP (ULONG *mode*)

Sets the current mobile IP setting.

Parameters

mode[IN]	• Mobile IP setting – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 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

9.47.5.29 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) – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 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) – Zero - Disabled – NonZero - Enabled
pHA-Authenticator[IN]	• MH-HA authenticator calculator (optional) – Zero - Disabled – NonZero - Enabled
pHA2002bis[IN]	• MH-HA RFC 2002bis authentication instead of RFC2002 (optional) – Zero - Disabled – 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

9.47.5.30 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

9.47.5.31 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

9.47.5.32 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

9.47.5.33 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. • See SLQSDeleteProfileParams for more details.
pExtendedErrorCode[OUT]	• The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_REG_xxx. • Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device. • 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

9.47.5.34 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

9.47.5.35 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
```

9.47.5.36 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

9.47.5.37 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
```

9.47.5.38 ULONG SLQSGetCurrentChannelRate (WDSSWICurrentChannelRates * *pRates,* BYTE *instance*)

This API Queries current bitrate of a packet data connection.

Parameters

pRates	[IN] • 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 SLQSSetCust-Features

Timeout: 2 seconds

9.47.5.39 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

qmiWDSData-Bearers[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

9.47.5.40 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] • See DataBearerTechExt for more information
instance	[IN] • 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
```

9.47.5.41 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

9.47.5.42 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
```

9.47.5.43 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 – 0 - UMTS
profileId	• Index of the configured profile for which settings are read – Value between 1 - 16 (EM/MC73xx or earlier) – 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 – 0 - PDP-IP (IPv4)
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 – 0x00000001 - PAP preference * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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. • 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. • Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device. • 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

9.47.5.44 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 • See GetProfileSettingIn for more information
pResp[OUT]	• The profile settings and/or extended error code returned by the device based on input parameters. • 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

9.47.5.45 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

9.47.5.46 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 – 1 - DISCONNECTED – 2 - CONNECTED – 3 - SUSPENDED (not supported) – 4 - AUTHENTICATING
pStateV6[OUT]	• Current link status for IPV6 Session – 1 - DISCONNECTED – 2 - CONNECTED – 3 - SUSPENDED (not supported) – 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

9.47.5.47 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

9.47.5.48 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
```

9.47.5.49 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

9.47.5.50 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 – 0 - UMTS

profileId	• Profile number to be modified – Value between 1 - 16 (EM/MC73xx or earlier) – 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) – 0 - PDP-IP (IPv4)
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) – 0x00000001 - PAP preference * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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) • 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.

9.47.5.51 ULONG SLQSSGetDHCPv4ClientConfig (WdsDHCPv4Config * *pReqResp*)

This API gets the DHCP Client V4 Configuration.

Parameters

pReq	[IN] • 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
```

9.47.5.52 ULONG SLQSSGetLoopback (WDSGetLoopbackData * *data*)

This API to Get the value of loopback mode and multiplier.

Parameters

pReq	[IN] • 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
```

9.47.5.53 ULONG SLQSSSetDHCPv4ClientConfig (WdsDHCPv4Config * *pReq*)

This API sets the DHCP Client V4 Configuration.

Parameters

pReq	[IN] • 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
```

9.47.5.54 ULONG SLQSSSetLoopback (WDSSetLoopbackData * *pReq*)

This API to Enable/disable Data Loopback Mode and set the value of loopback multiplier.

Parameters

pReq	[IN] • 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
```

9.47.5.55 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 • 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

9.47.5.56 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

9.47.5.57 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

9.47.5.58 ULONG SLQSWdsSetEventReport (wdsSetEventReportReq * *pSetEventReportReq*)

This API sets the wireless data connection state reporting conditions for the requesting control point.

Parameters

pSetEvent-ReportReq[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

9.47.5.59 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

9.47.5.60 BOOL WDS_IsGobiDevice ( )

9.48 qaNasGetRFBandInfo.h File Reference

Data Structures

- struct QmiNasGetRFBandInfoResp

Enumerations

- enum eQMI_NAS_GET_RF_INFO_RESP { eTLV_RF_BAND_INFO = 0x01 }

Functions

- enum eQCWWANError PkQmiNasGetRFBandInfo (WORD *pMlength, BYTE *pBuffer)
- enum eQCWWANError UpkQmiNasGetRFBandInfo (BYTE *pMdmResp, struct QmiNasGetRFBandInfo-Resp *pApiResp)

9.48.1 Enumeration Type Documentation

9.48.1.1 enum eQMI_NAS_GET_RF_INFO_RESP

Enumerator

eTLV_RF_BAND_INFO

9.48.2 Function Documentation

9.48.2.1 enum eQCWWANError PkQmiNasGetRFBandInfo (WORD * *pMlength,* BYTE * *pBuffer*)

9.48.2.2 enum eQCWWANError UpkQmiNasGetRFBandInfo (BYTE * *pMdmResp,* struct QmiNasGetRFBandInfoResp * *pApiResp*)

9.49 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 QmiNasPerform-NetworkScanResp *pAipResp)

9.49.1 Macro Definition Documentation

9.49.1.1 #define FORBIDDEN_INDEX 4

9.49.1.2 #define INDEX_ZERO 0

9.49.1.3 #define MAX_DESCRIPTION_LENGTH 255

9.49.1.4 #define PREFERRED_INDEX 6

9.49.1.5 #define QMI_NAS_MAX_INSTANCES 20

9.49.1.6 #define QMI_NAS_NETSTATUS_MASK 0x03

9.49.1.7 #define ROAMING_INDEX 2

9.49.2 Enumeration Type Documentation

9.49.2.1 enum eQMI_NAS_PERFORM_NETWORK_SCAN_RESP

Enumerator

eTLV_3GPP_NETWORK_INFO

9.49.3 Function Documentation

9.49.3.1 enum eQCWWANError PkQmiNasPerformNetworkScan (WORD * *pMlength*, BYTE * *pParamField*)

9.49.3.2 enum eQCWWANError UpkQmiNasPerformNetworkScan (BYTE * *pMdmResp*, struct QmiNasPerformNetworkScanResp * *pAipResp*)

9.50 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 }

9.50.1 Enumeration Type Documentation

9.50.1.1 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 - Device GPS service disabled

eQCWWAN_ERR_QMI_INVALID_OPERATION 1070 - Invalid operation specified

eQCWWAN_ERR_QMI_INVALID_QMI_CMD 1071 - Invalid/unknown QMI command specified

eQCWWAN_ERR_QMI_TPDU_TYPE 1072 - Message contains TPDU type that cannot be read as raw message

eQCWWAN_ERR_QMI_SMSC_ADDR 1073 - The SMSC address specified is invalid

eQCWWAN_ERR_QMI_INFO_UNAVAILABLE 1074 - Information element is unavailable at this point

eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG 1075 - Segment size too large

eQCWWAN_ERR_QMI_SEGMENT_ORDER 1076 - Segment order is incorrect

eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED 1077 - Bundling not supported

eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE 1078 - The operation failed partially

eQCWWAN_ERR_QMI_POLICY_MISMATCH 1079 - Policy mismatch

eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND 1080 - SIM file not found

eQCWWAN_ERR_QMI_EXTENDED_INTERNAL 1081 - Extended internal error

eQCWWAN_ERR_QMI_ACCESS_DENIED 1082 - Access to a required entity is not available

eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED 1083 - Selected operating mode is invalid with current hardware setting

eQCWWAN_ERR_QMI_ACK_NOT_SENT 1084 - ACK not sent

eQCWWAN_ERR_QMI_INJECT_TIMEOUT 1084 - Inject a timeout for the request

eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE 1090 - Incompatible state

eQCWWAN_ERR_QMI_FDN_RESTRICT 1091 - FDN Restrict

eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE 1092 - SUPS failure cause

eQCWWAN_ERR_QMI_NO_RADIO 1093 - No Radio

eQCWWAN_ERR_QMI_NOT_SUPPORTED 1094 - Not Supported

eQCWWAN_ERR_QMI_NO_SUBSCRIPTION 1095 - No Subscription

eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED 1096 - Card call control failed

eQCWWAN_ERR_QMI_NETWORK_ABORTED 1097 - Network Aborted

eQCWWAN_ERR_QMI_MSG_BLOCKED 1098 - Open Error

eQCWWAN_ERR_QMI_MAX Error - End of QMI specific defines

eQCWWAN_ERR_SWICM_START **Vendor defines - Connection Manager error codes**

eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED 0xE001 - The API is yet to be implemented

eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED 0xE002 - The service is not supported

eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED 0xE003 - The client is not supported

eQCWWAN_ERR_SWICM_TIMEOUT 0xE004 - API Timeout

eQCWWAN_ERR_SWICM_SOCKET_IN_USE 0xE005 - The communication socket is in use

eQCWWAN_ERR_SWICM_AM_VERS_ERROR 0xE006 - SLQS API and SDK version mismatch

eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS 0xE007 - Failed to kill SDK process

eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS 0xE008 - Call in progress

eQCWWAN_ERR_SWICM_V4DWN_V6DWN 0xE009 - IPV4 and IPV6 is down

eQCWWAN_ERR_SWICM_V4DWN_V6UP 0xE00A - IPV4 is down and IPV6 is up

eQCWWAN_ERR_SWICM_V4UP_V6DWN 0xE00B - IPV4 is up and IPV6 is down

eQCWWAN_ERR_SWICM_V4UP_V6UP 0xE00C - IPV4 and IPV6 is up

eQCWWAN_ERR_SWICM_INVALID_SESSION_ID 0xE00D - Invalid V4 Session ID

eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID 0xE00E - Invalid V4 Session ID

eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID 0xE00F - Invalid V6 Session ID

eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS 0xE010 - No available Session Manager slots for additional data sessions

eQCWWAN_ERR_SWICM_END 0xE011 - End of connection manager specific codes

eQCWWAN_ERR_SWISMS_START **Vendor defines - SMS Error codes**

eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG 0xE101 - SMS message length is long

eQCWWAN_ERR_SWISMS_MSG_CORRUPTED 0xE102 - The SMS message is corrupted (encoding wrong)

eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED 0xE103 - The SMS number is corrupted (incorrect number)

eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND 0xE104 - The SMS bearer data is not available

eQCWWAN_ERR_SWISM_END

eQCWWAN_ERR_SWIIM_START **Vendor defines - Image Management error codes**

eQCWWAN_ERR_SWIIM_INVALID_PATH 0xE801 - Invalid directory path

eQCWWAN_ERR_SWIIM_OPENING_DIR 0xE802 - Unable to open the directory

eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND 0xE803 - No Firmware image present in the path

eQCWWAN_ERR_SWIIM_OPENING_FILE 0xE804 - Unable to open the file

eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE 0xE805 - Firmware image is corrupted

eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED 0xE806 - No Firmware image download needed

eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL 0xE807 - Firmware update failed

eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH 0xE808 - Update success but pri/fw preference mismatch

eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS 0xE809 - Update successful

eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE 0xE80A - Enter Download Mode

eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE 0xE80B - File transfer to modem complete

eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT 0xE80C - Wait for modem to reboot

eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE 0xE80D - Invalid Crash State for Firmware Download

eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE 0xE80E - Same as current active image

eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX 0xE80F - invalid slot index

eQCWWAN_ERR_SWIIM_END

eQCWWAN_ERR_SWIDCS_START **Vendor defines - Device Connectivity error codes**

eQCWWAN_ERR_SWIDCS_IOCTL_ERR 0xE901 - IO Control error

eQCWWAN_ERR_SWIDCS_FILEIO_ERR 0xE902 - file open/read/write error

eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND 0xE903 - The device is not found

eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED 0xE904 - Application is disconnected from SDK

eQCWWAN_ERR_SWIDCS_END

eQCWWAN_ERR_QMI_CAT_START **QMI errors related to CAT**

eQCWWAN_ERR_QMI_EVENT_REG_FAILED 62441 - CAT event registration failed

eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP 62442 - Invalid terminal response

eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD 62443 - Invalid envelope command

eQCWWAN_ERR_QMI_CARD_BUSY_RSP 62444 - Card busy response for envelope command

eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE 62445 - Envelope command failure

eQCWWAN_ERR_QMI_CAT_END

eQCWWAN_ERR_NULL_TLV

eQCWWAN_ERR_QMI_WIDTH 0xFFFF - Not an error, represent the end of QMI errors

9.50.1.2 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

9.51 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

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)

9.51.1 Macro Definition Documentation

9.51.1.1 #define LITEQMI_MAX_QOS_FILTERS 25

9.51.1.2 #define LITEQMI_MAX_QOS_FLOW_PER_APN_STATS 10

9.51.1.3 #define LITEQMI_MAX_QOS_FLOWS 8

9.51.2 Function Documentation

9.51.2.1 int pack_qos_SLQSQosGetNetworkStatus (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen*)

Function to pack command to retrieve QoS status of the network. This maps to SLQSQosGetNetworkStatus

Parameters

pCtx[OUT]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • 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

9.51.2.2 int pack_qos_SLQSQosSwiReadApnExtraParams (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen,* pack_qos_SLQSQosSwiReadApnExtraParams_t *reqParam*)

Function to pack QMI command to query extra APN parameters This maps to SLQSQosSwiReadApnExtraParams

Parameters

pCtx[OUT]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes

pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.51.2.3 int pack_qos_SLQSQosSwiReadDataStats (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_qos_SLQSQosSwiReadDataStats_t *reqParam*)

Function to pack QMI command to query APN data statistics This maps to SLQSQosSwiReadDataStats

Parameters

pCtx[OUT]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.51.2.4 int pack_qos_SLQSSetQosEventCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_qos_SLQSSetQosEventCallback_t *reqParam*)

Function to pack QMI command to enable QoS event indications This maps to SLQSSetQosEventCallback

Parameters

pCtx[OUT]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.51.2.5 int unpack_qos_SLQSQosGetNetworkStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSQosGetNetworkStatus_t * *pOutput*)

Function to unpack the response to get NW QoS status command This maps to SLQSQosGetNetworkStatus

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.51.2.6 int unpack_qos_SLQSQosSwiReadApnExtraParams (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSQosSwiReadApnExtraParams_t * *pOutput*)

Function to unpack the response to get NW QoS status command This maps to SLQSQosSwiReadApnExtraParams

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.51.2.7 int unpack_qos_SLQSQosSwiReadDataStats (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSQosSwiReadDataStats_t * *pOutput*)

Function to unpack APN data statistics response This maps to SLQSQosSwiReadDataStats

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.51.2.8 int unpack_qos_SLQSSetQosEventCallback (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack enable QoS event indications command's response This maps to SLQSSetQosEventCallback

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• 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

9.51.2.9 int unpack_qos_SLQSSetQosEventCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosEventCallback_ind_t * *pOutput*)

Function to unpack QoS event indications This maps to SLQSSetQosEventCallback

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.51.2.10 int unpack_qos_SLQSSetQosNWStatusCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosNWStatusCallback_ind_t * *pOutput*)

Function to unpack QoS NW status indication. This maps to SLQSSetQosNWStatusCallback

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.51.2.11 int unpack_qos_SLQSSetQosPriEventCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosPriEventCallback_ind_t * *pOutput*)

Function to unpack QoS primary flow events. This maps to SLQSSetQosPriEventCallback

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.51.2.12 int unpack_qos_SLQSSetQosStatusCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosStatusCallback_ind_t * *pOutput*)

Function to unpack QoS status indications. This maps to SLQSSetQosStatusCallback

Parameters

pResp[IN]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.52 rms.h File Reference

Data Structures

- struct unpack_rms_GetSMSWake_t
- struct pack_rms_SetSMSWake_t

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)

9.52.1 Function Documentation

9.52.1.1 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	*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

9.52.1.2 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.52.1.3 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

9.52.1.4 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

9.53 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)

9.53.1 Function Documentation

9.53.1.1 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	*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

9.53.1.2 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.53.1.3 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

9.53.1.4 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

9.54 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 sMSMTMessage
- struct newMTMessageTlv
- struct sMSTransferRouteMTMessage
- struct transferRouteMessageTlv
- struct sMSMessageMode
- struct messageModeTlv
- struct sMSEtwsMessage
- struct sMSEtwsMessageTlv
- struct sMSEtwsPlmn
- struct eTWSPLMNInfoTlv
- struct sMSCAddress
- struct sMSCAddressTlv
- struct sMSOnIMS
- struct sMSOnIMSTlv
- struct unpack_sms_SetNewSMSCallback_ind_t
- struct unpack_sms_SLQSWmsMemoryFullCallBack_ind_t
- 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

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

Typedefs

- typedef struct sMSMTMessage sMSMTMessageInfo
- typedef struct sMSTransferRouteMTMessage sMSTransferRouteMTMessageInfo
- typedef struct sMSMessageMode sMSMessageModeInfo
- typedef struct sMSEtwsMessage sMSEtwsMessageInfo
- typedef struct sMSEtwsPlmn sMSEtwsPlmnInfo
- typedef struct sMSCAddress sMSCAddressInfo
- typedef struct sMSOnIMS sMSOnIMSInfo

Enumerations

- enum eqmiCbkSetStatus {
  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)

9.54.1 Macro Definition Documentation

9.54.1.1 #define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255

9.54.1.2 #define MAX_MS_TRANSFER_ROUTE_MSG 256

9.54.1.3 #define MAX_MSC_ADDRESS_SIZE 256

9.54.1.4 #define MAX_MSE_TWS_MSG 1254

9.54.1.5 #define MAX_SMS_LIST_SIZE 255

9.54.1.6 #define MAX_SMS_MESSAGE_SIZE 2048

9.54.1.7 #define SMS_CONFIG_LEN 0x05

9.54.1.8 #define SMS_MAX_SMS_ROUTES 0x0A

9.54.1.9 #define SMS_NUM_OF_SET 0xFF

9.54.1.10 #define SMSC_TYPE_LEN 0x03

9.54.2 Typedef Documentation

9.54.2.1 typedef struct sMSCAddress sMSCAddressInfo

Parameters

length	• Number of sets of following element
data	• SMSC address

9.54.2.2 typedef struct sMSEtwsMessage sMSEtwsMessageInfo

Parameters

notificationType	• Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS
length	• Number of sets of following elements
data	• Raw message data

9.54.2.3 typedef struct sMSEtwsPlmn sMSEtwsPlmnInfo

Parameters

mobileCountry-Code	• 16 bit representation of MCC value range : 0 -999
mobileNetwork-Code	• 16 bit representation of MNC value range : 0 -999

9.54.2.4 typedef struct sMSMessageMode sMSMessageModeInfo

Parameters

messageMode	Message Mode

9.54.2.5 typedef struct sMSMTMessage sMSMTMessageInfo

Parameters

storageType	memory storage 0x00-UIM 0x01-NV
messageIndex	MT Message index

9.54.2.6 typedef struct sMSOnIMS sMSOnIMSInfo

Parameters

smsOnIMS	SMS on IMS

9.54.2.7 typedef struct sMSTransferRouteMTMessage sMSTransferRouteMTMessageInfo

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

9.54.3 Enumeration Type Documentation

9.54.3.1 enum eqmiCbkSetStatus

Enumerator

LITEQMI_QMI_CBK_PARAM_RESET

LITEQMI_QMI_CBK_PARAM_SET

LITEQMI_QMI_CBK_PARAM_NOCHANGE

9.54.4 Function Documentation

9.54.4.1 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

9.54.4.2 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

9.54.4.3 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

9.54.4.4 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

9.54.4.5 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

9.54.4.6 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

9.54.4.7 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

9.54.4.8 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

9.54.4.9 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

9.54.4.10 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

9.54.4.11 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

9.54.4.12 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

9.54.4.13 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

9.54.4.14 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

9.54.4.15 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

9.54.4.16 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

9.54.4.17 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

9.54.4.18 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

9.54.4.19 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

9.54.4.20 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

9.54.4.21 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

9.54.4.22 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

9.54.4.23 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

9.54.4.24 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.25 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.26 int unpack_sms_SendSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SendSMS_t * *pOutput*)

send sms unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.27 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	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.28 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.29 int unpack_sms_SetSMSCAddress (uint8_t * *pResp*, uint16_t *respLen*)

Set the SMS center address unpack.

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.30 int unpack_sms_SLQSDeleteSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSDeleteSMS_t * *pOutput*)

delete sms unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.31 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.32 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.33 int unpack_sms_SLQSGetSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSMS_t * *pOutput*)

get sms unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.34 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.35 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.36 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.37 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.38 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	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.39 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	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.40 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	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.41 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	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.42 int unpack_sms_SLQSSetSmsBroadcastConfig (uint8_t * *pResp*, uint16_t *respLen*)

Sets the information about the SMS BroadcastConfiguration unpack.

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.43 int unpack_sms_SLQSSetSmsStorage (uint8_t * *pResp*, uint16_t *respLen*)

Sets the SMS Storage on the device unpack.

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.44 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.45 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.46 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	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.47 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.4.48 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55 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

9.55.1 Detailed Description

SWI data types.

9.55.2 Macro Definition Documentation

9.55.2.1 #define QMI_NO_LTE_FW_SUPPORT 0

9.55.2.2 #define QMI_TLV_PLACEHOLDER 0x8F

9.55.2.3 #define SWI_API

9.55.2.4 #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.

9.55.3 Typedef Documentation

9.55.3.1 typedef int BOOL

9.55.3.2 typedef unsigned char BYTE

9.55.3.3 typedef char CHAR

9.55.3.4 typedef float FLOAT

9.55.3.5 typedef signed int INT32

9.55.3.6 typedef signed char INT8

9.55.3.7 typedef const char* LPCSTR

9.55.3.8 typedef signed short SHORT

9.55.3.9 typedef unsigned int ULONG

9.55.3.10 typedef unsigned long long ULONGLONG

9.55.3.11 typedef unsigned short USHORT

9.55.3.12 typedef unsigned short WORD

9.56 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)

9.56.1 Function Documentation

9.56.1.1 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

9.56.1.2 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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.56.1.3 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	*pOuput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.56.1.4 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

9.57 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)

9.57.1 Macro Definition Documentation

9.57.1.1 #define LITEQMI_MAX_SWIOMA_STR_LEN 255

9.57.2 Function Documentation

9.57.2.1 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]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • 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

9.57.2.2 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]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.57.2.3 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]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes

pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.57.2.4 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]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • 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

9.57.2.5 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]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.57.2.6 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]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.57.2.7 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]	• See pack_qmi_t for more information
pReqBuf[IN/OUT]	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
pLen[IN/OUT]	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
reqParam[IN]	• 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

9.57.2.8 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]	• Response from modem
respLen[IN]	• 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

9.57.2.9 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]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.57.2.10 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]	• Response from modem
respLen[IN]	• 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

9.57.2.11 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]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.57.2.12 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]	• Response from modem
respLen[IN]	• Length of pResp from modem

pOutput[OUT]	• 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

9.57.2.13 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]	• Response from modem
respLen[IN]	• 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

9.57.2.14 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]	• Response from modem
respLen[IN]	• 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

9.57.2.15 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]	• Response from modem
respLen[IN]	• Length of pResp from modem
pOutput[OUT]	• 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

9.58 SWIWWANCMAPI.h File Reference

9.59 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

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)

9.59.1 Macro Definition Documentation

9.59.1.1 #define MAX_DESCRIPTION_LENGTH 255

9.59.1.2 #define MAX_ICCID_LENGTH 255

9.59.1.3 #define MAX_NO_OF_APPLICATIONS 10

9.59.1.4 #define MAX_NO_OF_SLOTS 5

9.59.1.5 #define MAX_SLOTS_STATUS 255

9.59.1.6 #define UIM_MAX_ACTIVE_PERS_FEATURES 12

9.59.1.7 #define UIM_MAX_CONTENT_LENGTH 1024

9.59.1.8 #define UIM_MAX_DESCRIPTION_LENGTH 255

9.59.1.9 #define UIM_MAX_NO_OF_APPLICATIONS 10

9.59.1.10 #define UIM_MAX_NO_OF_SLOTS 5

9.59.1.11 #define UIM_UINT8_MAX_STRING_SZ 255

9.59.2 Function Documentation

9.59.2.1 int pack_uim_ChangePin (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_ChangePin_t * *reqArg*)

Change Pin pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.59.2.2 int pack_uim_GetCardStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Card Status pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.59.2.3 int pack_uim_ReadTransparent (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_ReadTransparent_t * *reqArg*)

SLQS ReadTransparent pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.59.2.4 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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.59.2.5 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	*pReq*	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

9.59.2.6 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	*pReq*	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

9.59.2.7 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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.59.2.8 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	*pReq*	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

9.59.2.9 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	*pReq*	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

9.59.2.10 int pack_uim_SLQSUIMGetSlotsStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.11 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	*pReq*	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

9.59.2.12 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	*pReq*	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

9.59.2.13 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	*pReq*	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

9.59.2.14 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	*pReq*	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

9.59.2.15 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	*pReq*	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

9.59.2.16 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	*pReq*	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

9.59.2.17 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	*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

9.59.2.18 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	*pReq*	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

9.59.2.19 int pack_uim_UnblockPin (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_UnblockPin_t * *reqArg*)

Unblock Pin pack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.59.2.20 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	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response
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

9.59.2.21 int unpack_uim_ChangePin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_ChangePin_t * *pOutput*)

Change Pin unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.22 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.23 int unpack_uim_ReadTransparent (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_ReadTransparent_t * *pOutput*)

SLQS ReadTransparent unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.24 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.25 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	*pespLen*	qmi response length
out	*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

9.59.2.26 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.27 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.28 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.29 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.30 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.31 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

9.59.2.32 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

9.59.2.33 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

9.59.2.34 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

9.59.2.35 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	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.36 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

9.59.2.37 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

9.59.2.38 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	*pespLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.39 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	*pespLen*	qmi response length
out	*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

9.59.2.40 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

9.59.2.41 int unpack_uim_UnblockPin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_UnblockPin_t * *pOutput*)

Unblock Pin unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.59.2.42 int unpack_uim_VerifyPin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_VerifyPin_t * *pOutput*)

Verify Pin unpack

Parameters

in	*pResp*	qmi response
in	*pespLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60 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

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

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)

9.60.1 Macro Definition Documentation

9.60.1.1 #define BARRING_PASSWORD_LENGTH 4

9.60.1.2 #define MAX_VOICE_CALL_NO_LEN 81

9.60.1.3 #define MAX_VOICE_DESCRIPTION_LENGTH 255

9.60.1.4 #define MAXVOICEUSSDLENGTH 182

9.60.1.5 #define VOICE_MAX_NO_OF_CALLS 20

9.60.2 Function Documentation

9.60.2.1 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.2 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

9.60.2.3 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.4 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.5 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.6 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.7 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	*pReq*	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

9.60.2.8 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.9 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	*pReq*	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

9.60.2.10 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	*pReq*	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

9.60.2.11 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	*pReq*	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

9.60.2.12 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	*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

9.60.2.13 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	*pReq*	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

9.60.2.14 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	*pReq*	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

9.60.2.15 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	*pReq*	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

9.60.2.16 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	*pReq*	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

9.60.2.17 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	*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

9.60.2.18 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	*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

9.60.2.19 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	*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

9.60.2.20 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	*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

9.60.2.21 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	*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

9.60.2.22 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.23 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.24 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	*pReq*	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

9.60.2.25 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.26 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	*pReq*	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

9.60.2.27 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	*pReq*	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

9.60.2.28 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	*pReq*	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

9.60.2.29 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	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.30 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	*pReq*	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

9.60.2.31 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	*pReq*	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

9.60.2.32 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	*pReq*	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

9.60.2.33 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

9.60.2.34 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

9.60.2.35 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

9.60.2.36 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

9.60.2.37 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

9.60.2.38 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

9.60.2.39 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

9.60.2.40 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

9.60.2.41 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

9.60.2.42 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

9.60.2.43 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

9.60.2.44 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

9.60.2.45 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

9.60.2.46 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

9.60.2.47 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

9.60.2.48 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

9.60.2.49 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

9.60.2.50 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

9.60.2.51 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

9.60.2.52 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

9.60.2.53 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

9.60.2.54 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

9.60.2.55 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

9.60.2.56 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

9.60.2.57 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

9.60.2.58 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

9.60.2.59 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

9.60.2.60 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

9.60.2.61 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

9.60.2.62 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

9.60.2.63 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

9.60.2.64 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

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61 wds.h File Reference

Data Structures

- struct LibPackQosClassID
- struct LibPackTFTIDParams
- struct LibPackGPRSRequestedQoS
- struct LibPackUMTSQoS
- struct LibPackUMTSReqQoSSigInd
- struct pack_wds_SLQSStartDataSession_t
- struct unpack_wds_SLQSStartDataSession_t
- struct unpack_wds_SLQSSetPacketSrvStatusCallback_t
- struct pack_wds_SLQSStopDataSession_t
- struct wds_ProfileIdentifier
- struct wds_GPRSQoS
- struct wds_PCSCFIPv4ServerAddressList
- struct wds_PCSCFFQDNAddress
- struct wds_PCSCFFQDNAddressList
- struct wds_Domain
- struct wds_DomainNameList
- struct wds_IPV6AddressInfo
- struct wds_IPV6GWAddressInfo
- struct unpack_wds_SLQSGetRuntimeSettings_t
- struct wds_currNetworkInfo
- struct unpack_wds_SLQSSetWdsEventCallback_ind_t
- struct pack_wds_SLQSSetWdsEventCallback_t
- struct pack_wds_SLQSGetRuntimeSettings_t

- struct wds_UMTSMinQoS
- struct LibPackprofile_3GPP
- struct LibPackprofile_3GPP2
- union wds_profileInfo
- struct pack_wds_SLQSCreateProfile_t
- struct PackCreateProfileOut
- struct unpack_wds_SLQSCreateProfile_t
- struct pack_wds_SLQSModifyProfile_t
- struct unpack_wds_SLQSModifyProfile_t
- struct pack_wds_SLQSGetProfileSettings_t
- struct LibpackProfile3GPP
- struct LibpackProfile3GPP2
- union unpackWdsProfileParam
- struct UnPackGetProfileSettingOut
- struct unpack_wds_SLQSGetProfileSettings_t
- struct unpack_wds_GetSessionState_t
- struct pack_wds_GetDefaultProfile_t
- struct unpack_wds_GetDefaultProfile_t
- struct unpack_wds_GetConnectionRate_t
- struct pack_wds_GetPacketStatus_t
- struct unpack_wds_GetPacketStatus_t
- struct unpack_wds_GetSessionDuration_t
- struct pack_wds_GetSessionDuration_t
- struct unpack_wds_GetDormancyState_t
- struct pack_wds_GetDormancyState_t
- struct pack_wds_SLQSDeleteProfile_t
- struct unpack_wds_SLQSDeleteProfile_t
- struct pack_wds_SetDefaultProfile_t
- struct unpack_wds_SLQSGet3GPPConfigItem_t
- struct pack_wds_SLQSSet3GPPConfigItem_t
- struct unpack_wds_GetMobileIP_t
- struct pack_wds_GetMobileIP_t
- struct pack_wds_GetMobileIPProfile_t
- struct unpack_wds_GetMobileIPProfile_t
- struct currNetworkInfo
- struct unpack_wds_SLQSGetCurrDataSystemStat_t
- struct pack_wds_SLQSGetCurrDataSystemStat_t
- struct unpack_wds_GetLastMobileIPError_t
- struct pack_wds_GetLastMobileIPError_t
- struct rmTrasnferStaticsReq
- struct pack_wds_RMSetTransferStatistics_t
- struct unpack_wds_RMSetTransferStatistics_t
- struct pack_wds_SetMobileIPProfile_t
- struct unpack_wds_SetMobileIPProfile_t
- struct pack_wds_SLQSWdsSwiPDPRuntimeSettings_t
- struct ipv6AddressInfo
- struct unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t
- struct transferStatInd
- struct pack_wds_SLQSGetDUNCallInfo_t
- struct connectionStatus
- struct dunchannelRate
- struct unpack_wds_SLQSGetDUNCallInfo_t
- struct qmiWDSDataBearerTechnology
- struct unpack_wds_SLQSGetDataBearerTechnology_t
- struct pack_wds_SLQSGetDataBearerTechnology_t

- struct pack_wds_SLQSSetIPFamilyPreference_t
- struct unpack_wds_SLQSSetIPFamilyPreference_t
- struct pack_wds_SetDefaultProfileNum_t
- struct pack_wds_GetDefaultProfileNum_t
- struct unpack_wds_GetDefaultProfileNum_t
- struct wdsDhcpv4ProfileId
- struct wdsDhcpv4HwConfig
- struct wdsDhcpv4Option
- struct wdsDhcpv4OptionList
- struct pack_wds_SLQSSGetDHCPv4ClientConfig_t
- struct unpack_wds_SLQSSGetDHCPv4ClientConfig_t
- struct pack_wds_GetPacketStatistics_t
- struct unpack_wds_GetPacketStatistics_t
- struct unpack_wds_GetByteTotals_t
- struct unpack_wds_SLQSGetCurrentChannelRate_t
- struct unpack_wds_SLQSSGetLoopback_t
- struct pack_wds_SLQSSSetLoopback_t
- struct wds_DataULongTlv
- struct wds_DataULongLongTlv
- struct unpack_RMTransferStatistics_ind_t
- struct pack_wds_DHCPv4ClientLeaseChange_t
- struct wds_DHCPProfileIdTlv
- struct wds_DHCPLeaseStateTlv
- struct wds_IPv4AdTlv
- struct wds_DHCPOpt
- struct wds_DHCPLeaseOptTlv
- struct unpack_wds_DHCPv4ClientLease_ind_t
- struct pack_wds_SetMobileIP_t
- struct pack_wds_SetMobileIPParameters_t
- struct pack_wds_SetAutoconnect_t
- struct unpack_wds_GetAutoconnect_t
- struct wds_TrStatInd
- struct pack_wds_SLQSWdsSetEventReport_t
- struct wds_DHCPv4ProfileId
- struct wds_DHCPv4HWConfig
- struct wds_DHCPv4Option
- struct wds_DHCPv4OptionList
- struct pack_wds_SLQSSSetDHCPv4ClientConfig_t
- struct unpack_wds_GetDataBearerTechnology_t

Macros

- #define IPV6_ADDRESS_ARRAY_SIZE 8
- #define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24
- #define PACK_WDS_IPV4 4
- #define PACK_WDS_IPV6 6
- #define BYT_STAT_STAT_MASK 0X000000C0
- #define WDS_DHCP_MAX_NUM_OPTIONS 30
- #define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /∗ current max size of raw message in SDK process is 2048 ∗/

Typedefs

- typedef union unpackWdsProfileParam UnpackQmiProfileInfo

Functions

- int pack_wds_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)

9.61.1 Macro Definition Documentation

9.61.1.1 #define BYT_STAT_STAT_MASK 0X000000C0

9.61.1.2 #define IPV6_ADDRESS_ARRAY_SIZE 8

9.61.1.3 #define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24

9.61.1.4 #define PACK_WDS_IPV4 4

9.61.1.5 #define PACK_WDS_IPV6 6

9.61.1.6 #define WDS_DHCP_MAX_NUM_OPTIONS 30

9.61.1.7 #define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */

9.61.2 Typedef Documentation

9.61.2.1 typedef union unpackWdsProfileParam UnpackQmiProfileInfo

9.61.3 Function Documentation

9.61.3.1 int pack_wds_DHCPv4ClientLeaseChange (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_DHCPv4ClientLeaseChange_t * *reqArg*)

DHCPv4 lease state changes pack

Parameters

in,out	*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

9.61.3.2 int pack_wds_GetAutoconnect (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets auto connect data session setting pack

Parameters

in,out	*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

9.61.3.3 int pack_wds_GetByteTotals (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get Rx/Tx byte counts since the start of the last packet data session pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.4 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	*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

Note

PDN Specific: No

9.61.3.5 int pack_wds_GetDataBearerTechnology (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get current data bearer technology pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.6 int pack_wds_GetDefaultProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetDefaultProfile_t * *reqParam*)

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

Note

PDN Specific: Yes

9.61.3.7 int pack_wds_GetDefaultProfileNum (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetDefaultProfileNum_t * *pReqParam*)

get default profile number pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.61.3.8 int pack_wds_GetDormancyState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetDormancyState_t * *reqParam*)

get dormancy state pack

Parameters

in,out	*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

Note

PDN Specific: No

9.61.3.9 int pack_wds_GetLastMobileIPError (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetLastMobileIPError_t * *pReqParam*)

get current data system pack

Parameters

in,out	*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

Note

PDN Specific: No

9.61.3.10 int pack_wds_GetMobileIP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetMobileIP_t * *pReqParam*)

get mobile ip mode pack

Parameters

in,out	*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

Note

PDN Specific: No

9.61.3.11 int pack_wds_GetMobileIPProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetMobileIPProfile_t * *reqParam*)

get mobile ip profile pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.12 int pack_wds_GetPacketStatistics (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetPacketStatistics_t * *pReq*)

gets current packet transfer counter values pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.61.3.13 int pack_wds_GetPacketStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetPacketStatus_t * *reqParam*)

get packet status pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.14 int pack_wds_GetSessionDuration (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetSessionDuration_t * *reqParam*)

get session duration pack

Parameters

in,out	*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

Note

PDN Specific: No

9.61.3.15 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	*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

Note

PDN Specific: Yes

9.61.3.16 int pack_wds_RMSetTransferStatistics (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_RMSetTransferStatistics_t * *reqParam*)

rm set transfer statistics pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
out	*reqParam*	request parameter

Note

PDN Specific: No

9.61.3.17 int pack_wds_SetAutoconnect (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetAutoconnect_t * *reqArg*)

Auto connect data session parameters pack

Parameters

in,out	*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

9.61.3.18 int pack_wds_SetDefaultProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetDefaultProfile_t * *reqParam*)

set default profile pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.19 int pack_wds_SetDefaultProfileNum (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetDefaultProfileNum_t * *pReqParam*)

set default profile number pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.61.3.20 int pack_wds_SetMobileIP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetMobileIP_t * *reqArg*)

Sets the current mobile IP setting pack

Parameters

in,out	*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

9.61.3.21 int pack_wds_SetMobileIPParameters (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetMobileIPParameters_t * *reqArg*)

Sets the specified mobile IP parameters pack

Parameters

in,out	*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

9.61.3.22 int pack_wds_SetMobileIPProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetMobileIPProfile_t * *reqParam*)

set mobile ip profile pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
out	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.61.3.23 int pack_wds_SLQSCreateProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSCreateProfile_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.61.3.24 int pack_wds_SLQSDeleteProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSDeleteProfile_t * *reqParam*)

delete stored profile pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.25 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	*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

Note

PDN Specific: Yes

9.61.3.26 int pack_wds_SLQSGetCurrDataSystemStat (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetCurrDataSystemStat_t * *pReqParam*)

get current data system pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.27 int pack_wds_SLQSGetCurrentChannelRate (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get current Tx/Rx channel bitrate of the current packet data pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.28 int pack_wds_SLQSGetDataBearerTechnology (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetDataBearerTechnology_t * *pReqParam*)

get data bearer technology pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
out	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.61.3.29 int pack_wds_SLQSGetDUNCallInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetDUNCallInfo_t * *reqParam*)

get dun call info pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
out	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.61.3.30 int pack_wds_SLQSGetProfileSettings (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetProfileSettings_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.61.3.31 int pack_wds_SLQSGetRuntimeSettings (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetRuntimeSettings_t * *reqArg*)

get runtime settings pack

Parameters

in,out	*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

9.61.3.32 int pack_wds_SLQSModifyProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSModifyProfile_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.61.3.33 int pack_wds_SLQSResetPacketStatics (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Reset packet data transfer statistics pack

Parameters

in,out	*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

9.61.3.34 int pack_wds_SLQSSet3GPPConfigItem (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSet3GPPConfigItem_t * *reqParam*)

set 3Gpp config items pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.35 int pack_wds_SLQSSetIPFamilyPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSetIPFamilyPreference_t * *pReqParam*)

Set IP Family Preference pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
out	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.36 int pack_wds_SLQSSetWdsEventCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSetWdsEventCallback_t * *reqArg*)

set event callback pack

Parameters

in,out	*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

9.61.3.37 int pack_wds_SLQSSGetDHCPv4ClientConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSGetDHCPv4ClientConfig_t * *pReq*)

get DHCPv4 Client Config pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	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

9.61.3.38 int pack_wds_SLQSSGetLoopback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get the value of loopback mode and multiplier pack

Parameters

in,out	*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

Note

PDN Specific: Yes

9.61.3.39 int pack_wds_SLQSSSetDHCPv4ClientConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSSetDHCPv4ClientConfig_t * *reqArg*)

Gets the DHCP Client V4 Configuration pack

Parameters

in,out	*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

9.61.3.40 int pack_wds_SLQSSSetLoopback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSSSetLoopback_t * *reqArg*)

Enable/disable Data Loopback Mode and set the value of loopback multiplier pack

Parameters

in,out	*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

9.61.3.41 int pack_wds_SLQSStartDataSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSStartDataSession_t * *reqArg*)

Start data session

Parameters

in,out	*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

9.61.3.42 int pack_wds_SLQSStopDataSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSStopDataSession_t * *reqArg*)

stop data session pack

Parameters

in,out	*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

9.61.3.43 int pack_wds_SLQSWdsGoActive (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets the device into Active state pack

Parameters

in,out	*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

9.61.3.44 int pack_wds_SLQSWdsGoDormant (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Gets the device into dormant state pack

Parameters

in,out	*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

9.61.3.45 int pack_wds_SLQSWdsSetEventReport (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSWdsSetEventReport_t * reqArg)

Sets the event report parameters pack

Parameters

in,out	*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

9.61.3.46 int pack_wds_SLQSWdsSwiPDPRuntimeSettings (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t * reqParam)

swi pdp runtime settings pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
out	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.61.3.47 int unpack_wds_DHCPv4ClientLease_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_DHCPv4ClientLease_ind_t * *pOutput*)

DHCP lease state has changed indication unpack

Parameters

in	*pResp*	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

9.61.3.48 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

9.61.3.49 int unpack_wds_GetAutoconnect (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetAutoconnect_t * *pOutput*)

Gets auto connect data session setting unpack

Parameters

in	*pResp*	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

9.61.3.50 int unpack_wds_GetByteTotals (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetByteTotals_t * *pOutput*)

get Rx/Tx byte counts since the start of the last packet data session unpack

Parameters

in	*pResp*	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

9.61.3.51 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

9.61.3.52 int unpack_wds_GetDataBearerTechnology (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDataBearerTechnology_t * *pOutput*)

get current data bearer technology unpack

Parameters

in	*pResp*	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

9.61.3.53 int unpack_wds_GetDefaultProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDefaultProfile_t * *pOutput*)

Parameters

in	*pResp*	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

9.61.3.54 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

9.61.3.55 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

9.61.3.56 int unpack_wds_GetLastMobileIPError (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetLastMobileIP-Error_t * *pOutput*)

get current data system unpack

Parameters

in	*pResp*	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

9.61.3.57 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

9.61.3.58 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

9.61.3.59 int unpack_wds_GetPacketStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetPacketStatistics_t * *pOutput*)

gets current packet transfer counter values unpack

Parameters

in	*pResp*	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

9.61.3.60 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

9.61.3.61 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

9.61.3.62 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

9.61.3.63 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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.64 int unpack_wds_RMTransferStatistics_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_RMTransferStatistics_ind_t * *pOutput*)

RM transfer statistics indication unpack

Parameters

in	*pResp*	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

9.61.3.65 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

9.61.3.66 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
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.67 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

9.61.3.68 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

9.61.3.69 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

9.61.3.70 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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.71 int unpack_wds_SLQSCreateProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSCreateProfile_t * *pOutput*)

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response structure to fill
in	*pProfileId*	profile id pointer passed in req

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.72 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

9.61.3.73 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

9.61.3.74 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

9.61.3.75 int unpack_wds_SLQSGetCurrentChannelRate (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetCurrentChannelRate_t * *pOutput*)

get current Tx/Rx channel bitrate of the current packet data unpack

Parameters

in	*pResp*	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

9.61.3.76 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

9.61.3.77 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

9.61.3.78 int unpack_wds_SLQSGetProfileSettings (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetProfileSettings_t * *pOutput*)

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

9.61.3.79 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

9.61.3.80 int unpack_wds_SLQSModifyProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSModifyProfile_t * *pOutput*)

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

9.61.3.81 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

9.61.3.82 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
out	*pOutput*	response unpacked

9.61.3.83 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

9.61.3.84 int unpack_wds_SLQSSetPacketSrvStatusCallback (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSSetPacketSrvStatusCallback_t * *pOutput*)

set packet srv status callback unpack

Parameters

in	*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

9.61.3.85 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

9.61.3.86 int unpack_wds_SLQSSetWdsEventCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSSetWdsEventCallback_ind_t * *pOutput*)

set event callback unpack

Parameters

in	*pResp*	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

9.61.3.87 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

9.61.3.88 int unpack_wds_SLQSSGetLoopback (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSSGetLoopback_t * *pOutput*)

get the value of loopback mode and multiplier unpack

Parameters

in	*pResp*	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

9.61.3.89 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

9.61.3.90 int unpack_wds_SLQSSetLoopback (uint8_t * *pResp*, uint16_t *respLen*)

Enable/disable Data Loopback Mode and set the value of loopback multiplier unpack

Parameters

in	*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

9.61.3.91 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

9.61.3.92 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

9.61.3.93 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

9.61.3.94 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

9.61.3.95 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

9.61.3.96 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, 1174
_GetProfileSettingIn, 64
ProfileID, 64
ProfileType, 64
_GetProfileSettingOut, 64
curProfile, 65
pExtErrCode, 65
_MitigationDevInfo, 67
deviceID, 67
deviceIdLen, 67
_QmiSwiAvmsEventReportBinaryUpdateSessionInfo-
Resp, 71
bBinaryType, 73
bSerity, 73
bState, 73
bUserInputRequest, 73
szDescription, 73
szName, 73
szVersion, 73
TlvPresent, 73
ulPkgDownloadComplete, 73
ulPkgDownloadSize, 73
wDescriptionLength, 73
wNameLength, 73
wUpdateCompeteStatus, 73
wUserInputTimeout, 73
wVersionLength, 73
_QmiSwiAvmsEventReportConfigResp, 73
bState, 74
bUserInputRequest, 74
szAlertMsg, 74
TlvPresent, 74
wAlertMsgLength, 74
wUserInputTimeout, 74
_QmiSwiAvmsEventReportConnectionRequestResp,
74
bUserInputRequest, 75
TlvPresent, 75
wUserInputTimeout, 75
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
bType, 75
TlvPresent, 75
wErrorCode, 75
_QmiSwiAvmsEventReportHTTPStatusResp, 75
TlvPresent, 76
wHTTPStatus, 76
_QmiSwiAvmsEventReportNotificationResp, 77
bNotification, 77
TlvPresent, 77
wSessionStatus, 78
_QmiSwiAvmsEventReportPackageIDResp, 78
bPackageID, 78
TlvPresent, 78
_QmiSwiAvmsEventReportRegStatusResp, 78
bRegStatus, 78
TlvPresent, 78
_QmiSwiAvmsEventReportSessionTypeResp, 78
bType, 79
TlvPresent, 79
_QmiSwiAvmsEventReportWAMSParamChangeResp,
79
TlvPresent, 79
wWamsChangeMask, 79
_QmiSwiAvmsSetSettingsAPNInfo, 79
bAPNLength, 80
bPWDLength, 80
bUnameLength, 80
szAPN, 80
szPWD, 80
szUname, 80
_QmiSwiAvmsSetSettingsConnectionRetryTimers, 80
Timers, 80
_QmiSwiAvmsSetSettingsPeriodInfo, 80
ulMax, 80
ulMin, 80
_SLQSAVMSSetSettingsReqPara, 83
pAPNInfo, 85
pAutoConnect, 85
pAutoReboot, 85
pConnectionRetryTimers, 85
pFwAutoSDM, 85
pFwAutoUpdate, 85
pFwAutodownload, 85
pNotifStore, 85
pPeroidInfo, 85
pPollingTimer, 85
_SLQSAVMSSettings, 85
pAPNInfo, 85
pAutoConnect, 85
pAutoReboot, 85
pConnectionRetryTimers, 85
pFwAutoSDM, 85
pFwAutoUpdate, 85
pFwAutodownload, 85
pNotificationStore, 85
pOMADMEnabled, 86
pPeroidsInfo, 86

pPollingTimer, 86
_SLQSAVMSSettingsAPNInfo, 86
bAPNLength, 86
bPWDLength, 86
bUnameLength, 86
pAPN, 86
pPWD, 86
pUname, 86
_SLQSAVMSSettingsConnectionRetryTimers, 86
Timers, 86
_SLQSAVMSSettingsPeriodsInfo, 86
max, 87
min, 87
_SLQSOMADMSessionInfo, 89
pDate, 91
pDateLength, 91
pPkgDescLength, 91
pPkgDescription, 91
pPkgName, 91
pPkgNameLength, 91
pRetryCount, 91
pSessionState, 91
pSessionType, 91
pSeverity, 91
pSource, 91
pSourceLength, 91
pStatus, 91
pTime, 91
pTimeLength, 91
pUpdateCompleteStatus, 91
_SLQSOMADMSettings, 91
pAutosdm, 93
pFOTAUpdate, 93
pFOTAdownload, 93
pFwAutoCheck, 93
pOMADMEnabled, 93
_SLQSOMADMSettingsReqParams, 93
FOTAUpdate, 94
FOTAdownload, 93
pAutosdm, 94
_SLQSOMADMSettingsReqParams3, 94
FOTAUpdate, 95
FOTAdownload, 94
pAutosdm, 95
pFwAutoCheck, 95
_SLQSSwiGetHostDevInfoParams, 95
bManSize, 96
bModelSize, 96
bPlasmaIDSize, 96
bSWVerSize, 96
pManString, 96
pModelString, 96
pPlasmaIDString, 96
pSWVerString, 96
_SLQSSwiGetOSInfoParams, 96
bNameSize, 96
bVersionSize, 96
pNameString, 96
pVersionString, 97
_SLQSSwiGetSerialNoExtParams, 97
meidLength, 97
pMeidString, 97
_SLQSSwiSetHostDevInfoParams, 97
bManSize, 98
bModelSize, 98
bPlasmaIDSize, 98
bSWVerSize, 98
pManString, 98
pModelString, 98
pPlasmaIDString, 98
pSWVerString, 98
_SLQSSwiSetOSInfoParams, 98
bNameSize, 99
bVersionSize, 99
pNameString, 99
pVersionString, 99
SlqsNas3GppNetworkRAT, 87
MCC, 87
MNC, 87
RAT, 87
_SwiAvmsEventInd, 99
BinaryUpdateSessionInfo, 99
Config, 99
ConnectionRequest, 99
DataSessionStatus, 99
HTTPStatus, 100
Notification, 100
PackageID, 100
RegStatus, 100
SessionType, 100
WAMSParaChanged, 100
_getIndicationRegResp, 63
pRegCallStatInfoEvt, 64
pRegTransLayerInfoEvt, 64
pRegTransNWRegInfoEvt, 64
_getResetInfoNotification, 65
source, 66
type, 66
_getTransLayerInfoResp, 66
pRegInd, 66
pTransLayerInfo, 66
_getTransNWRegInfoResp, 66
pRegStatus, 67
_modemTempNotification, 67
ModemTempState, 68
ModemTemperature, 68
_packetSrvStatus, 68
bearerID, 70
connStatus, 70
ipFamily, 70
pQmiInterfaceInfo, 70
reconfigReqd, 70
sessionEndReason, 70
techName, 70
verboseSessnEndReason, 70
verboseSessnEndReasonType, 70

_qaQmi3GPP2BroadcastCfgInfo, 70
activated_ind, 71
CDMABroadcastConfig, 71
num_instances, 71
_qaQmi3GPPBroadcastCfgInfo, 71
activated_ind, 71
broadcastConfig, 71
num_instances, 71
_setIndicationRegReq, 80
pRegCallStatInfoEvt, 81
pRegTransLayerInfoEvt, 81
pRegTransNWRegInfoEvt, 81
_slqs3GPPConfigItem, 81
LTEAttachProfileListLen, 83
p3gppRelease, 83
pDefaultPDNEnabled, 83
pLTEAttachProfile, 83
pLTEAttachProfileList, 83
pProfileList, 83
_slqsNetworkScanInfo, 87
pNetworkInfo, 88
pNetworkInfoInstances, 88
pPCSDigitInfo, 88
pPCSDigitInstances, 88
pRATInfo, 88
pRATInstances, 88
pScanResult, 88
_sysSelectPrefInfo, 100
pBandPref, 104
pEmerMode, 104
pGWAcqOrderPref, 104
pLTEBandPref, 104
pModePref, 104
pNetSelPref, 104
pPRLPref, 104
pRoamPref, 104
pSrvDomainPref, 104
_sysSelectPrefParams, 105
pAcqOrderPref, 110
pBandPref, 110
pCSGID, 110
pChgDuration, 110
pEmerMode, 110
pGWAcqOrderPref, 110
pLTEBandPref, 110
pMNCIncPCSDigStat, 110
pModePref, 110
pNetSelPref, 110
pPRLPref, 110
pRAT, 110
pRoamPref, 110
pSrvDomainPref, 111
pSrvRegRestriction, 111
pTdscdmaBandPref, 111
_transLayerInfoNotification, 111
pTransLayerInfo, 112
regInd, 112
_transLayerinfo, 111
TransCap, 111
TransType, 111
_transNWRegInfoNotification, 112
NWRegStat, 112

AAASPI
unpack_wds_GetMobileIPProfile_t, 1168
AAAState
unpack_wds_GetMobileIPProfile_t, 1168
ABSOLUTE_VALIDITY
qaGobiApiSms.h, 1696
ACT_CODE_MAX_SIZE
dms.h, 1336
ALS
getAllCallInformation, 251
voice_getAllCallInformation, 1221
AMSSString
unpack_dms_GetFirmwareRevision_t, 997
unpack_dms_GetFirmwareRevisions_t, 998
APNName
unpack_wds_SLQSGetRuntimeSettings_t, 1179
AVMS (SWIAVMS), 59
absoluteValidity
cdmaMsgDecodingParams, 166
accelAcceptReady
qaGobiApiCbk.h, 1463
accelAcceptReady_s, 113
batchPerSec, 113
injectEnable, 113
samplesPerBatch, 113
accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 602
accelTempAcceptReady
qaGobiApiCbk.h, 1463
accelTempAcceptReady_s, 113
batchPerSec, 114
injectEnable, 114
samplesPerBatch, 114
acceleroData
pack_loc_SLQSLOCInjectSensorData_t, 602
acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 602
AccessMac
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733
accolc
pack_nas_SetACCOLC_t, 606
accuracy
pack_pds_SetPDSDefaults_t, 630
ackIndicator
SMSTransferRouteMTMessage, 882
sMSTransferRouteMTMessage, 883
acqOrdeLen
acqOrderPref, 115
nas_acqOrderPref, 409
acqOrderPref, 114
acqOrdeLen, 115
pAcqOrder, 115
acroamsetting

pack_wds_SetAutoconnect_t, 681
slqsautoconnect, 839
acsetting
pack_wds_SetAutoconnect_t, 681
slqsautoconnect, 839
actCode
pack_dms_ActivateAutomatic_t, 577
ActPilotPNElement, 115
ActSetPilotPN, 115
ActSetPilotPNStrength, 115
ActSetCnt
nas_NetworkStat1x, 470
NetworkStat1x, 563
ActSetPilotPN
ActPilotPNElement, 115
nas_ActPilotPNElement, 409
ActSetPilotPNStrength
ActPilotPNElement, 115
nas_ActPilotPNElement, 409
action
pack_nas_InitiateDomainAttach_t, 606
slqsautoconnect, 839
ssdatasession_params, 888
ActivateAutomatic
qaGobiApiDms.h, 1555
activated_ind
_qaQmi3GPP2BroadcastCfgInfo, 71
_qaQmi3GPPBroadcastCfgInfo, 71
sms_qaQmi3GPP2BroadcastCfgInfo, 862
sms_qaQmi3GPPBroadcastCfgInfo, 862
activationState
pack_dms_SetActivationStatusCallback_t, 579
activationStatus
dms_ActivationStatusTlv, 222
ActivationStatusTlv
unpack_dms_SetEventReport_ind_t, 1006
activeBandClass
nas_RFInfoTlv, 488
RFBandInfoElements, 781
activeChannel
nas_RFInfoTlv, 488
RFBandInfoElements, 781
activeInd
messageWaitingInfoContent, 404
sms_messageWaitingInfoContent, 860
ActiveStatus
CLIPResp, 179
CLIRResp, 179
CNAPResp, 181
COLPResp, 182
COLRResp, 183
voice_CLIPResp, 1213
voice_CLIRResp, 1214
voice_CNAPResp, 1214
voice_COLPResp, 1215
voice_COLRResp, 1216
ActiveTechPref
unpack_nas_GetNetworkPreference_t, 1049
AddCDMASysInfo, 115
geoSysIdx, 116
regPrd, 116
AddSysInfo, 116
cellBroadcastCap, 116
geoSysIdx, 116
addr
IPv4Addr, 326
IPv6Addr, 326
unpack_qos_IPv4Addr_t, 1091
unpack_qos_IPv6Addr_t, 1091
address
unpack_wds_GetMobileIPProfile_t, 1168
addressSize
unpack_sms_GetSMSCAddress_t, 1112
aid
uim_refreshevent, 951
uim_sessionInformation, 953
uim_UIMSessionInformation, 957
UIMRefreshEvent, 971
UIMSessionInformation, 974
aidLength
appStats, 124
appStatus, 127
uim_appStatus, 936
uim_refreshevent, 951
uim_sessionInformation, 953
uim_UIMSessionInformation, 957
UIMRefreshEvent, 971
UIMSessionInformation, 974
aidVal
appStats, 124
appStatus, 127
uim_appStatus, 936
aidingIndicatorMask
loc_sensorDataUsage, 368
sensorDataUsage_s, 800
t_sensor, 912
airTimer, 116
airTimerValue, 117
namID, 117
airTimerValue
airTimer, 117
voice_airTimer, 1195
alertPitch
signalInfo, 834
alertingPattern
arrAlertingPattern, 128
voice_arrAlertingPattern, 1198
AlertingType
arrAlertingType, 129
voice_arrAlertingType, 1199
alertmsg
omaDmConfigTlv, 569
omaDmConfigTlvExt, 571
unpack_omaDmConfigTlv_t, 1085
alertmsglength
omaDmConfigTlv, 569

omaDmConfigTlvExt, 571
unpack_omaDmConfigTlv_t, 1085
allCallsAlphaIDInfo, 117
AlphaIDInfo, 117
callID, 117
allCallsAlphaIDInfoArr
arrAlphaID, 129
voice_arrAlphaID, 1199
allCallsDiagInfo, 117
callID, 118
DiagInfo, 118
AllCallsUUSInfo
arrUUSInfo, 135
voice_arrUUSInfo, 1205
allCallsUUSInfo, 118
callID, 118
uusInfo, 118
alphaDcs
alphaIDInfo, 119
voice_alphaIDInfo, 1197
AlphaID
CatAlPhaIdentifierTlv, 158
AlphaIDInfo
allCallsAlphaIDInfo, 117
voice_allCallsAlphaIDInfo, 1195
alphaIDInfo, 118
alphaDcs, 119
alphaLen, 119
alphaText, 119
alphaIDLen
SMSAsyncRawSend_s, 869
AlphaIDLength
CatAlPhaIdentifierTlv, 158
alphaLen
alphaIDInfo, 119
voice_alphaIDInfo, 1197
alphaText
alphaIDInfo, 119
voice_alphaIDInfo, 1197
alphabet
wcdmaMsgEncodingParams, 1288
altSrcInfo_t, 120
coverage, 121
linkage, 121
source, 121
Altitude
GPSStateInfo, 287
altitudeAssumed
unpack_loc_GnssSvInfo_Ind_t, 1035
altitudeSrcInfo, 119
coverage, 120
linkage, 120
pack_loc_SLQSLOCInjectPosition_t, 599
source, 120
altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 599
altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 599
ambr_dl
sApnExtraParams, 793
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1096
ambr_dl_ext
sApnExtraParams, 793
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1096
ambr_dl_ext2
sApnExtraParams, 793
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1096
ambr_ul
sApnExtraParams, 793
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1096
ambr_ul_ext
sApnExtraParams, 793
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1096
ambr_ul_ext2
sApnExtraParams, 793
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1096
amssSize
unpack_dms_GetFirmwareRevision_t, 997
unpack_dms_GetFirmwareRevisions_t, 998
AnswerUSSD
qaGobiApiVoice.h, 1802
apdoxypages.c, 1325
apnId
pack_qos_SLQSQosSwiReadApnExtraParams_t, 632
pack_qos_SLQSQosSwiReadDataStats_t, 632
sApnExtraParams, 793
sQosStat, 885
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1096
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
apnName
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1186
apnname
unpack_wds_GetDefaultProfile_t, 1166
apnsize
unpack_wds_GetDefaultProfile_t, 1166
appNameLength
loc_LocApplicationInfo, 364
LocApplicationInfo, 371
appProviderLength
loc_LocApplicationInfo, 364
LocApplicationInfo, 371
appState
appStats, 124
appStatus, 127
uim_appStatus, 936
appStats, 121
aidLength, 124

aidVal, 124
appState, 124
appType, 124
persoFeature, 124
persoRetries, 124
persoState, 124
persoUnblockRetries, 124
pin1Retries, 124
pin1State, 124
pin2Retries, 124
pin2State, 124
puk1Retries, 124
puk2Retries, 124
univPin, 124
AppStatus
slotInf, 836
slotInfo, 838
uim_slotInfo, 956
appStatus, 124
aidLength, 127
aidVal, 127
appState, 127
appType, 127
persoFeature, 127
persoRetries, 127
persoState, 127
persoUnblockRetries, 127
pin1Retries, 127
pin1State, 127
pin2Retries, 127
pin2State, 127
puk1Retries, 127
puk2Retries, 127
univPin, 127
appType
appStats, 124
appStatus, 127
uim_appStatus, 936
appVersionLength
loc_LocApplicationInfo, 364
LocApplicationInfo, 371
appVersionValid
loc_LocApplicationInfo, 364
LocApplicationInfo, 371
Application
unpack_nas_GetCDMANetworkParameters_t, 1045
appversion_str
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
arfcn
GERANInfo, 249
gsmCellInfo, 289
nas_GERANInfo, 434
nas_gsmCellInfo, 435
arrAlertingPattern, 127
alertingPattern, 128
callID, 128
numInstances, 128
arrAlertingType, 128
AlertingType, 129
callID, 129
numInstances, 129
arrAlphaID, 129
allCallsAlphaIDInfoArr, 129
numInstances, 129
arrCallEndReason, 130
callEndReason, 131
callID, 131
numInstances, 131
arrCallInfo, 131
getAllCallInfo, 131
numInstances, 131
arrCallInfomation
voiceSetAllCallStatusCbkInfo, 1268
arrCalledPartyNum, 129
CalledPartyNum, 130
numInstances, 130
arrConnectPartyNum, 131
ConnectedPartyNum, 132
numInstances, 132
arrDiagInfo, 132
DiagInfo, 132
numInstances, 132
arrRedirPartyNum, 132
numInstances, 133
RedirPartyNum, 133
arrRemotePartyName, 133
GetAllCallRmtPtyName, 133
numInstances, 133
arrRemotePartyNum, 133
numInstances, 134
RmtPtyNum, 134
arrSvcOption, 134
callID, 134
numInstances, 134
srvOption, 134
arrUUSInfo, 135
AllCallsUUSInfo, 135
numInstances, 135
arrfileInfo
registerRefresh, 777
uim_refreshevent, 951
uim_registerRefresh, 952
UIMRefreshEvent, 971
AtCmdPort
DcsUsbPortNames, 215
Audio Service (AUDIO), 53
auth
unpack_wds_GetDefaultProfile_t, 1166
authData
pack_uim_SLQSUIMAuthenticate_t, 652
UIMAuthenticateReq, 959
AuthProt
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733

authenticateResult, 135
content, 136
contentLen, 136
Authentication
unpack_wds_SLQSGetRuntimeSettings_t, 1179
authentication
pack_wds_SetDefaultProfile_t, 682
authenticationData, 136
context, 136
data, 137
dataLen, 137
Autosdm
unpack_swioma_SLQSOMADMGetSettings_t, 1127
avgPeriod
LTESigRptCfg, 395
LTESigRptConfig, 396
nas_LTESigRptCfg, 459
nas_LTESigRptConfig, 460
azimuth
loc_satelliteInfo, 367
satelliteInfo, 795

bAPNLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bAltitudeAssumed
gnssSvInfoNotification, 282
bAuto
pack_pds_SetPortAutomaticTracking_t, 631
bBinaryType
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
bEnable
pack_dms_UIMSetPINProtection_t, 587
pack_nas_SLQSSetSignalStrengthsCallback_t, 624
bForceDownload
pack_fms_SetImagesPreference_t, 590
bICCID
slot_t, 835
UIMSlotStatus, 976
bICCIDLength
slot_t, 835
UIMSlotStatus, 976
bLogicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 657
slot_t, 835
UIMSlotStatus, 976
UIMSwitchSlotReq, 978
bManSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bModelSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bNameSize
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
bNotification
_QmiSwiAvmsEventReportNotificationResp, 77
bNumberOfPhySlots
UIMSlotStatusChangeInfo, 977
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 1131
BOOL
SwiDataTypes.h, 1904
BPTlv
NASQmiCbkNasSystemSelPrefInd, 549
bPWDLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bPackageID
_QmiSwiAvmsEventReportPackageIDResp, 78
bPlasmaIDSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bRegStatus
_QmiSwiAvmsEventReportRegStatusResp, 78
bResetStatistics
rmTrasnferStaticsReq, 781
swiRMTrasnferStaticsReq, 910
bSWVerSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bSerity
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
bState
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
bType
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
_QmiSwiAvmsEventReportSessionTypeResp, 79
BUILD_ID_LEN
qaGobiApiFms.h, 1588
BUILD_ID_MAX_LEN
qaGobiApiFms.h, 1588
bUnameLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bUserInputRequest
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
_QmiSwiAvmsEventReportConnectionRequestResp, 75
bVersionSize
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
BYT_STAT_STAT_MASK
wds.h, 1966
BYTE
SwiDataTypes.h, 1904
band

LTEInfo, 387
nas_LTEInfo, 451
band1900
gsmCellInfo, 289
nas_gsmCellInfo, 436
band_pref
NASBandPreferenceTlv, 521
BandCapability
unpack_dms_GetBandCapability_t, 992
bandCapability
BandCapabilityResp, 140
unpack_dms_SLQSGetBandCapability_t, 1012
BandCapabilityResp, 137
bandCapability, 140
pLteBandCapability, 140
pTdsBandCapability, 140
bandPref
pack_nas_SLQSSetBandPreference_t, 623
bandwidth
LTEInfo, 387
nas_LTEInfo, 451
baseId
CDMAInfo, 164
CDMASysInfo, 173
nas_CDMAInfo, 414
nas_CDMASysInfo, 418
baseLat
CDMAInfo, 164
CDMASysInfo, 173
nas_CDMAInfo, 414
nas_CDMASysInfo, 418
baseLong
CDMAInfo, 164
CDMASysInfo, 173
nas_CDMAInfo, 414
nas_CDMASysInfo, 418
BasestationID
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
BasestationLatitude
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
BasestationLongitude
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
batchPerSec
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 294
gyroTempAcceptReady_s, 295
BdsSV, 140
id, 140
mask, 140
BdsSVInfo, 140
len, 141
pSV, 141
BearerID
unpack_qos_QosFlowInfo_t, 1094
bearerID
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1181
bearerId
sQosFlowStat, 884
unpack_QosFlowStat_t, 1110
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
BinaryUpdateSessionInfo
_SwiAvmsEventInd, 99
bootSize
unpack_dms_GetFirmwareRevisions_t, 998
BootString
unpack_dms_GetFirmwareRevisions_t, 998
bootversion_str
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
Broadcast
unpack_nas_GetCDMANetworkParameters_t, 1045
broadcastActivate
pack_sms_SLQSSetSmsBroadcastActivation_t, 641
BroadcastConfig, 141
fromServiceId, 142
selected, 142
toServiceId, 142
broadcastConfig
_qaQmi3GPPBroadcastCfgInfo, 71
sms_qaQmi3GPPBroadcastCfgInfo, 862
bsInfoValid
CDMASysInfo, 173
nas_CDMASysInfo, 418
bsPRev
CDMASysInfo, 173
nas_CDMASysInfo, 418
bsPRevValid
CDMASysInfo, 173
nas_CDMASysInfo, 418
bsic
GERANInfo, 249
nas_GERANInfo, 434
bsicId
gsmCellInfo, 289
nas_gsmCellInfo, 436
bucketSz
tokenBucket, 926
unpack_qos_tokenBucket_t, 1108
buildID
CurrImageInfo, 199
FMSImageIdElement, 245
image_info_t, 305
ImageIdElement, 307
buildIDLen
CurrImageInfo, 199
image_info_t, 305
buildIDLength

FMSImageIdElement, 245
ImageIdElement, 307
buildId
FMSImageElement, 244
ImageElement, 306
buildIdLength
FMSImageElement, 244
ImageElement, 306
BurstDTMFInfo
pack_voice_SLQSVoiceBurstDTMF_t, 663
voiceBurstDTMFInfo, 1233
burstDTMFInfo, 142
digitCnt, 142
pCallID, 142
pDigitBuff, 142
ByteLoopbackMode
unpack_wds_SLQSSGetLoopback_t, 1184
WDSGetLoopbackData, 1317
ByteLoopbackMultiplier
unpack_wds_SLQSSGetLoopback_t, 1184
WDSGetLoopbackData, 1317
ByteTotalsElmntsV4
WdsByteTotals, 1308
ByteTotalsElmntsV6
WdsByteTotals, 1308

CATEventDataType, 159
eventMask, 159
pErrorMask, 159
CATSendEnvelopeCommand
qaGobiApiCat.h, 1452
CATSendTerminalResponse
qaGobiApiCat.h, 1452
CBK_DISABLE_EVENT
qaGobiApiCbk.h, 1461
CBK_ENABLE_EVENT
qaGobiApiCbk.h, 1461
CBK_NOCHANGE
qaGobiApiCbk.h, 1461
CCETlv
QmiCbkCatEventStatusReportInd, 739
CDMA_P_Rev
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
CDMABroadcastConfig, 161
_qaQmi3GPP2BroadcastCfgInfo, 71
language, 162
selected, 162
serviceCategory, 162
sms_qaQmi3GPP2BroadcastCfgInfo, 862
CDMAChannel, 162
priChA, 162
priChB, 162
secChA, 162
secChB, 163
CDMAECIOThresh, 163
CDMAECIOThreshListLen, 163
pCDMAECIOThreshList, 163
CDMAECIOThreshListLen
CDMAECIOThresh, 163
nas_CDMAECIOThresh, 413
CDMAInfo, 163
baseId, 164
baseLat, 164
baseLong, 164
nid, 164
refpn, 164
sid, 164
CDMARSSIThresh, 169
CDMARSSIThreshListLen, 169
pCDMARSSIThreshList, 169
CDMARSSIThreshListLen
CDMARSSIThresh, 169
nas_CDMARSSIThresh, 415
CDMASSInfo, 169
ecio, 170
rssi, 170
unpack_nas_SLQSNasGetSigInfo_t, 1072
CDMASysInfo, 170
baseId, 173
baseLat, 173
baseLong, 173
bsInfoValid, 173
bsPRev, 173
bsPRevValid, 173
ccsSupported, 173
ccsSupportedValid, 173
cdmaSysIdValid, 173
isSysPrlMatch, 173
isSysPrlMatchValid, 173
MCC, 173
MNC, 173
networkID, 173
networkIdValid, 173
pRevInUse, 174
pRevInUseValid, 174
packetZone, 174
packetZoneValid, 174
sysInfoCDMA, 174
systemID, 174
CDMASysInfoExt, 174
imsi_11_12, 174
MCC, 174
CDMASystemInfoExt
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
CHAR
SwiDataTypes.h, 1904
CK_MAX_SIZE
dms.h, 1336
CLIPResp, 178
ActiveStatus, 179
ProvisionStatus, 179
CLIRResp, 179
ActiveStatus, 179
ProvisionStatus, 179
CNAPResp, 181

ActiveStatus, 181
ProvisionStatus, 181
COLPResp, 181
ActiveStatus, 182
ProvisionStatus, 182
COLRResp, 182
ActiveStatus, 183
ProvisionStatus, 183
CONFIG_LEN
qaGobiApiSms.h, 1696
CQIValueCW0
LteCQIParm, 383
unpack_nas_SLQSSwiGetLteCQI_t, 1080
CQIValueCW1
LteCQIParm, 383
unpack_nas_SLQSSwiGetLteCQI_t, 1080
CSDomain
unpack_nas_GetServingNetwork_t, 1051
CSGID, 192
id, 193
mcc, 193
mnc, 193
mncPcsDigits, 193
rat, 193
CUGIndex
CUGInfo, 194
voice_CUGInfo, 1218
CUGInfo, 193
CUGIndex, 194
SuppOA, 194
SuppPrefCUG, 194
CallBacK registration (CBK), 41
CallBarStatus
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
callBarStatus, 143
csBarStatus, 144
psBarStatus, 144
CallBarringSysInfo, 142
csBarStatus, 143
psBarStatus, 143
callDuration
unpack_wds_GetSessionDuration_t, 1172
CallEndReason
DUNCallInfoInd, 231
callEndReason
arrCallEndReason, 131
unpack_wds_SLQSGetDUNCallInfo_t, 1177
voice_arrCallEndReason, 1200
CallFWExtInfo
getCallFWExtInfo, 258
voice_getCallFWExtInfo, 1222
callFWExtInfo, 148
noReplyTimer, 150
numLen, 150
numPlan, 150
numType, 150
number, 150
PI, 150
SI, 150
SvcClass, 150
SvcStatus, 150
CallFWInfo
getCallFWInfo, 259
voice_getCallFWInfo, 1223
callFWInfo, 150
noReplyTimer, 151
numLen, 151
number, 151
SvcClass, 151
SvcStatus, 151
callFwdTypeAndPlan, 147
numberPlan, 148
numberType, 148
callID
allCallsAlphaIDInfo, 117
allCallsDiagInfo, 118
allCallsUUSInfo, 118
arrAlertingPattern, 128
arrAlertingType, 129
arrCallEndReason, 131
arrSvcOption, 134
callInfo, 152
DTMFInfo, 230
getAllCallRmtPtyName, 252
getAllCallRmtPtyNum, 252
pack_voice_SLQSVoiceGetCallInfo_t, 667
pack_voice_SLQSVoiceStopContDTMF_t, 678
peerNumberInfo, 705
unpack_voice_SLQSVoiceStopContDTMF_t, 1162
voice_allCallsAlphaIDInfo, 1195
voice_allCallsDiagInfo, 1196
voice_allCallsUUSInfo, 1196
voice_arrAlertingPattern, 1198
voice_arrAlertingType, 1199
voice_arrCallEndReason, 1200
voice_arrSvcOption, 1204
voice_callInfo, 1211
voice_getAllCallRmtPtyName, 1221
voice_getAllCallRmtPtyNum, 1222
voice_peerNumberInfo, 1225
voiceCallInfoReq, 1233
voiceInfoRec, 1262
voiceOTASPStatusInfo, 1265
voicePrivacyInfo, 1266
voiceStopContDTMFinfo, 1278
voiceSUPSNotification, 1282
callInfo, 151
callID, 152
callState, 152
callType, 153
direction, 153
mode, 153
callNumber
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238

callState
callInfo, 152
voice_callInfo, 1211
callType
callInfo, 153
voice_callInfo, 1212
calledPartyInfo, 144
numLen, 145
numPlan, 145
numType, 145
number, 145
PI, 146
SI, 146
CalledPartyNum
arrCalledPartyNum, 130
voice_arrCalledPartyNum, 1200
calledPartySubAdd, 146
extBit, 146
oddEvenInd, 146
subAddr, 146
subAddrLen, 146
subAddrType, 146
callerID
callerIDInfo, 147
connectNumInfo, 187
voice_connectNumInfo, 1217
callerIDInfo, 147
callerID, 147
callerIDLen, 147
PI, 147
callerIDLen
callerIDInfo, 147
connectNumInfo, 187
voice_connectNumInfo, 1217
callerName
remotePartyName, 778
voice_remotePartyName, 1227
Callinfo
getAllCallInformation, 251
voice_getAllCallInformation, 1221
callingPartyInfo, 153
numLen, 154
numPlan, 154
numType, 154
number, 154
PI, 154
SI, 154
CancelUSSD
qaGobiApiVoice.h, 1802
Card Application Toolkit (CAT), 44
cardResult, 154
sw1, 155
sw2, 155
cardState
slotInf, 836
slotInfo, 838
uim_slotInfo, 956
cardStatus, 155
index1xPri, 156
index1xSec, 156
indexGwPri, 156
indexGwSec, 156
numSlot, 156
SlotInfo, 156
Carrier
fwinfo_s, 248
carrier
CurrentImgList, 197
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1015
carrier_str
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
CarrierImage_t, 156
m_FwBuildId, 157
m_FwImageId, 157
m_PriBuildId, 157
m_PriImageId, 157
m_nCarrierId, 157
m_nFolderId, 157
m_nStorage, 157
CatAlPhaIdentifierTlv, 157
AlphaID, 158
AlphaIDLength, 158
ReferenceID, 158
CatAlphaIdtfr
currentCatEvent, 196
CatCommonEventTlv, 158
CatEvent, 158
EventID, 158
EventLength, 158
TlvPresent, 158
CatEndPS
currentCatEvent, 196
CatEndProactiveSessionTlv, 158
EndProactiveSession, 159
CatEvIDData
currentCatEvent, 196
CatEvent
CatCommonEventTlv, 158
CatEventIDDataTlv, 159
Data, 159
DataLength, 159
ReferenceID, 159
CatEventListTlv, 159
SetupEventList, 160
CatEventLst
currentCatEvent, 196
CatRefresh
currentCatEvent, 196
CatRefreshTlv, 160
RefreshMode, 160
RefreshStage, 160
causeCode
SMSAsyncRawSend_s, 869
ccSUPSType, 160
reason, 161

svcType, 161
ccsSupported
CDMASysInfo, 173
nas_CDMASysInfo, 418
ccsSupportedValid
CDMASysInfo, 173
nas_CDMASysInfo, 418
cdmaMsgDecodingParams, 164
absoluteValidity, 166
mcTimeStamp, 166
messageLength, 166
pAlertPriority, 166
pCallbkAddr, 166
pCallbkAddrLength, 166
pDisplayMode, 166
pLanguage, 166
pMessage, 167
pMessageID, 167
pPriority, 167
pPrivacy, 167
pReadAcknowledgementReq, 167
pRelativeValidity, 167
pSenderAddr, 167
pSenderAddrLength, 167
pTextMsg, 167
pTextMsgLength, 167
pUserAcknowledgementReq, 167
cdmaMsgEncodingParams, 167
messageId, 168
pCallbackAddr, 168
pDestAddr, 168
pEncodingAlphabet, 168
pMessage, 168
pMessageSize, 168
pPriority, 168
pRelValidity, 168
pTextMsg, 169
textMsgLength, 169
cdmaSSInfo, 170
ecio, 170
rssi, 170
cdmaSysIdValid
CDMASysInfo, 173
nas_CDMASysInfo, 418
cell_resel_priority
infoInterFreq, 324
nas_infoInterFreq, 446
cellBroadcastCap
AddSysInfo, 116
nas_AddSysInfo, 410
CellDb, 174
mask, 175
CellID
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
cellID
GERANInfo, 249
nas_GERANInfo, 434
nas_UMTSInfo, 511
UMTSInfo, 981
cellId
GSMSysInfo, 293
LTESysInfo, 401
nas_GSMSysInfo, 439
nas_LTESysInfo, 464
nas_WCDMASysInfo, 520
WCDMASysInfo, 1291
cellIdValid
gsmCellInfo, 289
GSMSysInfo, 293
LTESysInfo, 401
nas_gsmCellInfo, 436
nas_GSMSysInfo, 439
nas_LTESysInfo, 464
nas_WCDMASysInfo, 520
WCDMASysInfo, 1292
cellInterFreqParams
infoInterFreq, 324
nas_infoInterFreq, 446
cellIsTDD
nas_umtsLTENbrCell, 513
umtsLTENbrCell, 983
CellParams
LTEInfoIntrafreq, 389
nas_LTEInfoIntrafreq, 454
cellParams, 175
pci, 176
rsrp, 176
rsrq, 176
rssi, 176
srxlev, 176
cellReselPriority
lteGsmCellInfo, 385
LTEInfoIntrafreq, 389
lteWcdmaCellInfo, 402
nas_lteGsmCellInfo, 449
nas_LTEInfoIntrafreq, 454
nas_lteWcdmaCellInfo, 465
cells_len
infoInterFreq, 324
lteGsmCellInfo, 385
nas_infoInterFreq, 446
nas_lteGsmCellInfo, 449
cellsLen
LTEInfoIntrafreq, 389
lteWcdmaCellInfo, 402
nas_LTEInfoIntrafreq, 454
nas_lteWcdmaCellInfo, 465
chaddr
wds_DHCPv4HWConfig, 1296
WdsDHCPv4HWConfig, 1313
wdsDhcpv4HwConfig, 1312
chaddrLen
wds_DHCPv4HWConfig, 1296
WdsDHCPv4HWConfig, 1313
wdsDhcpv4HwConfig, 1312

changePIN
pack_uim_ChangePin_t, 649
UIMChangePinReq, 960
changeUIMPIN, 176
oldPINLen, 177
oldPINVal, 177
pinID, 177
pinLen, 177
pinVal, 177
ChannelRate, 177
CurrChanRxRate, 178
CurrChanTxRate, 178
DUNCallInfoInd, 231
MaxChanRxRate, 178
MaxChanTxRate, 178
channelRate, 177
CurrChanRxRate, 177
CurrChanTxRate, 177
unpack_wds_SLQSGetDUNCallInfo_t, 1177
Chipset
DeviceConfigDetail, 219
nas_DeviceConfigDetail, 429
ckLen
depersonalizationInformation, 217
uim_depersonalizationInformation, 942
ckVal
depersonalizationInformation, 217
uim_depersonalizationInformation, 942
clear
pack_dms_SLQSSwiGetCrashInfo_t, 582
ClkInfo, 180
mask, 181
codingScheme
nas_PLMNNetworkNameData, 483
PLMNNetworkNameData, 716
remotePartyName, 778
voice_remotePartyName, 1228
CommInfo, 183
imsRegState, 184
modemMode, 184
psState, 184
systemMode, 184
temperature, 184
common.h
eCAT, 1328
eCTL, 1328
eDMS, 1328
eIND, 1329
eLOC, 1328
eLOG_DEBUG, 1327
eLOG_FATAL, 1328
eLOG_INFO, 1327
eLOG_WARN, 1327
eNAS, 1328
ePDS, 1328
eQOS, 1328
eREQ, 1329
eRMS, 1328
eRSP, 1329
eSAR, 1328
eSMS, 1328
eSWILOC, 1328
eSWIM2MCMD, 1328
eSWIOMA, 1328
eSWIOMAEXT, 1328
eTIMEOUT_10_S, 1328
eTIMEOUT_20_S, 1328
eTIMEOUT_2_S, 1328
eTIMEOUT_300_S, 1328
eTIMEOUT_30_S, 1328
eTIMEOUT_5_S, 1328
eTIMEOUT_60_S, 1328
eTIMEOUT_8_S, 1328
eTIMEOUT_DEFAULT, 1328
eTMD, 1328
eUIM, 1328
eVOICE, 1328
eWDS, 1328
common.h, 1325
eLOG_LEVEL, 1327
eQMI_SVC, 1328
eTimeout, 1328
fill_pack_ctx, 1329
fill_sdu_hdr, 1329
get_version, 1329
glog, 1330
gloglvl, 1330
helper_get_resp_ctx, 1329
helper_get_xid, 1329
helper_set_log_func, 1329
helper_set_log_lvl, 1329
liteqmi_GetVersion, 1329
liteqmi_log, 1329
logger, 1327
MINREQBKLEN, 1327
MSGID_AND_LEN, 1327
MSGID_DONT_CARE, 1327
msgtype, 1328
SDU_HDR_LEN, 1327
UNUSEDPARAM, 1327
unpack_result_code_only, 1330
commonInfo
swiModemStatusResp, 897
unpack_nas_SLQSNasSwiModemStatus_t, 1076
ConcSvcInfo
unpack_nas_SLQSGetServingSystem_t, 1061
concSvcInfo
qaQmiServingSystemParam, 738
confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 603
Config
_SwiAvmsEventInd, 99
conn_status
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1181
ConnRateElmntsV4

WdsConnectionRate, 1309
ConnRateElmntsV6
WdsConnectionRate, 1309
connStatus
_packetSrvStatus, 70
connectNumInfo, 185
callerID, 187
callerIDLen, 187
numPlan, 187
numPresInd, 187
numType, 187
screeningInd, 187
ConnectedPartyNum
arrConnectPartyNum, 132
voice_arrConnectPartyNum, 1201
ConnectionRequest
_SwiAvmsEventInd, 99
ConnectionStatus, 184
MDMCallDuration, 185
MDMConnStatus, 185
connectionStatus, 185
MDMCallDuration, 185
MDMConnStatus, 185
unpack_wds_GetSessionState_t, 1172
unpack_wds_SLQSGetDUNCallInfo_t, 1177
connetionState
imsaPdpStatusInfo, 310
content
authenticateResult, 136
readResult, 774
uim_authenticateResult, 937
uim_readResult, 949
contentLen
authenticateResult, 136
readResult, 774
uim_authenticateResult, 937
uim_readResult, 949
context
authenticationData, 136
uim_authenticationData, 938
contextId
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 698
swiPDPRuntimeSettingsReq, 899
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1186
contextType
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 698
swiPDPRuntimeSettingsReq, 899
ControlMac
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733
Count1
RankIndicatorInd, 773
Count2
RankIndicatorInd, 773
countryInitials
nas_PLMNNetworkNameData, 483
PLMNNetworkNameData, 716
coverage
altitudeSrcInfo, 120
altSrcInfo_t, 121
cpich_ecno
nas_wcdmaCellInfo, 515
wcdmaCellInfo, 1282
cpich_rscp
nas_wcdmaCellInfo, 515
wcdmaCellInfo, 1282
cradleMountConfigStatus
QmiCbkLocCradleMountInd, 746
crashAction
pack_dms_SetCrashAction_t, 579
crashData
CrashInfo, 188
crashInformation, 190
crashId
CrashInfo, 188
crashInformation, 190
CrashInfo, 187
crashData, 188
crashId, 188
crashStrLen, 188
gcDumpStrLen, 188
numCrashes, 188
pCrashString, 188
pGCDumpString, 188
crashInfo
crashInfoParams, 189
crashInfoParam
unpack_dms_SLQSSwiGetCrashInfo_t, 1013
CrashInfoParams, 189
pCrashInfo, 189
pDevCrashStatus, 189
crashInfoParams, 188
crashInfo, 189
crashStatus, 189
crashInformation, 189
crashData, 190
crashId, 190
crashString, 190
crashStrlen, 190
gcdumpString, 190
gcdumpStrlen, 190
numCrashes, 190
crashStatus
crashInfoParams, 189
crashStrLen
CrashInfo, 188
crashString
crashInformation, 190
crashStrlen
crashInformation, 190
CreateProfileIn, 190
curProfile, 191
pProfileID, 191

pProfileType, 191
CreateProfileOut, 191
pExtErrorCode, 192
pProfileIndex, 192
pProfileType, 192
csAttachState
nas_servSystem, 498
NASServingSystemInfo, 551
ServingSystemInfo, 803
servSystem, 805
csBarStatus
CallBarringSysInfo, 143
callBarStatus, 144
nas_CallBarringSysInfo, 411
nas_callBarStatus, 412
CsgId, 192
csgId, 192
TlvPresent, 192
csgId
CsgId, 192
cur_carr_name
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
cur_carr_rev
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
curAMRConfig, 194
gsmAmrStat, 195
wcdmaAmrStat, 195
curDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 1176
curProfile
_GetProfileSettingOut, 65
CreateProfileIn, 191
ModifyProfileIn, 407
pack_wds_SLQSModifyProfile_t, 690
UnPackGetProfileSettingOut, 1187
CurrChanRxRate
ChannelRate, 178
channelRate, 177
dunchannelRate, 232
CurrChanTxRate
ChannelRate, 178
channelRate, 177
dunchannelRate, 232
currDBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1182
CurrDataSysStat, 195
pCurrNetworkInfo, 196
pNetworkInfoLen, 196
pPrefNetwork, 196
CurrImageInfo, 198
buildID, 199
buildIDLen, 199
imageType, 199
uniqueID, 199
currNWInfo
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1182
CurrNetworkInfo, 199
NetworkType, 200
RATMask, 201
SOMask, 201
currNetworkInfo, 201
NetworkType, 201
RATMask, 201
SOMask, 201
unpack_wds_SLQSGetCurrDataSystemStat_t, 1175
current_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1176
WDSSWICurrentChannelRates, 1324
current_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1176
WDSSWICurrentChannelRates, 1324
currentCatEvent, 196
CatAlphaIdtfr, 196
CatEndPS, 196
CatEvIDData, 196
CatEventLst, 196
CatRefresh, 196
currentChannelRXRate
unpack_wds_GetConnectionRate_t, 1164
currentChannelTXRate
unpack_wds_GetConnectionRate_t, 1164
currentDataBearer
pack_wds_SLQSSetWdsEventCallback_t, 692
CurrentImgList, 196
carrier, 197
fwvers, 197
numEntries, 197
pCurrImgInfo, 197
pkgver, 197
priver, 197
currentMitigationLvl
QmiCbkTmdMitiLvlRptInd, 759
currentNetwork
dataBearerTechnology, 211
qmiWDSDataBearerTechnology, 763
CurrentPLMN
qaQmiServingSystemParam, 738
unpack_nas_SLQSGetServingSystem_t, 1061
currentPLMN, 197
MCC, 198
MNC, 198
netDescr, 198
netDescrLength, 198
cust_attr
custSettingInfo, 206
DMScustSettingInfo, 224
cust_id
custSettingInfo, 206

DMScustSettingInfo, 224
DMSgetCustomInput, 226
getCustomInput, 260
pack_dms_GetCustFeaturesV2_t, 578
pack_dms_SetCustFeaturesV2_t, 581
setCustomSettingV2, 812
cust_value
custSettingInfo, 206
DMScustSettingInfo, 224
pack_dms_SetCustFeaturesV2_t, 581
setCustomSettingV2, 812
custFeaturesInfo, 201
GpsEnable, 203
pDHCPRelayEnabled, 203
pDisableIMSI, 203
pGPSLPM, 203
pGPSSel, 203
pIPFamSupport, 203
pIsVoiceEnabled, 203
pRMAutoConnect, 203
pSMSSupport, 203
qaGobiApiDms.h, 1548
custFeaturesSetting, 204
pDHCPRelayEnabled, 205
pGPSEnable, 205
pGPSLPM, 205
pGPSSel, 205
pIsVoiceEnabled, 205
qaGobiApiDms.h, 1550
custSetting
custSettingList, 206
DMScustSettingList, 225
custSettingInfo, 205
cust_attr, 206
cust_id, 206
cust_value, 206
id_length, 206
value_length, 206
custSettingList, 206
custSetting, 206
list_type, 206
num_instances, 207
CustomSCP
unpack_nas_GetCDMANetworkParameters_t, 1046
CwtMute
GetM2MAudioProfileResp, 273
GetM2MAVMuteResp, 275

DEVICE_STATE_BOOT
qaGobiApiCbk.h, 1501
DEVICE_STATE_DISCONNECTED
qaGobiApiCbk.h, 1501
DEVICE_STATE_READY
qaGobiApiCbk.h, 1501
DEVICE_STATE_SERIAL_ONLY
qaGobiApiCbk.h, 1501
dBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1182
dBTechExtRatValue
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
dBTechExtSoMask
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
dBTechnology
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
dBtechExtAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
dBtechnologyExt
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
DEFAULTBYTEVALUE
qaGobiApiPds.h, 1670
DEFAULTLONGVALUE
qaGobiApiPds.h, 1670
DEFAULTWORDVALUE
qaGobiApiPds.h, 1670
DEREGISTER_EVENT
qaGobiApiCbk.h, 1461
DEREGISTER_SRV
qaGobiApiCbk.h, 1461
DEVICE_OFFLINE
qaGobiApiFms.h, 1588
DEVICE_RESET
qaGobiApiFms.h, 1588
DEVICE_SHUTDOWN
qaGobiApiFms.h, 1588
DHCPOption, 220
optCode, 220
optValLen, 220
pOptVal, 220
DHCPOptionList, 220
numOpt, 221
pOptions, 221
DHCPRelayEnabled
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
DHCPv4LeaseOptTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1162
DHCPv4LeaseStateTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1162
DMS_IMGDETAILS_LEN
dms.h, 1336
DMS_PM_FACTORY
dms.h, 1336
DMS_PM_LOW
dms.h, 1336
DMS_PM_OFFLINE
dms.h, 1336
DMS_PM_ONLINE
dms.h, 1336
DMS_PM_RESET

dms.h, 1336
DMS_PM_SHUT_DOWN
dms.h, 1336
DMScustSettingInfo, 224
cust_attr, 224
cust_id, 224
cust_value, 224
id_length, 224
value_length, 224
DMScustSettingList, 225
custSetting, 225
list_type, 225
num_instances, 225
DMSgetCustomFeatureV2, 225
pCustSettingInfo, 226
pCustSettingList, 226
pGetCustomInput, 226
DMSgetCustomInput, 226
cust_id, 226
list_type, 226
DRCCover
DRCParams, 229
nas_DRCParams, 430
DRCParams, 229
DRCCover, 229
DRCValue, 229
DRCValue
DRCParams, 229
nas_DRCParams, 430
DTMFEvent
DTMFInfo, 230
DTMFInfo, 229
callID, 230
DTMFEvent, 230
digitBuff, 230
digitCnt, 230
DTMFInformation
voiceDTMFEventInfo, 1240
DTMFInterdigitInterval
DTMFLengths, 231
voice_DTMFLengths, 1220
DTMFLengths, 230
DTMFInterdigitInterval, 231
DTMFPulseWidth, 231
DTMFPulseWidth
DTMFLengths, 231
voice_DTMFLengths, 1220
DTMFdigit
pack_voice_SLQSVoiceStartContDTMF_t, 678
voiceContDTMFinfo, 1239
DTMInd
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1061
DUNCallInfoInd, 231
CallEndReason, 231
ChannelRate, 231
DataBearerTech, 231
DormancyStatus, 231
MdmConnStatus, 231
RXOKBytesCount, 231
TXOKBytesCount, 231
Data
CatEventIDDataTlv, 159
data
authenticationData, 137
SMSCAddress, 870
sMSCAddress, 870
SMSEtwsMessage, 872
sMSEtwsMessage, 871
SMSTransferRouteMTMessage, 882
sMSTransferRouteMTMessage, 883
SwiOTAMsg_s, 898
uim_authenticationData, 938
data_buf
NASOTAMessageTlv, 541
data_len
NASOTAMessageTlv, 541
SwiOTAMsg_s, 898
dataBearer
pack_wds_SLQSSetWdsEventCallback_t, 692
dataBearerMask
dataBearers, 207
unpack_wds_SLQSGetDataBearerTechnology_t, 1176
DataBearerTech, 207
DUNCallInfoInd, 231
ratValue, 209
soMask, 209
techType, 209
dataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1177
DataBearerTechExt, 209
pBearerTech, 210
pLastBearerTech, 210
dataBearerTechExt
pack_wds_SLQSSetWdsEventCallback_t, 693
dataBearerTechnology, 210
currentNetwork, 211
ratMask, 211
soMask, 211
dataBearers, 207
dataBearerMask, 207
pCurDataBearerTechnology, 207
pLastCallDataBearerTechnology, 207
dataCapabilities
dataSrvCapabilities, 212
nas_dataSrvCapabilities, 424
dataCapabilitiesLen
dataSrvCapabilities, 212
nas_dataSrvCapabilities, 424
DataCaps
unpack_nas_GetServingNetwork_t, 1051
unpack_nas_GetServingNetworkCapabilities_t, 1051
dataCaps

unpack_nas_SetDataCapabilitiesCallback_ind_t, 1053
DataCapsLen
unpack_nas_GetServingNetwork_t, 1051
unpack_nas_GetServingNetworkCapabilities_t, 1051
dataCapsSize
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1053
dataLen
authenticationData, 137
uim_authenticationData, 938
DataLength
CatEventIDDataTlv, 159
DataRate
unpack_qos_swiQosFlow_t, 1107
dataRate, 211
dataRateMax, 212
guaranteedRate, 212
dataRateMax
dataRate, 212
unpack_qos_dataRate_t, 1090
dataServiceCaCapability
unpack_dms_GetDeviceCapabilities_t, 995
DataServiceCapability
unpack_dms_GetDeviceCap_t, 994
DataSessionStatus
_SwiAvmsEventInd, 99
DataSrvCapabilities
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1061
dataSrvCapabilities, 212
dataCapabilities, 212
dataCapabilitiesLen, 212
DataStatusDetail, 213
IPAddress, 214
LastErrCode, 214
dataSysStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1182
dataSystemStatus
pack_wds_SLQSSetWdsEventCallback_t, 693
DataULongLongTlv, 215
TlvPresent, 215
ullData, 215
DataULongTlv, 215
TlvPresent, 215
ulData, 215
Date
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
wcdmaLongMsgDecodingParams, 1285
wcdmaMsgDecodingParams, 1287
DateLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
day
nas_timeInfo, 508
nas_UniversalTime, 514
timeInfo, 922
UniversalTime, 991
dayLtSavingAdj
nas_timeInfo, 508
timeInfo, 922
dayOfWeek
nas_timeInfo, 508
nas_UniversalTime, 514
timeInfo, 922
UniversalTime, 991
daylightSavings
nas_qaQmi3Gpp2TimeZone, 485
qaQmi3Gpp2TimeZone, 734
DcsUsbPortNames, 215
AtCmdPort, 215
DmPort, 215
NmeaPort, 215
defaultPDNEnabled
unpack_wds_SLQSGet3GPPConfigItem_t, 1174
DefaultRoamInd
unpack_nas_SLQSGetServingSystem_t, 1061
defaultRoamInd
qaQmiServingSystemParam, 739
delAssistDataStatus, 215
status, 216
delayClass
GPRSQoS, 283
GPRSRequestedQoS, 284
LibPackGPRSRequestedQoS, 329
wds_GPRSQoS, 1300
DeleteStoredImage
qaGobiApiFms.h, 1591
deliveryErrSDU
LibPackUMTSQoS, 356
UMTSMinQoS, 985
UMTSQoS, 988
wds_UMTSMinQoS, 1306
depersonalizationInformation, 216
ckLen, 217
ckVal, 217
feature, 217
operation, 217
depersonilisationInfo
pack_uim_SLQSUIMDepersonalization_t, 652
UIMDepersonalizationReq, 961
Description
SlqsNas3GppNetworkInfo, 842
description
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
unpack_omaDmFotaTlv_t, 1087
descriptionlength
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
unpack_omaDmFotaTlv_t, 1087
Desription
nas_QmiNas3GppNetworkInfo, 486

destPortRangeEnd
LibPackTFTIDParams, 353
TFTIDParams, 920
destPortRangeStart
LibPackTFTIDParams, 354
TFTIDParams, 920
detailSvcInfo, 217
hdrHybrid, 218
hdrSrvStatus, 218
isSysForbidden, 218
srvCapability, 218
srvStatus, 219
DetailedSvcInfo
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1061
dev
qmifwinfo_s, 760
DevCrashState
unpack_dms_GetCrashAction_t, 992
Device
SetM2MAudioAVCFGReq, 818
Device Connectivity Service (DCS), 37
Device Management Service (DMS), 39
device_state_enum
qaGobiApiCbk.h, 1501
DeviceConfigDetail, 219
Chipset, 219
HWVersion, 219
QLIC, 220
Technology, 220
deviceID
_MitigationDevInfo, 67
deviceIdLen
_MitigationDevInfo, 67
DiagInfo
allCallsDiagInfo, 118
arrDiagInfo, 132
voice_allCallsDiagInfo, 1196
voice_arrDiagInfo, 1202
diagInfo, 221
diagInfoLen, 221
diagnosticInfo, 221
diagInfoLen
diagInfo, 221
voice_diagInfo, 1219
diagnosticInfo
diagInfo, 221
voice_diagInfo, 1219
digitBuff
DTMFInfo, 230
digitCnt
burstDTMFInfo, 142
DTMFInfo, 230
voice_burstDTMFInfo, 1205
dirNum, 221
dirNum, 222
dirNumLen, 222
dirNum, 222
nas_dirNum, 429
dirNumLen
dirNum, 222
nas_dirNum, 429
direction
callInfo, 153
voice_callInfo, 1212
DisableIMSI
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
dispType
extDispRecInfo, 238
displayCondition
nas_serviceProviderName, 496
serviceProviderName, 802
dl_bw_value
nas_PhyCaAggPcellInfo, 475
nas_PhyCaAggScellDlBw, 476
nas_PhyCaAggScellInfo, 480
NASPhyCaAggPcellInfo, 542
NASPhyCaAggScellDlBw, 542
NASPhyCaAggScellInfo, 545
PhyCaAggPcellInfo, 707
PhyCaAggScellDlBw, 708
PhyCaAggScellInfo, 712
DmPort
DcsUsbPortNames, 215
dms.h, 1330
ACT_CODE_MAX_SIZE, 1336
CK_MAX_SIZE, 1336
DMS_IMGDETAILS_LEN, 1336
DMS_PM_FACTORY, 1336
DMS_PM_LOW, 1336
DMS_PM_OFFLINE, 1336
DMS_PM_ONLINE, 1336
DMS_PM_RESET, 1336
DMS_PM_SHUT_DOWN, 1336
ERI_DATA_MAX_SIZE, 1337
MAX_BUILD_ID_LEN, 1337
MEID_MAX_SIZE, 1337
pack_dms_ActivateAutomatic, 1337
pack_dms_GetActivationState, 1338
pack_dms_GetBandCapability, 1338
pack_dms_GetCrashAction, 1338
pack_dms_GetCustFeature, 1339
pack_dms_GetCustFeaturesV2, 1339
pack_dms_GetDeviceCap, 1339
pack_dms_GetDeviceCapabilities, 1339
pack_dms_GetDeviceHardwareRev, 1340
pack_dms_GetDeviceMfr, 1340
pack_dms_GetDeviceSerialNumbers, 1340
pack_dms_GetFSN, 1342
pack_dms_GetFirmwareInfo, 1341
pack_dms_GetFirmwareRevision, 1341
pack_dms_GetFirmwareRevisions, 1342
pack_dms_GetHardwareRevision, 1342
pack_dms_GetIMSI, 1343
pack_dms_GetManufacturer, 1343

pack_dms_GetModelID, 1343
pack_dms_GetNetworkTime, 1344
pack_dms_GetOfflineReason, 1344
pack_dms_GetPRLVersion, 1345
pack_dms_GetPower, 1344
pack_dms_GetSerialNumbers, 1345
pack_dms_GetUSBComp, 1345
pack_dms_GetVoiceNumber, 1346
pack_dms_ResetToFactoryDefaults, 1346
pack_dms_SLQSDmsSwiGetResetInfo, 1349
pack_dms_SLQSDmsSwiIndicationRegister, 1350
pack_dms_SLQSGetBandCapability, 1350
pack_dms_SLQSGetERIFile, 1351
pack_dms_SLQSSwiClearDyingGaspStatistics, 1351
pack_dms_SLQSSwiGetCrashInfo, 1351
pack_dms_SLQSSwiGetDyingGaspCfg, 1352
pack_dms_SLQSSwiGetDyingGaspStatistics, 1352
pack_dms_SLQSSwiGetFirmwareCurr, 1352
pack_dms_SLQSSwiGetFwUpdateStatus, 1353
pack_dms_SLQSSwiGetHostDevInfo, 1353
pack_dms_SLQSSwiGetOSInfo, 1353
pack_dms_SLQSSwiGetSerialNoExt, 1354
pack_dms_SLQSSwiSetDyingGaspCfg, 1354
pack_dms_SLQSSwiSetHostDevInfo, 1354
pack_dms_SLQSSwiSetOSInfo, 1355
pack_dms_SLQSUIMGetState, 1355
pack_dms_SetActivationStatusCallback, 1347
pack_dms_SetCrashAction, 1347
pack_dms_SetCustFeature, 1347
pack_dms_SetCustFeaturesV2, 1348
pack_dms_SetEventReport, 1348
pack_dms_SetFirmwarePreference, 1348
pack_dms_SetPower, 1349
pack_dms_SetUSBComp, 1349
pack_dms_UIMChangePIN, 1355
pack_dms_UIMGetControlKeyStatus, 1356
pack_dms_UIMGetICCID, 1356
pack_dms_UIMGetPINStatus, 1357
pack_dms_UIMSetControlKeyProtection, 1357
pack_dms_UIMSetPINProtection, 1357
pack_dms_UIMUnblockControlKey, 1358
pack_dms_UIMUnblockPIN, 1358
pack_dms_UIMVerifyPIN, 1358
pack_dms_ValidateSPC, 1359
SPC_SIZE, 1337
UNIQUE_ID_LEN, 1337
unpack_dms_ActivateAutomatic, 1359
unpack_dms_GetActivationState, 1359
unpack_dms_GetBandCapability, 1360
unpack_dms_GetCrashAction, 1360
unpack_dms_GetCustFeature, 1361
unpack_dms_GetCustFeaturesV2, 1361
unpack_dms_GetDeviceCap, 1361
unpack_dms_GetDeviceCapabilities, 1361
unpack_dms_GetDeviceHardwareRev, 1362
unpack_dms_GetDeviceMfr, 1362
unpack_dms_GetDeviceSerialNumbers, 1362
unpack_dms_GetFSN, 1364
unpack_dms_GetFirmwareInfo, 1363
unpack_dms_GetFirmwareRevision, 1363
unpack_dms_GetFirmwareRevisions, 1363
unpack_dms_GetHardwareRevision, 1364
unpack_dms_GetIMSI, 1364
unpack_dms_GetManufacturer, 1365
unpack_dms_GetModelID, 1365
unpack_dms_GetNetworkTime, 1365
unpack_dms_GetOfflineReason, 1366
unpack_dms_GetPRLVersion, 1366
unpack_dms_GetPower, 1366
unpack_dms_GetSerialNumbers, 1367
unpack_dms_GetUSBComp, 1367
unpack_dms_GetVoiceNumber, 1367
unpack_dms_ResetToFactoryDefaults, 1368
unpack_dms_SLQSDmsSwiGetResetInfo, 1371
unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 1371
unpack_dms_SLQSDmsSwiIndicationRegister, 1372
unpack_dms_SLQSGetBandCapability, 1372
unpack_dms_SLQSGetERIFile, 1373
unpack_dms_SLQSSwiClearDyingGaspStatistics, 1373
unpack_dms_SLQSSwiGetCrashInfo, 1373
unpack_dms_SLQSSwiGetDyingGaspCfg, 1374
unpack_dms_SLQSSwiGetDyingGaspStatistics, 1374
unpack_dms_SLQSSwiGetFirmwareCurr, 1374
unpack_dms_SLQSSwiGetFwUpdateStatus, 1375
unpack_dms_SLQSSwiGetHostDevInfo, 1375
unpack_dms_SLQSSwiGetOSInfo, 1375
unpack_dms_SLQSSwiGetSerialNoExt, 1376
unpack_dms_SLQSSwiSetDyingGaspCfg, 1376
unpack_dms_SLQSSwiSetHostDevInfo, 1376
unpack_dms_SLQSSwiSetOSInfo, 1377
unpack_dms_SLQSUIMGetState, 1377
unpack_dms_SetActivationStatusCallback, 1368
unpack_dms_SetCrashAction, 1368
unpack_dms_SetCustFeature, 1369
unpack_dms_SetCustFeaturesV2, 1369
unpack_dms_SetEventReport, 1369
unpack_dms_SetEventReport_ind, 1370
unpack_dms_SetFirmwarePreference, 1370
unpack_dms_SetPower, 1370
unpack_dms_SetUSBComp, 1371
unpack_dms_UIMChangePIN, 1377
unpack_dms_UIMGetControlKeyStatus, 1378
unpack_dms_UIMGetICCID, 1378
unpack_dms_UIMGetPINStatus, 1378
unpack_dms_UIMSetControlKeyProtection, 1379
unpack_dms_UIMSetPINProtection, 1379
unpack_dms_UIMUnblockControlKey, 1379
unpack_dms_UIMUnblockPIN, 1380
unpack_dms_UIMVerifyPIN, 1380
unpack_dms_ValidateSPC, 1380

dms_ActivationStatusTlv, 222
activationStatus, 222
TlvPresent, 222
dms_OperatingModeTlv, 223
operatingMode, 223
TlvPresent, 223
dmsCurrentPRLInfo, 223
pPRLPreference, 224
pPRLVersion, 224
qaGobiApiDms.h, 1551
dmsIndicationRegisterReq, 226
pSwiGetResetInd, 227
dmsSwiGetResetInfo, 227
source, 227
type, 228
Domain, 228
domainLen, 228
domainName, 228
domain
DomainNameList, 228
wds_DomainNameList, 1299
domainLen
Domain, 228
wds_Domain, 1299
DomainList
unpack_wds_SLQSGetRuntimeSettings_t, 1179
domainName
Domain, 228
wds_Domain, 1299
DomainNameList, 228
domain, 228
numInstances, 229
dormancyStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
dormancyState
unpack_wds_GetDormancyState_t, 1167
DormancyStatus
DUNCallInfoInd, 231
dormancyStatus
pack_wds_SLQSSetWdsEventCallback_t, 693
unpack_wds_SLQSGetDUNCallInfo_t, 1177
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
downLink
NSSAudioCtrl, 568
DownloadToSlot
qaGobiApiFms.h, 1591
dscp
QosMap, 773
dtmSupp
GSMSysInfo, 293
nas_GSMSysInfo, 439
dtmSuppValid
GSMSysInfo, 293
nas_GSMSysInfo, 439
dunchannelRate, 232
CurrChanRxRate, 232
CurrChanTxRate, 232
MaxChanRxRate, 232
MaxChanTxRate, 232
Duration
pack_nas_SetNetworkPreference_t, 608
unpack_nas_GetNetworkPreference_t, 1049

eCAT
common.h, 1328
eCTL
common.h, 1328
eDMS
common.h, 1328
eGOBI_DEV_SERIES_9X07_9X50
qaGobiApiFms.h, 1589
eGOBI_DEV_SERIES_9X15
qaGobiApiFms.h, 1589
eGOBI_DEV_SERIES_9X30
qaGobiApiFms.h, 1589
eGOBI_DEV_SERIES_G3K
qaGobiApiFms.h, 1589
eGOBI_DEV_SERIES_NON_GOBI
qaGobiApiFms.h, 1589
eGOBI_DEV_SERIES_SIERRA_GOBI
qaGobiApiFms.h, 1589
eGOBI_DEV_SERIES_UNKNOWN
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_3
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_AERIS
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_ALLTEL
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_AMX_TELCEL
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_ATT
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_BELL
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_BHARTI
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_BRASIL_VIVO
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_CHINA_MOBILE
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_CHINA_TELECOM
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_CHINA_UNICOM
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_EMOBILE
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_FACTORY
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_GENERIC
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_GENERIC_CDMA
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_IUSACELL
qaGobiApiFms.h, 1589

eGOBI_IMG_CAR_KDDI
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_KT_FREETEL
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_LEAP
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_METROPCS
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_NETCOM
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_NORF
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_NTT_DOCOMO
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_O2
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_OMH
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_ORANGE
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_RELIANCE1
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_RELIANCE2
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_ROGERS
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_SFR
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_SINGTEL_OPTUS
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_SK_TELCOM1
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_SK_TELCOM2
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_SOFTBANK
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_SPRINT
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_SWISSCOM
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_TATA
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_TELCOM_ITALIA
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_TELCOM_NZ
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_TELEFONICA
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_TELENOR
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_TELIASONERA
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_TELSTRA1
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_TELSTRA2
qaGobiApiFms.h, 1590
eGOBI_IMG_CAR_TELUS
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_TMOBILE
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_US
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_VERIZON
qaGobiApiFms.h, 1589
eGOBI_IMG_CAR_VODAFONE
qaGobiApiFms.h, 1589
eGOBI_IMG_GPS_ASSISTED
qaGobiApiFms.h, 1590
eGOBI_IMG_GPS_NO_XTRA
qaGobiApiFms.h, 1590
eGOBI_IMG_GPS_NONE
qaGobiApiFms.h, 1590
eGOBI_IMG_GPS_STAND_ALONE
qaGobiApiFms.h, 1590
eGOBI_IMG_REG_ASIA
qaGobiApiFms.h, 1590
eGOBI_IMG_REG_AUS
qaGobiApiFms.h, 1590
eGOBI_IMG_REG_EU
qaGobiApiFms.h, 1590
eGOBI_IMG_REG_GLOBAL
qaGobiApiFms.h, 1590
eGOBI_IMG_REG_LA
qaGobiApiFms.h, 1590
eGOBI_IMG_REG_NA
qaGobiApiFms.h, 1590
eGOBI_IMG_TECH_CDMA
qaGobiApiFms.h, 1591
eGOBI_IMG_TECH_UMTS
qaGobiApiFms.h, 1591
eGobi_DEV_SERIES_MC83
qaGobiApiFms.h, 1589
eIND
common.h, 1329
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
nas.h, 1407
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
nas.h, 1407
eLOC

common.h, 1328
eLOG_DEBUG
common.h, 1327
eLOG_FATAL
common.h, 1328
eLOG_INFO
common.h, 1327
eLOG_WARN
common.h, 1327
eNAS
common.h, 1328
eNAS_LTE_CPHY_CA_BW_NRB_100
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_CA_BW_NRB_15
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_CA_BW_NRB_25
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_CA_BW_NRB_50
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_CA_BW_NRB_6
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_CA_BW_NRB_75
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
nas.h, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
nas.h, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
nas.h, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
nas.h, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
nas.h, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
nas.h, 1407
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE
nas.h, 1408
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
nas.h, 1408
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
qaGobiApiNas.h, 1637
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
nas.h, 1408
eNAS_RADIO_IF_GSM
qaGobiApiNas.h, 1636
eNAS_RADIO_IF_LTE
qaGobiApiNas.h, 1636
eNAS_RADIO_IF_TDSCDMA
qaGobiApiNas.h, 1636
eNAS_RADIO_IF_UMTS
qaGobiApiNas.h, 1636
ePDS
common.h, 1328
eQA_QMI_SVC_NA
qaGobiApiCbk.h, 1501
eQA_QMI_SVC_NAS
qaGobiApiCbk.h, 1501
eQA_QMI_SVC_WDS
qaGobiApiCbk.h, 1501
eQCWWAN_ERR_API_MUTEX_TIMEOUT
qmerrno.h, 1865
eQCWWAN_ERR_BUFFER_SZ
qmerrno.h, 1864
eQCWWAN_ERR_CANCEL_OP
qmerrno.h, 1865
eQCWWAN_ERR_DRIVER
qmerrno.h, 1865
eQCWWAN_ERR_ENUM_BEGIN
qmerrno.h, 1864
eQCWWAN_ERR_ENUM_END
qmerrno.h, 1865
eQCWWAN_ERR_FILE_COPY
qmerrno.h, 1864
eQCWWAN_ERR_FILE_OPEN
qmerrno.h, 1864
eQCWWAN_ERR_GENERAL
qmerrno.h, 1864
eQCWWAN_ERR_INTERNAL
qmerrno.h, 1864
eQCWWAN_ERR_INVALID_ARG
qmerrno.h, 1864
eQCWWAN_ERR_INVALID_DEVID
qmerrno.h, 1864
eQCWWAN_ERR_INVALID_FILE
qmerrno.h, 1864
eQCWWAN_ERR_INVALID_QMI_RSP
qmerrno.h, 1864
eQCWWAN_ERR_INVALID_XID
qmerrno.h, 1865
eQCWWAN_ERR_MALFORMED_QMI_RSP
qmerrno.h, 1864
eQCWWAN_ERR_MEMORY
qmerrno.h, 1864
eQCWWAN_ERR_MULTIPLE_DEVICES
qmerrno.h, 1865
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
qmerrno.h, 1865
eQCWWAN_ERR_NO_CANCELABLE_OP
qmerrno.h, 1865
eQCWWAN_ERR_NO_CONNECTION
qmerrno.h, 1864
eQCWWAN_ERR_NO_DEVICE
qmerrno.h, 1864
eQCWWAN_ERR_NO_SIGNAL
qmerrno.h, 1865
eQCWWAN_ERR_NONE
qmerrno.h, 1864

eQCWWAN_ERR_NULL_TLV
qmerrno.h, 1868
eQCWWAN_ERR_OFFLINE
qmerrno.h, 1865
eQCWWAN_ERR_PDU_GENERATION
qmerrno.h, 1865
eQCWWAN_ERR_QMI_ABORTED
qmerrno.h, 1865
eQCWWAN_ERR_QMI_ACCESS_DENIED
qmerrno.h, 1867
eQCWWAN_ERR_QMI_ACK_NOT_SENT
qmerrno.h, 1867
eQCWWAN_ERR_QMI_ARG_TOO_LONG
qmerrno.h, 1865
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
qmerrno.h, 1866
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
qmerrno.h, 1866
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
qmerrno.h, 1867
eQCWWAN_ERR_QMI_CALL_FAILED
qmerrno.h, 1865
eQCWWAN_ERR_QMI_CARD_BUSY_RSP
qmerrno.h, 1868
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
qmerrno.h, 1867
eQCWWAN_ERR_QMI_CAT_END
qmerrno.h, 1868
eQCWWAN_ERR_QMI_CAT_START
qmerrno.h, 1868
eQCWWAN_ERR_QMI_CAUSE_CODE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
qmerrno.h, 1865
eQCWWAN_ERR_QMI_CONNECT
qmerrno.h, 1864
eQCWWAN_ERR_QMI_DEVICE_IN_USE
qmerrno.h, 1865
eQCWWAN_ERR_QMI_DEVICE_NOT_READY
qmerrno.h, 1866
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
qmerrno.h, 1866
eQCWWAN_ERR_QMI_DISABLED
qmerrno.h, 1866
eQCWWAN_ERR_QMI_ENCODING
qmerrno.h, 1866
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
qmerrno.h, 1868
eQCWWAN_ERR_QMI_EVENT_REG_FAILED
qmerrno.h, 1868
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
qmerrno.h, 1867
eQCWWAN_ERR_QMI_FDN_RESTRICT
qmerrno.h, 1867
eQCWWAN_ERR_QMI_FLOW_SUSPENDED
qmerrno.h, 1866
eQCWWAN_ERR_QMI_GENERAL
qmerrno.h, 1866
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
qmerrno.h, 1867
eQCWWAN_ERR_QMI_IFACE
qmerrno.h, 1864
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
qmerrno.h, 1867
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INCORRECT_PIN
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
qmerrno.h, 1867
eQCWWAN_ERR_QMI_INJECT_TIMEOUT
qmerrno.h, 1867
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INTERNAL
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_ARG
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
qmerrno.h, 1868
eQCWWAN_ERR_QMI_INVALID_HANDLE
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_ID
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_INDEX
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_OPERATION
qmerrno.h, 1867
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_PINID
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_PROFILE
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_QMI_CMD
qmerrno.h, 1867

eQCWWAN_ERR_QMI_INVALID_QOS_ID
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTIO-N
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_TECH_PREF
qmerrno.h, 1865
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
qmerrno.h, 1868
eQCWWAN_ERR_QMI_INVALID_TRANSITION
qmerrno.h, 1866
eQCWWAN_ERR_QMI_INVALID_TX_ID
qmerrno.h, 1865
eQCWWAN_ERR_QMI_MALFORMED_MSG
qmerrno.h, 1865
eQCWWAN_ERR_QMI_MAX
qmerrno.h, 1867
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_I-N_USE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_-USE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAIL-URE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
qmerrno.h, 1866
eQCWWAN_ERR_QMI_MISSING_ARG
qmerrno.h, 1865
eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, 1867
eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, 1867
eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, 1866
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, 1865
eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, 1866
eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, 1865
eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, 1865
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
qmerrno.h, 1865
eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, 1867
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, 1867
eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, 1865
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, 1865
eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, 1867
eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, 1865
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTE-D
qmerrno.h, 1865
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPOR-TED
qmerrno.h, 1865
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, 1867
eQCWWAN_ERR_QMI_OUT_OF_CALL
qmerrno.h, 1865
eQCWWAN_ERR_QMI_PIN_BLOCKED
qmerrno.h, 1866
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
qmerrno.h, 1866
eQCWWAN_ERR_QMI_POLICY_MISMATCH
qmerrno.h, 1867
eQCWWAN_ERR_QMI_REQ
qmerrno.h, 1864
eQCWWAN_ERR_QMI_REQ_SCH
qmerrno.h, 1864
eQCWWAN_ERR_QMI_REQ_TO
qmerrno.h, 1864
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUP-PORTED
qmerrno.h, 1866
eQCWWAN_ERR_QMI_RSP
qmerrno.h, 1864
eQCWWAN_ERR_QMI_RSP_TO
qmerrno.h, 1864
eQCWWAN_ERR_QMI_SEGMENT_ORDER
qmerrno.h, 1867
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
qmerrno.h, 1867
eQCWWAN_ERR_QMI_SESSION_INACTIVE
qmerrno.h, 1866
eQCWWAN_ERR_QMI_SESSION_INVALID
qmerrno.h, 1866
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
qmerrno.h, 1866
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
qmerrno.h, 1867
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
qmerrno.h, 1866
eQCWWAN_ERR_QMI_SMSC_ADDR
qmerrno.h, 1867
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
qmerrno.h, 1867
eQCWWAN_ERR_QMI_TPDU_TYPE
qmerrno.h, 1867
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACT-ION
qmerrno.h, 1865

eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, 1866
eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, 1868
eQCWWAN_ERR_RESET
qmerrno.h, 1865
eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_END
qmerrno.h, 1868
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_-PROCESS
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_I-D
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_I-D
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPO-RTED
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPO-RTED
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SE-SSIONS
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_START
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, 1867
eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, 1867
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
qmerrno.h, 1868
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUN-D
qmerrno.h, 1868
eQCWWAN_ERR_SWIDCS_END
qmerrno.h, 1868
eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, 1868
eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, 1868
eQCWWAN_ERR_SWIDCS_START
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_END
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWN-LOADED
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD-_MODE
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDE-X
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISM-ATCH
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT-_ACTIVE_IMAGE
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, 1868
eQCWWAN_ERR_SWIIM_START
qmerrno.h, 1868
eQCWWAN_ERR_SWISM_END
qmerrno.h, 1868
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_F-OUND
qmerrno.h, 1868
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, 1868
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, 1868
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPT-ED
qmerrno.h, 1868
eQCWWAN_ERR_SWISMS_START
qmerrno.h, 1868
eQMI_LOC_SESS_STATUS_FAILURE
loc.h, 1389

eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, 1389
eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, 1389
eQMI_LOC_SESS_STATUS_TIMEOUT
loc.h, 1389
eQOS
common.h, 1328
eREQ
common.h, 1329
eRMS
common.h, 1328
eRSP
common.h, 1329
eSAR
common.h, 1328
eSMS
common.h, 1328
eSWILOC
common.h, 1328
eSWIM2MCMD
common.h, 1328
eSWIOMA
common.h, 1328
eSWIOMAEXT
common.h, 1328
eSYS_SRV_DOMAIN_CAMPED
qaGobiApiNas.h, 1636
eSYS_SRV_DOMAIN_CS_ONLY
qaGobiApiNas.h, 1636
eSYS_SRV_DOMAIN_CS_PS
qaGobiApiNas.h, 1636
eSYS_SRV_DOMAIN_NO_SRV
qaGobiApiNas.h, 1636
eSYS_SRV_DOMAIN_PS_ONLY
qaGobiApiNas.h, 1636
eSYS_SRV_DOMAIN_UNKNOWN
qaGobiApiNas.h, 1636
eSetServiceAutomaticTrackingDisable
qaGobiApiPds.h, 1670
eSetServiceAutomaticTrackingEnable
qaGobiApiPds.h, 1670
eTIMEOUT_10_S
common.h, 1328
eTIMEOUT_20_S
common.h, 1328
eTIMEOUT_2_S
common.h, 1328
eTIMEOUT_300_S
common.h, 1328
eTIMEOUT_30_S
common.h, 1328
eTIMEOUT_5_S
common.h, 1328
eTIMEOUT_60_S
common.h, 1328
eTIMEOUT_8_S
common.h, 1328
eTIMEOUT_DEFAULT
common.h, 1328
eTLV_3GPP_NETWORK_INFO
qaNasPerformNetworkScan.h, 1862
eTLV_CBK_ALPHA_IDENTIFIER
qaCbkCatEventReportInd.h, 1446
eTLV_CBK_DISPLAY_TEXT
qaCbkCatEventReportInd.h, 1446
eTLV_CBK_END_PROACTIVE_SESSION
qaCbkCatEventReportInd.h, 1447
eTLV_CBK_GET_IN_KEY
qaCbkCatEventReportInd.h, 1446
eTLV_CBK_GET_INPUT
qaCbkCatEventReportInd.h, 1446
eTLV_CBK_LANGUAGE_NOTIFICATION
qaCbkCatEventReportInd.h, 1447
eTLV_CBK_REFRESH
qaCbkCatEventReportInd.h, 1447
eTLV_CBK_SELECT_ITEM
qaCbkCatEventReportInd.h, 1446
eTLV_CBK_SETUP_EVENT_LIST
qaCbkCatEventReportInd.h, 1447
eTLV_CBK_SETUP_IDLE_MODE_TEXT
qaCbkCatEventReportInd.h, 1447
eTLV_CBK_SETUP_MENU
qaCbkCatEventReportInd.h, 1446
eTLV_END_PROACTIVE_SESSION_LENGTH
qaCbkCatEventReportInd.h, 1447
eTLV_IND_OMA_DM_CONFIG
qaCbkSwiOmaDmEventReportInd.h, 1448
eTLV_IND_OMA_DM_FOTA
qaCbkSwiOmaDmEventReportInd.h, 1448
eTLV_IND_OMA_DM_NOT
qaCbkSwiOmaDmEventReportInd.h, 1448
eTLV_REFRESH_LENGTH
qaCbkCatEventReportInd.h, 1447
eTLV_RF_BAND_INFO
qaNasGetRFBandInfo.h, 1861
eTLV_SETUP_EVENT_LIST_LENGTH
qaCbkCatEventReportInd.h, 1447
eTMD
common.h, 1328
eUIM
common.h, 1328
eVOICE
common.h, 1328
eWDS
common.h, 1328
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID-
_IDENT_FOR_PROFILE
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT-
_DEFINED
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_-
PROFILES

qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_END
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_FAIL
qmerrno.h, 1869
eWDS_ERR_PROFILE_REG_RESULT_LIST_END
qmerrno.h, 1869
ECIOThresListLen
ECIOThresh, 233
nas_ECIOThresh, 431
ECIOThresh, 233
ECIOThresListLen, 233
pECIOThresList, 233
ECTCallState
ECTNum, 234
ECTNum, 234
ECTCallState, 234
number, 234
presentationInd, 234
eDevState
qaGobiApiCbk.h, 1464
eDevice
sGetDeviceSeriesResult, 831
eGetDeviceSeries
qaGobiApiFms.h, 1592
eGobiDeviceSeries
qaGobiApiFms.h, 1589
eGobiImageCarrier
qaGobiApiFms.h, 1589
eGobiImageGPS
qaGobiApiFms.h, 1590
eGobiImageRegion
qaGobiApiFms.h, 1590
eGobiImageTech
qaGobiApiFms.h, 1590
eLOG_LEVEL
common.h, 1327
EMTlv
NASQmiCbkNasSystemSelPrefInd, 549
eQCWWANError
qmerrno.h, 1864
eQMI_SVC
common.h, 1328
eQMISARRFState
qaGobiApiSar.h, 1692
eQaQMIService
qaGobiApiCbk.h, 1501
ERI_DATA_MAX_SIZE
dms.h, 1337
ERIFileparams, 235
pFile, 236
pFileSize, 236
qaGobiApiDms.h, 1552
eSMSEventType
qaGobiApiCbk.h, 1464
ESNString
unpack_dms_GetDeviceSerialNumbers_t, 996
eSYS_SRV_DOMAIN
qaGobiApiNas.h, 1636
ETWSPLMNInfo
eTWSPLMNInfoTlv, 237
eTWSPLMNInfoTlv, 237
ETWSPLMNInfo, 237
TlvPresent, 237
ETWSPLMNTlv
unpack_sms_SetNewSMSCallback_ind_t, 1114
ETWSTlv
unpack_sms_SetNewSMSCallback_ind_t, 1114
eTimeout
common.h, 1328
EVENT_MASK_CARD
qaGobiApiCbk.h, 1461
eValid
LibPackTFTIDParams, 354
TFTIDParams, 920
EarMute
GetAudioProfileResp, 256
GetM2MAudioProfileResp, 273

GetM2MAVMuteResp, 275
SetAudioProfileReq, 810
SetM2MAVMuteReq, 821
earfcn
infoInterFreq, 324
LTEInfoIntrafreq, 389
ltePCI, 392
nas_infoInterFreq, 446
nas_LTEInfoIntrafreq, 454
nas_ltePCI, 456
nas_umtsLTENbrCell, 513
umtsLTENbrCell, 983
earfcn0
lteEARFCN, 383
nas_lteEARFCN, 448
earfcn1
lteEARFCN, 384
nas_lteEARFCN, 448
ecio
CDMASSInfo, 170
cdmaSSInfo, 170
ecioListElement, 233
HDRSSInfo, 301
hdrSSInfo, 302
nas_ecioListElement, 430
nas_rxInfo, 493
nas_UMTSInfo, 511
rxInfo, 788
TDSCDMASigInfoExt, 915
tdscdmaSigInfoExt, 915
UMTSInfo, 981
ecioDelta
nas_SLQSSignalStrengthsIndReq, 499
SLQSSignalStrengthsIndReq, 851
ecioInfo
nas_SLQSSignalStrengthsInformation, 500
SLQSSignalStrengthsInformation, 852
ecioList
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
ecioListElement, 232
ecio, 233
radioIf, 233
ecioListLen
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
ecioThresholdList
nas_SLQSSignalStrengthsIndReq, 499
SLQSSignalStrengthsIndReq, 851
ecioThresholdListLen
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
egprsSupp
GSMSysInfo, 293
nas_GSMSysInfo, 440
egprsSuppValid
GSMSysInfo, 293
nas_GSMSysInfo, 440
elevation
loc_satelliteInfo, 367
satelliteInfo, 795
EmerMode
NASEmergencyModeTlv, 523
emmConnState
LTEInfo, 387
nas_LTEInfo, 451
emmState
LTEInfo, 387
nas_LTEInfo, 451
emmSubState
LTEInfo, 387
nas_LTEInfo, 451
Enable
SetM2MAudioLPBKReq, 819
enable
pack_pds_SetPDSState_t, 630
pack_qos_SLQSSetQosEventCallback_t, 633
enabled
pack_rms_SetSMSWake_t, 633
unpack_rms_GetSMSWake_t, 1110
unpack_wds_GetMobileIPProfile_t, 1168
EncryptProt
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733
EncryptedPIN1
pack_uim_ChangePin_t, 649
pack_uim_SetPinProtection_t, 651
pack_uim_UnblockPin_t, 658
encryptedPIN1, 234
pin1Len, 235
pin1Val, 235
EndProactiveSession
CatEndProactiveSessionTlv, 159
EngineState
GPSStateInfo, 287
engineState
QmiCbkLocEngineStateInd, 746
unpack_loc_EngineState_Ind_t, 1034
entries
t_Sv, 913
eqmiCbkSetStatus
sms.h, 1886
eriData
eriDataparams, 235
eriDataLen
eriDataparams, 235
eriDataparams, 235
eriData, 235
eriDataLen, 235
eriFile
unpack_dms_SLQSGetERIFile_t, 1012
error
unpack_wds_GetLastMobileIPError_t, 1167
errorClass
SMSAsyncRawSend_s, 869
errorRate

errorRateListElement, 236
nas_errorRateListElement, 432
errorRateInfo
nas_SLQSSignalStrengthsInformation, 500
SLQSSignalStrengthsInformation, 853
errorRateList
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
errorRateListElement, 236
errorRate, 236
radioIf, 237
errorRateListLen
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
errorState
slotInf, 836
slotInfo, 838
uim_slotInfo, 956
esn
unpack_dms_GetSerialNumbers_t, 1002
esnSize
serialNumbersInfo, 801
unpack_dms_GetDeviceSerialNumbers_t, 996
EspSpi
unpack_qos_swiQosFilter_t, 1103
EtwsMessageInfo
sMSEtwsMessageTlv, 872
event
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 1099
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1100
event_Index
QmiCbkCatEventStatusReportInd, 739
EventID
CatCommonEventTlv, 158
EventLength
CatCommonEventTlv, 158
eventMask
CATEventDataType, 159
pack_uim_SLQSUIMEventRegister_t, 652
UIMEventRegisterReqResp, 962
unpack_uim_SLQSUIMEventRegister_t, 1132
eventRegister
LOCEventRegisterReqResp, 374
pack_loc_EventRegister_t, 594
eventType
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1122
evrcCapability
prefVoiceSO, 720
voice_prefVoiceSO, 1227
executingImage
FMSImageIDEntries, 246
ImageIDEntries, 308
exponent
pktErrRate, 713
unpack_qos_pktErrRate_t, 1092
extBit
calledPartySubAdd, 146
voice_calledPartySubAdd, 1206
extDispInfo
extDispRecInfo, 238
extDispInfoLen
extDispRecInfo, 238
extDispRecInfo, 237
dispType, 238
extDispInfo, 238
extDispInfoLen, 238
ExtErrorCode
PackCreateProfileOut, 698
extPowerState
LOCExtPowerStateReqResp, 374
pack_loc_SetExtPowerState_t, 594
extendedErrorCode
unpack_wds_SLQSDeleteProfile_t, 1173

FIRST_INSTANCE
qaGobiApiCbk.h, 1461
FLOAT
SwiDataTypes.h, 1904
FMSImageElement, 244
buildId, 244
buildIdLength, 244
imageId, 244
imageType, 244
FMSImageIDEntries, 245
executingImage, 246
imageIDElement, 246
imageIDSize, 246
imageType, 246
maxImages, 246
FMSImageIdElement, 244
buildID, 245
buildIDLength, 245
failureCount, 245
imageID, 245
storageIndex, 245
FMSImageList, 246
imageIDEntries, 247
listSize, 247
FMSPrefImageList, 247
listEntries, 247
listSize, 247
FORBIDDEN_INDEX
qaNasPerformNetworkScan.h, 1862
FOTAUpdate
_SLQSOMADMSettingsReqParams, 94
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 647
unpack_swioma_SLQSOMADMGetSettings_t, 1127
FOTAdownload
_SLQSOMADMSettingsReqParams, 93
_SLQSOMADMSettingsReqParams3, 94
pack_swioma_SLQSOMADMSetSettings_t, 647

unpack_swioma_SLQSOMADMGetSettings_t, 1127
FSNumber
FactorySequenceNumber, 238
facility
pack_dms_UIMGetControlKeyStatus_t, 585
pack_dms_UIMSetControlKeyProtection_t, 586
pack_dms_UIMUnblockControlKey_t, 587
facilityCk
pack_dms_UIMSetControlKeyProtection_t, 586
pack_dms_UIMUnblockControlKey_t, 588
facilityState
pack_dms_UIMSetControlKeyProtection_t, 586
unpack_dms_UIMGetControlKeyStatus_t, 1021
FactorySequenceNumber, 238
FSNumber, 238
failureCount
FMSImageIdElement, 245
ImageIdElement, 307
failureReason
ssdatasession_params, 888
failureReasonv4
ssdatasession_params, 888
failureReasonv6
ssdatasession_params, 888
family
pack_wds_GetDefaultProfileNum_t, 679
pack_wds_SetDefaultProfileNum_t, 683
feature
depersonalizationInformation, 217
personalizationStatus, 707
uim_depersonalizationInformation, 942
uim_personalizationStatus, 948
fileAttributes, 238
fileID, 241
fileSize, 241
fileType, 241
rawLen, 241
rawValue, 241
recordCount, 241
recordSize, 241
secActivate, 241
secActivateMask, 241
secDeactivate, 241
secDeactivateMask, 241
secIncrease, 241
secIncreaseMask, 241
secRead, 241
secReadMask, 241
secWrite, 241
secWriteMask, 241
fileID
fileAttributes, 241
fileInfo, 242
uim_fileAttributes, 946
uim_fileInfo, 947
fileIndex
pack_uim_ReadTransparent_t, 650
pack_uim_SLQSUIMGetFileAttributes_t, 654
UIMGetFileAttributesReq, 964
UIMReadTransparentReq, 968
fileInfo, 241
fileID, 242
path, 242
pathLen, 242
fileSize
fileAttributes, 241
uim_fileAttributes, 946
fileType
fileAttributes, 241
uim_fileAttributes, 946
fill_pack_ctx
common.h, 1329
fill_sdu_hdr
common.h, 1329
filterId
LibPackTFTIDParams, 354
TFTIDParams, 920
Firmware Management Service (FMS), 46
FirmwareID
fwinfo_s, 248
FirmwareUpdatStat, 242
pImgType, 243
pLogString, 243
pLogStringLen, 244
pRefData, 244
pRefString, 244
pRefStringLen, 244
ResCode, 244
fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 644
SwiLocGetAutoStartResp, 895
SwiLocSetAutoStartReq, 896
unpack_swiloc_SwiLocGetAutoStart_t, 1121
fix_rate_reported
SwiLocGetAutoStartResp, 895
unpack_swiloc_SwiLocGetAutoStart_t, 1122
fix_type
pack_swiloc_SwiLocSetAutoStart_t, 644
SwiLocGetAutoStartResp, 895
SwiLocSetAutoStartReq, 896
unpack_swiloc_SwiLocGetAutoStart_t, 1122
fix_type_reported
SwiLocGetAutoStartResp, 895
unpack_swiloc_SwiLocGetAutoStart_t, 1122
flags
sensorData, 798
sensorData_t, 799
flowLabel
LibPackTFTIDParams, 354
TFTIDParams, 920
fms.h, 1381
GetValidFwPriCombinations, 1382
pack_fms_GetImagesPreference, 1382
pack_fms_GetStoredImages, 1383
pack_fms_SetImagesPreference, 1383

unpack_fms_GetImagesPreference, 1383
unpack_fms_GetStoredImages, 1383
unpack_fms_SetImagesPreference, 1383
Forbidden
nas_QmiNas3GppNetworkInfo, 486
SlqsNas3GppNetworkInfo, 842
ForceRev0
unpack_nas_GetCDMANetworkParameters_t, 1046
ForceXTRADownload
qaGobiApiPds.h, 1671
format
SMSTransferRouteMTMessage, 882
sMSTransferRouteMTMessage, 883
ForwardMac
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733
fqdnAddr
PCSCFFQDNAddress, 700
wds_PCSCFFQDNAddress, 1302
fqdnLen
PCSCFFQDNAddress, 700
wds_PCSCFFQDNAddress, 1302
freeSlots
sms_maxStorageSizeResp, 860
smsMaxStorageSizeResp, 876
freq
nas_PhyCaAggPcellInfo, 475
nas_PhyCaAggScellIndType, 477
nas_PhyCaAggScellInfo, 480
NASPhyCaAggPcellInfo, 542
NASPhyCaAggScellIndType, 544
NASPhyCaAggScellInfo, 545
PhyCaAggPcellInfo, 707
PhyCaAggScellIndType, 709
PhyCaAggScellInfo, 712
freqsLen
LTEInfoInterfreq, 387
LTEInfoNeighboringGSM, 390
LTEInfoNeighboringWCDMA, 391
nas_LTEInfoInterfreq, 452
nas_LTEInfoNeighboringGSM, 455
nas_LTEInfoNeighboringWCDMA, 456
fromServiceId
BroadcastConfig, 142
sms_BroadcastConfig, 855
fumoResultCode
omaDmFotaTlvExt, 575
function
pack_swiloc_SwiLocSetAutoStart_t, 644
SwiLocGetAutoStartResp, 895
SwiLocSetAutoStartReq, 896
unpack_swiloc_SwiLocGetAutoStart_t, 1122
function_reported
SwiLocGetAutoStartResp, 895
unpack_swiloc_SwiLocGetAutoStart_t, 1122
FwAutoCheck
unpack_swioma_SLQSOMADMGetSettings_t, 1127
FwAvailability
unpack_swioma_SLQSOMADMStartSession_t, 1127
fwdloadsize
omaDmFotaTlv, 573
unpack_omaDmFotaTlv_t, 1087
fwinfo_s, 247
Carrier, 248
FirmwareID, 248
GPSCapability, 248
Region, 248
Technology, 248
fwloadComplete
omaDmFotaTlv, 573
unpack_omaDmFotaTlv_t, 1087
fwvers
CurrentImgList, 197
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1015

g
qmifwinfo_s, 761
gDlBitRate
LibPackQosClassID, 352
QosClassID, 769
GERANInfo, 248
arfcn, 249
bsic, 249
cellID, 249
insNmrCellInfo, 249
lac, 250
nmrInst, 250
plmn, 250
rxLev, 250
timingAdvance, 250
GPRSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1179
GPRSQoS, 282
delayClass, 283
meanThroughputClass, 283
peakThroughputClass, 283
precedenceClass, 283
reliabilityClass, 283
GPRSRequestedQoS, 283
delayClass, 284
meanThroughputClass, 284
peakThroughputClass, 284
precedenceClass, 284
reliabilityClass, 284
GPSCapability
fwinfo_s, 248
GPSLPM
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
GPSSel
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
GPSStateInfo, 284

Altitude, 287
EngineState, 287
glo_almanac_sv_msk, 287
glo_ephemeris_sv_msk, 287
glo_health_sv_msk, 287
glo_visible_sv_msk, 287
gps_almanac_sv_msk, 287
gps_ephemeris_sv_msk, 287
gps_health_sv_msk, 287
gps_visible_sv_msk, 287
HorizontalUncertainty, 287
Iono_valid, 287
Latitude, 287
Longitude, 287
sbas_almanac_sv_msk, 287
sbas_ephemeris_sv_msk, 287
sbas_health_sv_msk, 287
sbas_visible_sv_msk, 288
Time_uncert_ms, 288
TimeStmp_gps_week, 288
TimeStmp_tow_ms, 288
ValidMask, 288
VerticalUncertainty, 288
xtra_start_gps_minutes, 288
xtra_start_gps_week, 288
xtra_valid_duration_hours, 288
GSMRSSIThresh, 290
GSMRSSIThreshListLen, 290
pGSMRSSIThreshList, 290
GSMRSSIThreshListLen
GSMRSSIThresh, 290
nas_GSMRSSIThresh, 436
GSMSSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1072
GSMSrvStatusInfo, 290
isPrefDataPath, 291
srvStatus, 291
trueSrvStatus, 291
GSMSysInfo, 291
cellId, 293
cellIdValid, 293
dtmSupp, 293
dtmSuppValid, 293
egprsSupp, 293
egprsSuppValid, 293
lac, 293
lacValid, 294
MCC, 294
MNC, 294
networkIdValid, 294
regRejectInfoValid, 294
rejCause, 294
rejectSrvDomain, 294
sysInfoGSM, 294
gUlBitRate
LibPackQosClassID, 352
QosClassID, 769
GWAOPTlv
NASQmiCbkNasSystemSelPrefInd, 549
GWAcqOrderPref
NASGWAcqOrderPrefTlv, 531
GWAddressV4
unpack_wds_SLQSGetRuntimeSettings_t, 1179
gcDumpStrLen
CrashInfo, 188
gcdumpString
crashInformation, 190
gcdumpStrlen
crashInformation, 190
Generator
GetAudioProfileReq, 255
GetAudioVolTLBConfigReq, 257
GetM2MAudioProfileResp, 273
GetM2MAudioVolumeReq, 274
SetAudioProfileReq, 810
SetAudioVolTLBConfigReq, 811
SetM2MAudioVolumeReq, 821
geoSysIdx
AddCDMASysInfo, 116
AddSysInfo, 116
nas_AddCDMASysInfo, 410
nas_AddSysInfo, 410
geranArfcn
geranInstInfo, 250
nas_geranInstInfo, 434
geranBsicBcc
geranInstInfo, 250
nas_geranInstInfo, 434
geranBsicNcc
geranInstInfo, 250
nas_geranInstInfo, 435
geranInst
nas_UMTSInfo, 511
UMTSInfo, 981
GeranInstInfo
nas_UMTSInfo, 511
UMTSInfo, 981
geranInstInfo, 250
geranArfcn, 250
geranBsicBcc, 250
geranBsicNcc, 250
geranRssi, 250
geranRssi
geranInstInfo, 250
nas_geranInstInfo, 435
get_version
common.h, 1329
GetACCOLC
qaGobiApiNas.h, 1637
GetANAAAAuthenticationStatus
qaGobiApiNas.h, 1637
GetActivationState
qaGobiApiDms.h, 1556
getAllCallInfo
arrCallInfo, 131
voice_arrCallInfo, 1201

getAllCallInformation, 251
ALS, 251
Callinfo, 251
isEmpty, 251
GetAllCallRmtPtyName
arrRemotePartyName, 133
voice_arrRemotePartyName, 1203
getAllCallRmtPtyName, 251
callID, 252
RemotePartyName, 252
getAllCallRmtPtyNum, 252
callID, 252
RemotePartyNum, 252
GetAudioPathConfigReq, 252
Item, 253
Profile, 253
GetAudioPathConfigResp, 253
pCodecSTGain, 254
pDTMFTXGain, 255
pECMode, 255
pMICGainSelect, 255
pNSEnable, 255
pRXAGCList, 255
pRXAVCAGCSwitch, 255
pRXAVCList, 255
pRXPCMIIRFltr, 255
pTXAGCList, 255
pTXAVCSwitch, 255
pTXGain, 255
pTXPCMIIRFltr, 255
GetAudioProfileReq, 255
Generator, 255
GetAudioProfileResp, 255
EarMute, 256
MicMute, 256
Profile, 256
Volume, 256
GetAudioVolTLBConfigReq, 257
Generator, 257
Item, 257
Profile, 257
Volume, 257
GetAudioVolTLBConfigResp, 257
ResCode, 258
GetAutoconnect
qaGobiApiWds.h, 1825
GetByteTotals
qaGobiApiWds.h, 1825
GetCDMANetworkParameters
qaGobiApiNas.h, 1638
getCallFWExtInfo, 258
CallFWExtInfo, 258
numInstances, 258
getCallFWInfo, 258
CallFWInfo, 259
numInstances, 259
GetConnectionRate
qaGobiApiWds.h, 1826
GetCustomFeatureV2
unpack_dms_GetCustFeaturesV2_t, 993
getCustomFeatureV2, 259
pCustSettingInfo, 259
pCustSettingList, 259
pGetCustomInput, 259
getCustomInput, 260
cust_id, 260
list_type, 260
getDUNCallInfoReq, 260
Mask, 261
pReportChannelRate, 261
pReportConnStatus, 261
pReportDataBearerTech, 261
pReportDormStatus, 261
pTransferStatInd, 261
getDUNCallInfoResp, 262
pCallEndReason, 264
pChannelRate, 264
pConnectionStatus, 264
pDataBearerTech, 264
pDormancyStatus, 264
pLastCallDataBearerTech, 264
pLastCallRXOKBytesCnt, 264
pLastCallTXOKBytesCnt, 264
pMdmCallDurationActive, 264
pRXOKBytesCount, 264
pTXOKBytesCount, 264
GetDataBearerTechnology
qaGobiApiWds.h, 1827
GetDefaultProfile
qaGobiApiWds.h, 1827
GetDefaultProfileLTE
qaGobiApiWds.h, 1829
GetDefaultProfileNum
qaGobiApiWds.h, 1831
GetDeviceCapabilities
qaGobiApiDms.h, 1556
GetDormancyState
qaGobiApiWds.h, 1831
getDyingGaspCfg, 264
pDestSMSContent, 265
pDestSMSNum, 265
getDyingGaspStatistics, 265
pSMSAttemptedFlag, 265
pTimeStamp, 265
GetErrRateResp, 266
pCDMAFrameErrRate, 266
pGSMBER, 266
pHDRPackErrRate, 266
pWCDMABER, 266
GetFirmwareRevision
qaGobiApiDms.h, 1557
GetFirmwareRevisions
qaGobiApiDms.h, 1558
GetHRPDStatsResp, 266
pDRCParams, 267
pPilotSetData, 267

pUATI, 267
GetHardwareRevision
qaGobiApiDms.h, 1558
GetHomeNetwork
qaGobiApiNas.h, 1639
GetHomeNetwork3GPP2
qaGobiApiNas.h, 1640
GetIMSI
qaGobiApiDms.h, 1559
GetIMSSMSConfigParams, 267
pPhoneCtxtURI, 268
pPhoneCtxtURILen, 268
pSMSFormat, 268
pSMSOverIPNwInd, 268
pSettingResp, 268
GetIMSUserConfigParams, 268
pIMSDomain, 269
pIMSDomainLen, 269
pSettingResp, 269
GetIMSVoIPConfigResp, 269
pAmrMode, 271
pAmrOctetAligned, 271
pAmrWBMode, 271
pAmrWBOctetAligned, 271
pAmrWbEnable, 271
pMinSessionExpiryTimer, 271
pRTPRTCPInactTimer, 271
pRingBackTimer, 271
pRingingTimer, 271
pScrAmrEnable, 271
pScrAmrWbEnable, 271
pSessionExpiryTimer, 271
pSettingResp, 271
GetIPAddressLTE
qaGobiApiWds.h, 1832
GetImageStore
qaGobiApiFms.h, 1593
GetImagesPreference
qaGobiApiFms.h, 1592
getIndicationRegResp
qaGobiApiSms.h, 1696
GetInstIDResp, 271
pIPFamily, 271
pInstanceID, 271
GetLastMobileIPError
qaGobiApiWds.h, 1832
GetM2MAVMuteReq, 274
Profile, 274
GetM2MAVMuteResp, 275
CwtMute, 275
EarMute, 275
MicMute, 275
GetM2MAudioProfileReq, 272
pGenerator, 272
GetM2MAudioProfileResp, 272
CwtMute, 273
EarMute, 273
Generator, 273
MicMute, 273
Profile, 273
Volume, 273
GetM2MAudioVolumeReq, 273
Generator, 274
Profile, 274
GetM2MAudioVolumeResp, 274
Level, 274
GetM2MSpkrGainReq, 275
Profile, 276
GetM2MSpkrGainResp, 276
Value, 276
GetManufacturer
qaGobiApiDms.h, 1559
GetMobileIP
qaGobiApiWds.h, 1833
GetMobileIPProfile
qaGobiApiWds.h, 1833
GetModelID
qaGobiApiDms.h, 1560
getMsgWaitingInfo, 276
msgWaitInfo, 277
numInstances, 277
GetNetworkPreference
qaGobiApiNas.h, 1642
GetNetworkTime
qaGobiApiDms.h, 1560
GetNetworkTimeResp, 277
p3GPP2TimeInfo, 277
p3GPPTimeInfo, 277
GetOfflineReason
qaGobiApiDms.h, 1561
GetPDSDefaults
qaGobiApiPds.h, 1671
GetPDSState
qaGobiApiPds.h, 1672
GetPRLVersion
qaGobiApiDms.h, 1562
GetPacketStatistics
qaGobiApiWds.h, 1834
GetPacketStatus
qaGobiApiWds.h, 1835
GetPortAutomaticTracking
qaGobiApiPds.h, 1672
GetPower
qaGobiApiDms.h, 1562
GetProfileSettingIn
qaGobiApiWds.h, 1822
GetProfileSettingOut
qaGobiApiWds.h, 1822
GetRFInfo
qaGobiApiNas.h, 1643
GetRegMgrConfigParams, 277
pIMSTestMode, 278
pPCSCFPort, 278
pPriCSCFPortName, 278
pPriCSCFPortNameLen, 278
pSettingResp, 278

GetSIPConfigResp, 278
pSIPLocalPort, 279
pSettingResp, 279
pSigCompEnabled, 279
pSubscribeTimer, 279
pTimerSIPReg, 279
pTimerT1, 279
pTimerT2, 279
pTimerTf, 279
GetSMSCAddress
qaGobiApiSms.h, 1700
GetSMSWake
qaGobiApiRms.h, 1691
GetSerialNumbers
qaGobiApiDms.h, 1562
GetServiceAutomaticTracking
qaGobiApiPds.h, 1673
GetServingNetwork
qaGobiApiNas.h, 1643
GetServingNetworkCapabilities
qaGobiApiNas.h, 1644
GetSessionDuration
qaGobiApiWds.h, 1836
GetSessionIDResp, 278
pSessionIDv4, 278
pSessionIDv6, 278
GetSessionState
qaGobiApiWds.h, 1836
GetSignalStrengths
qaGobiApiNas.h, 1645
GetStoredImages
qaGobiApiFms.h, 1593
getTransLayerInfoResp
qaGobiApiSms.h, 1697
getTransNWRegInfoResp
qaGobiApiSms.h, 1697
GetValidFwPriCombinations
fms.h, 1382
GetVoiceNumber
qaGobiApiDms.h, 1563
GetXTRAAutomaticDownload
qaGobiApiPds.h, 1673
GetXTRANetwork
qaGobiApiPds.h, 1674
GetXTRAValidity
qaGobiApiPds.h, 1674
glo_almanac_sv_msk
GPSStateInfo, 287
glo_ephemeris_sv_msk
GPSStateInfo, 287
glo_health_sv_msk
GPSStateInfo, 287
glo_visible_sv_msk
GPSStateInfo, 287
globalCellId
LTEInfoIntrafreq, 389
nas_LTEInfoIntrafreq, 454
glog
common.h, 1330
gloglvl
common.h, 1330
GnssData, 280
mask, 281
gnssSvId
loc_satelliteInfo, 367
satelliteInfo, 795
gnssSvInfoNotification, 281
bAltitudeAssumed, 282
pSatelliteInfo, 282
gnssSvUsedList
loc_svUsedforFix, 370
svUsedforFix_s, 892
gnssSvUsedList_len
loc_svUsedforFix, 370
svUsedforFix_s, 892
Gpp2TimeZone
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1061
GppNetworkDSTAdjustment
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
GppTimeZone
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
gps_almanac_sv_msk
GPSStateInfo, 287
gps_ephemeris_sv_msk
GPSStateInfo, 287
gps_health_sv_msk
GPSStateInfo, 287
gps_visible_sv_msk
GPSStateInfo, 287
GpsEnable
custFeaturesInfo, 203
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
gpsTime
qaGobiApiCbk.h, 1464
gpsTime_s, 288
gpsTimeOfWeekMs, 288
gpsWeek, 288
gpsTimeOfWeekMs
gpsTime_s, 288
loc_gpsTime, 363
t_gpsTime, 912
gpsWeek
gpsTime_s, 288
loc_gpsTime, 363
t_gpsTime, 912
grntDownlinkBitrate
LibPackUMTSQoS, 356
UMTSMinQoS, 985
UMTSQoS, 988
wds_UMTSMinQoS, 1306
grntUplinkBitrate
LibPackUMTSQoS, 356

UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
gsmAmrStat
curAMRConfig, 195
voice_curAMRConfig, 1219
GsmCellInfo
lteGsmCellInfo, 385
nas_lteGsmCellInfo, 449
gsmCellInfo, 288
arfcn, 289
band1900, 289
bsicId, 289
cellIdValid, 289
rssi, 289
srxlev, 289
gsmUmtsDl
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
gsmUmtsUl
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
guaranteedRate
dataRate, 212
unpack_qos_dataRate_t, 1090
gwAddressV6
IPV6GWAddressInfo, 328
wds_IPV6GWAddressInfo, 1301
gwV6PrefixLen
IPV6GWAddressInfo, 328
wds_IPV6GWAddressInfo, 1301
gyroAcceptReady
qaGobiApiCbk.h, 1464
gyroAcceptReady_s, 294
batchPerSec, 294
injectEnable, 294
samplesPerBatch, 295
gyroData
pack_loc_SLQSLOCInjectSensorData_t, 602
gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 602
gyroTempAcceptReady
qaGobiApiCbk.h, 1465
gyroTempAcceptReady_s, 295
batchPerSec, 295
injectEnable, 295
samplesPerBatch, 295
gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 602

HASPI
unpack_wds_GetMobileIPProfile_t, 1168
HAState
unpack_wds_GetMobileIPProfile_t, 1168
HDOP
loc_precisionDilution, 365
precisionDilution_s, 718
HDRECIOThresh, 295
HDRECIOThreshListLen, 296
pHDRECIOThreshList, 296
HDRECIOThreshListLen
HDRECIOThresh, 296
nas_HDRECIOThresh, 440
HDRIOThresh, 296
HDRIOThreshListLen, 296
pHDRIOThreshList, 296
HDRIOThreshListLen
HDRIOThresh, 296
nas_HDRIOThresh, 441
HDRPersonalityInd, 296
pCurrentPersonality, 297
pPersonalityListLength, 297
pProtocolSubtypeElement, 297
HDRPersonalityResp, 297
pCurrentPersonality, 297
pPersonalityListLength, 297
pProtocolSubtypeElement, 297
HDRProtSubtypResp, 297
pAppSubType, 298
pCurrentPrsnlty, 298
pPersonalityListLength, 298
pProtoSubTypElmnt, 298
HDRRSSIThresh, 298
HDRRSSIThreshListLen, 299
pHDRRSSIThreshList, 299
HDRRSSIThreshListLen
HDRRSSIThresh, 299
nas_HDRRSSIThresh, 441
HDRSINRThresListLen
HDRSINRThresh, 299
nas_HDRSINRThresh, 442
HDRSINRThresh, 299
HDRSINRThresListLen, 299
pHDRSINRThresList, 299
HDRSINRThreshListLen
HDRSINRThreshold, 300
nas_HDRSINRThreshold, 443
HDRSINRThreshold, 299
HDRSINRThreshListLen, 300
pHDRSINRThreshList, 300
HDRSSInfo, 300
ecio, 301
io, 301
rssi, 301
sinr, 301
unpack_nas_SLQSNasGetSigInfo_t, 1072
HDRSysInfo, 302
hdrActiveProt, 304
hdrActiveProtValid, 304
hdrPersonality, 304
hdrPersonalityValid, 304
is856SysId, 304
is856SysIdValid, 304
isSysPrlMatch, 304
isSysPrlMatchValid, 304
sysInfoHDR, 304
HTTPStatus

_SwiAvmsEventInd, 100
HWVersion
DeviceConfigDetail, 219
nas_DeviceConfigDetail, 429
HardwareControlledMode
unpack_dms_GetPower_t, 1002
has_accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 602
has_acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 602
has_accleroData
pack_loc_SLQSLOCInjectSensorData_t, 602
has_altitudeSrcInfo
pack_loc_SLQSLOCInjectPosition_t, 599
has_altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 599
has_altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 599
has_confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 603
has_gyroData
pack_loc_SLQSLOCInjectSensorData_t, 602
has_gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 602
has_gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 602
has_horConfidence
pack_loc_SLQSLOCInjectPosition_t, 599
has_horReliability
pack_loc_SLQSLOCInjectPosition_t, 599
has_horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 599
has_latitude
pack_loc_SLQSLOCInjectPosition_t, 600
has_longitude
pack_loc_SLQSLOCInjectPosition_t, 600
has_opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 602
has_positionSrc
pack_loc_SLQSLOCInjectPosition_t, 600
has_rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 600
has_rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 600
has_timestampAge
pack_loc_SLQSLOCInjectPosition_t, 600
has_timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 600
has_vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 600
has_vertRelicability
pack_loc_SLQSLOCInjectPosition_t, 600
has_vertUnc
pack_loc_SLQSLOCInjectPosition_t, 600
hdrActiveProt
HDRSysInfo, 304
nas_HDRSysInfo, 445
hdrActiveProtValid
HDRSysInfo, 304
nas_HDRSysInfo, 445
hdrHybrid
detailSvcInfo, 218
nas_detailSvcInfo, 428
HdrPersonality
unpack_nas_SLQSGetServingSystem_t, 1062
hdrPersonality
HDRSysInfo, 304
nas_HDRSysInfo, 445
NASServingSystemInfo, 551
qaQmiServingSystemParam, 739
ServingSystemInfo, 803
hdrPersonalityValid
HDRSysInfo, 304
nas_HDRSysInfo, 445
hdrSSInfo, 301
ecio, 302
io, 302
rssi, 302
sinr, 302
hdrSrvStatus
detailSvcInfo, 218
nas_detailSvcInfo, 428
healthStatus
loc_satelliteInfo, 367
satelliteInfo, 795
helper_get_resp_ctx
common.h, 1329
helper_get_xid
common.h, 1329
helper_set_log_func
common.h, 1329
helper_set_log_lvl
common.h, 1329
homeOrigVoiceSO
prefVoiceSO, 720
voice_prefVoiceSO, 1227
homePageVoiceSO
prefVoiceSO, 720
voice_prefVoiceSO, 1227
homeSIDNID, 304
numInstances, 304
SidNid, 304
horConfidence
pack_loc_SLQSLOCInjectPosition_t, 600
horReliability
pack_loc_SLQSLOCInjectPosition_t, 600
horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 600
HorizontalUncertainty
GPSStateInfo, 287
hotSwap
hotSwapStatus, 305
uim_hotSwapStatus, 947
hotSwapLength
hotSwapStatus, 305
uim_hotSwapStatus, 947

hotSwapStatus, 304
  hotSwap, 305
  hotSwapLength, 305
hour
  nas_timeInfo, 508
  nas_UniversalTime, 514
  timeInfo, 922
  UniversalTime, 991
hsCallStatus
  nas_WCDMASysInfo, 520
  WCDMASysInfo, 1292
hsCallStatusValid
  nas_WCDMASysInfo, 520
  WCDMASysInfo, 1292
hsInd
  nas_WCDMASysInfo, 520
  WCDMASysInfo, 1292
hsIndValid
  nas_WCDMASysInfo, 520
  WCDMASysInfo, 1292
hwType
  wds_DHCPv4HWConfig, 1297
  WdsDHCPv4HWConfig, 1313
  wdsDhcpv4HwConfig, 1312
hwVer
  unpack_dms_GetHardwareRevision_t, 999

iFaceTab
  PCMparams, 700
iFaceTabLen
  PCMparams, 700
iGetByteTotals
  qaGobiApiWds.h, 1837
iGetConnectionRate
  qaGobiApiWds.h, 1837
iGetPacketStatistics
  qaGobiApiWds.h, 1837
iLTEbandValue
  nas_PhyCaAggPcellInfo, 475
  nas_PhyCaAggScellInfo, 480
  NASPhyCaAggPcellInfo, 542
  NASPhyCaAggScellInfo, 545
  PhyCaAggPcellInfo, 707
  PhyCaAggScellInfo, 712
IMCNflag
  unpack_wds_SLQSGetRuntimeSettings_t, 1179
IMEIString
  unpack_dms_GetDeviceSerialNumbers_t, 996
IMG_ID_LEN
  qaGobiApiFms.h, 1588
IMGDETAILS_LEN
  qaGobiApiFms.h, 1588
IMS Service (IMS), 55
IMSAIndRegisterInfo, 308
  pPdpStatusConfig, 309
  pRatHandoverStatusConfig, 309
  pRegStatusConfig, 309
  pServiceStatusConfig, 309
IMSARegistrationStatus, 311
  pImsRegErrCode, 312
  pImsRegStatus, 312
  pNewImsRegStatus, 312
IMSAServiceStatus, 312
  pSmsServiceRat, 314
  pSmsServiceStatus, 314
  pUtServiceRat, 314
  pUtServiceStatus, 314
  pVoipServiceRat, 314
  pVoipServiceStatus, 314
  pVsServiceRat, 314
  pVsServiceStatus, 314
  pVtServiceRat, 314
  pVtServiceStatus, 314
IMSASupportedFieldsResp, 314
  pIndFieldsList, 315
  pReqFieldsList, 315
  pRespFieldsList, 315
IMSASupportedMsgInfo, 315
  pSupportedMsgList, 315
IMSI_M_S1_LENGTH
  qaGobiApiNas.h, 1624
IMSI_M_S2_LENGTH
  qaGobiApiNas.h, 1624
IMSInfo
  sMSOnIMSTlv, 880
IMSTlv
  unpack_sms_SetNewSMSCallback_ind_t, 1114
INDEX_ZERO
  qaNasPerformNetworkScan.h, 1862
INT32
  SwiDataTypes.h, 1904
INT8
  SwiDataTypes.h, 1904
INVALID_INSTACNE
  qaGobiApiCbk.h, 1461
IOThresListLen
  IOThresh, 325
  nas_IOThresh, 447
IOThresh, 325
  IOThresListLen, 325
  pIOThresList, 325
IPAddress
  DataStatusDetail, 214
  nas_DataStatusDetail, 426
IPAddressV6
  IPV6AddressInfo, 327
  ipv6AddressInfo, 327
  wds_IPV6AddressInfo, 1301
IPFamSupport
  pack_dms_SetCustFeature_t, 580
  unpack_dms_GetCustFeature_t, 993
IPFamilyPreference
  pack_wds_SLQSSetIPFamilyPreference_t, 692
  unpack_wds_SLQSGetRuntimeSettings_t, 1179
IPSECSPI
  LibPackTFTIDParams, 354
  TFTIDParams, 920

IPV4
qaGobiApiCbk.h, 1462
IPV4V6
qaGobiApiCbk.h, 1462
IPV6
qaGobiApiCbk.h, 1462
IPV6AddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1179
IPV6AddressInfo, 326
IPAddressV6, 327
IPV6PrefixLen, 327
IPV6GWAddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1179
IPV6GWAddressInfo, 327
gwAddressV6, 328
gwV6PrefixLen, 328
IPV6PrefixLen
IPV6AddressInfo, 327
ipv6AddressInfo, 327
wds_IPV6AddressInfo, 1301
IPv4
unpack_wds_SLQSGetRuntimeSettings_t, 1179
IPv4Addr, 325
addr, 326
subnetMask, 326
wds_IPv4AdTlv, 1300
IPv4AddrTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1163
IPv4DstAddr
unpack_qos_swiQosFilter_t, 1103
IPv4SrcAddr
unpack_qos_swiQosFilter_t, 1103
IPv4Tos
unpack_qos_swiQosFilter_t, 1103
IPv6Addr, 326
addr, 326
prefixLen, 326
IPv6DstAddr
unpack_qos_swiQosFilter_t, 1103
IPv6Label
unpack_qos_swiQosFilter_t, 1103
IPv6SrcAddr
unpack_qos_swiQosFilter_t, 1103
IPv6TrafCls, 328
mask, 328
unpack_qos_swiQosFilter_t, 1103
val, 328
iSLQSMISetIPFamilyPreference
qaGobiApiWds.h, 1837
iSLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1502
iSLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1502
iSLQSSetWdsFirstInstEventCallback
qaGobiApiCbk.h, 1502
iSLQSSetWdsSecondInstEventCallback
qaGobiApiCbk.h, 1502
iSLQSSetWdsThirdInstEventCallback
qaGobiApiCbk.h, 1502
iSLQSSetWdsXferStatsFirstInstCallback
qaGobiApiCbk.h, 1502
iSLQSSetWdsXferStatsSecondInstCallback
qaGobiApiCbk.h, 1502
iSetCATEventCallback
qaGobiApiCbk.h, 1502
iSetSignalStrengthCallback
qaGobiApiCbk.h, 1502
Id
unpack_qos_swiQosFilter_t, 1103
id
BdsSV, 140
CSGID, 193
loc_BdsSV, 358
loc_SV, 369
nas_CSGID, 422
pack_dms_UIMChangePIN_t, 585
pack_dms_UIMSetPINProtection_t, 587
pack_dms_UIMUnblockPIN_t, 588
pack_dms_UIMVerifyPIN_t, 589
QosFlowInfoState, 772
SV, 891
swiQosModifyReq, 908
unpack_qos_QosFlowInfoState_t, 1095
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1100
id_length
custSettingInfo, 206
DMScustSettingInfo, 224
IdleState
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733
image_info_t, 305
buildID, 305
buildIDLen, 305
imageType, 305
uniqueID, 305
ImageElement, 305
buildId, 306
buildIdLength, 306
imageId, 306
imageType, 306
imageID
FMSImageIdElement, 245
ImageIdElement, 307
imageIDElement
FMSImageIDEntries, 246
ImageIDEntries, 308
ImageIDEntries, 307
executingImage, 308
imageIDElement, 308
imageIDSize, 308
imageType, 308
maxImages, 308
imageIDEntries
FMSImageList, 247
ImageList, 308

imageIDSize
FMSImageIDEntries, 246
ImageIDEntries, 308
imageId
FMSImageElement, 244
ImageElement, 306
ImageIdElement, 306
buildID, 307
buildIDLength, 307
failureCount, 307
imageID, 307
storageIndex, 307
ImageList, 308
imageIDEntries, 308
listSize, 308
imageList
unpack_fms_GetStoredImages_t, 1026
ImageListSize
unpack_fms_GetImagesPreference_t, 1025
imageListSize
pack_fms_SetImagesPreference_t, 590
imageType
CurrImageInfo, 199
FMSImageElement, 244
FMSImageIDEntries, 246
image_info_t, 305
ImageElement, 306
ImageIDEntries, 308
ImageTypes
unpack_fms_SetImagesPreference_t, 1027
ImageTypesSize
unpack_fms_SetImagesPreference_t, 1027
imagelistSize
unpack_fms_GetStoredImages_t, 1026
imei_no
unpack_dms_GetSerialNumbers_t, 1003
imeiSize
serialNumbersInfo, 801
unpack_dms_GetDeviceSerialNumbers_t, 996
imeiSvnSize
serialNumbersInfo, 801
unpack_dms_GetDeviceSerialNumbers_t, 996
ImeiSvnString
unpack_dms_GetDeviceSerialNumbers_t, 996
imeisv_svn
unpack_dms_GetSerialNumbers_t, 1003
imgType
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1016
imsCfgIndRegisterInfo, 316
pRegMgrConfigEvents, 317
pSIPConfigEvents, 317
pSMSConfigEvents, 317
pUserConfigEvents, 317
pVoIPConfigEvents, 318
imsRegMgrConfigInfo, 318
pCSCFPortName, 318
pIMSTestMode, 318
pPriCSCFPort, 318
imsRegState
CommInfo, 184
nas_CommInfo, 422
imsSIPConfigInfo, 318
pSIPLocalPort, 319
pSigCompEnabled, 319
pSubscribeTimer, 319
pTimerSIPReg, 319
pTimerT1, 319
pTimerT2, 319
pTimerTf, 319
imsSMSConfigInfo, 319
pPhoneCtxtURI, 320
pSMSFormat, 320
pSMSOverIPNwInd, 320
imsUserConfigInfo, 320
pIMSDomain, 320
imsVoIPConfigInfo, 321
pAmrMode, 323
pAmrOctetAligned, 323
pAmrWBMode, 323
pAmrWBOctetAligned, 323
pAmrWbEnable, 323
pMinSessionExpiryTimer, 323
pRTPRTCPInactTimer, 323
pRingBackTimer, 323
pRingingTimer, 323
pScrAmrEnable, 323
pScrAmrWbEnable, 323
pSessionExpiryTimer, 323
imsaPdpStatusInfo, 309
connetionState, 310
pFailErrorCode, 310
imsaRatStatusInfo, 310
pErrorCodeStr, 311
pRATStatus, 311
pSrcRAT, 311
pTgtRAT, 311
imsaRegStatusInfo, 312
pImsRegStatus, 312
pRegStatusErrorCode, 312
pbIMSRegistered, 312
imsaSvcStatusInfo, 315
pSMSSvcRAT, 316
pSMSSvcStatus, 316
pUTSvcRAT, 316
pUTSvcStatus, 316
pVOIPSvcRAT, 316
pVOIPSvcStatus, 316
pVTSvcRAT, 316
pVTSvcStatus, 316
imsi
unpack_dms_GetIMSI_t, 999
imsi_11_12
CDMASysInfoExt, 174
nas_CDMASysInfoExt, 419
imsiM1112

minBasedIMSI, 405
nas_minBasedIMSI, 466
imsiMS1
minBasedIMSI, 405
nas_minBasedIMSI, 466
imsiMS2
minBasedIMSI, 405
nas_minBasedIMSI, 466
imsiT1112
nas_trueIMSI, 509
trueIMSI, 930
imsiTS1
nas_trueIMSI, 509
trueIMSI, 930
imsiTS2
nas_trueIMSI, 509
trueIMSI, 930
imsiTaddrNum
nas_trueIMSI, 509
trueIMSI, 930
InUse
nas_QmiNas3GppNetworkInfo, 486
SlqsNas3GppNetworkInfo, 842
includes_pcs_digit
nas_QmisNasPcsDigit, 487
SlqsNasPcsDigit, 843
IndFieldsList, 323
indicationFields, 323
indicationFieldsLen, 323
index
pack_wds_GetMobileIPProfile_t, 680
pack_wds_SetDefaultProfileNum_t, 683
pack_wds_SetMobileIPProfile_t, 685
swiQosFilter, 903
swiQosFlow, 906
swiQosReq, 909
unpack_qos_swiQosFilter_t, 1103
unpack_qos_swiQosFlow_t, 1107
unpack_wds_GetDefaultProfileNum_t, 1167
index1xPri
cardStatus, 156
uim_cardStatus, 940
index1xSec
cardStatus, 156
uim_cardStatus, 940
indexGwPri
cardStatus, 156
uim_cardStatus, 940
indexGwSec
cardStatus, 156
uim_cardStatus, 940
indicationFields
IndFieldsList, 323
indicationFieldsLen
IndFieldsList, 323
Info
unpack_nas_SLQSNasSwiOTAMessageCallback-_ind_t, 1076
unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind_t, 1078
infoInterFreq, 324
cell_resel_priority, 324
cellInterFreqParams, 324
cells_len, 324
earfcn, 324
threshXHigh, 325
threshXLow, 325
InfoInterfreq
LTEInfoInterfreq, 387
nas_LTEInfoInterfreq, 452
InitiateDomainAttach
qaGobiApiNas.h, 1646
InitiateNetworkRegistration
qaGobiApiNas.h, 1646
injectEnable
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 294
gyroTempAcceptReady_s, 295
injectSensorDataStatus
QmiCbkLocInjectSensorDataInd, 749
injectTimeSyncStatus
QmiCbkLocInjectTimeInd, 749
insNmrCellInfo
GERANInfo, 249
nas_GERANInfo, 434
instanceId
ssdatasession_params, 888
instancesSize
unpack_nas_GetRFInfo_t, 1050
interval
pack_pds_SetPDSDefaults_t, 630
pack_wds_SLQSSetWdsEventCallback_t, 693
TransferStatsDataType, 928
Io
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
io
HDRSSInfo, 301
hdrSSInfo, 302
nas_SLQSSignalStrengthsInformation, 500
SLQSSignalStrengthsInformation, 853
ioDelta
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
Iono_valid
GPSStateInfo, 287
ip
WdsIpAddressInfoReq, 1317
ipAddress
pack_wds_SetDefaultProfile_t, 682
ipFamily
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1181
ipVersion

LibPackTFTIDParams, 354
TFTIDParams, 920
ipaddr
unpack_wds_GetDefaultProfile_t, 1166
ipaddrv6
unpack_wds_GetDefaultProfile_t, 1166
ipfamily
ssdatasession_params, 888
ipv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
ipv4GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
ipv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
ipv6AddressInfo, 327
IPAddressV6, 327
IPV6PrefixLen, 327
ipv6GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
is856SysId
HDRSysInfo, 304
nas_HDRSysInfo, 445
is856SysIdValid
HDRSysInfo, 304
nas_HDRSysInfo, 445
is_DataRate_Available
unpack_qos_swiQosFlow_t, 1107
is_EspSpi_Available
unpack_qos_swiQosFilter_t, 1103
is_IPv4DstAddr_Available
unpack_qos_swiQosFilter_t, 1103
is_IPv4SrcAddr_Available
unpack_qos_swiQosFilter_t, 1103
is_IPv4Tos_Available
unpack_qos_swiQosFilter_t, 1103
is_IPv6DstAddr_Available
unpack_qos_swiQosFilter_t, 1103
is_IPv6Label_Available
unpack_qos_swiQosFilter_t, 1103
is_IPv6SrcAddr_Available
unpack_qos_swiQosFilter_t, 1103
is_IPv6TrafCls_Available
unpack_qos_swiQosFilter_t, 1103
is_Id_Available
unpack_qos_swiQosFilter_t, 1103
is_Jitter_Available
unpack_qos_swiQosFlow_t, 1107
is_Latency_Available
unpack_qos_swiQosFlow_t, 1107
is_LteBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1012
is_LteQci_Available
unpack_qos_swiQosFlow_t, 1107
is_MaxAllowedPktSz_Available
unpack_qos_swiQosFlow_t, 1107
is_MinPolicedPktSz_Available
unpack_qos_swiQosFlow_t, 1107
is_NxtHdrProto_Available
unpack_qos_swiQosFilter_t, 1103
is_PktErrRate_Available
unpack_qos_swiQosFlow_t, 1107
is_Precedence_Available
unpack_qos_swiQosFilter_t, 1103
is_ProfileId3GPP2_Available
unpack_qos_swiQosFlow_t, 1107
is_RxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1094
is_TCPDstPort_Available
unpack_qos_swiQosFilter_t, 1103
is_TCPSrcPort_Available
unpack_qos_swiQosFilter_t, 1103
is_TdsBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1012
is_TokenBucket_Available
unpack_qos_swiQosFlow_t, 1107
is_TrafficClass_Available
unpack_qos_swiQosFlow_t, 1107
is_TranDstPort_Available
unpack_qos_swiQosFilter_t, 1104
is_TranSrcPort_Available
unpack_qos_swiQosFilter_t, 1104
is_TxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1094
is_UDPDstPort_Available
unpack_qos_swiQosFilter_t, 1104
is_UDPSrcPort_Available
unpack_qos_swiQosFilter_t, 1104
is_val_3GPP2Pri_Available
unpack_qos_swiQosFlow_t, 1107
is_val_3GPPImCn_Available
unpack_qos_swiQosFlow_t, 1107
is_val_3GPPResResidualBER_Available
unpack_qos_swiQosFlow_t, 1107
is_val_3GPPSigInd_Available
unpack_qos_swiQosFlow_t, 1107
is_val_3GPPTraHdlPri_Available
unpack_qos_swiQosFlow_t, 1107
isEmpty
getAllCallInformation, 251
voice_getAllCallInformation, 1221
isInTraffic
nas_txInfo, 510
txInfo, 931
isModByCC
SUPSInfo, 890
isNewFlow
QosFlowInfoState, 772
unpack_qos_QosFlowInfoState_t, 1095
isPrefDataPath
GSMSrvStatusInfo, 291
nas_GSMSrvStatusInfo, 437
nas_SrvStatusInfo, 502

SrvStatusInfo, 886
isRadioTuned
nas_rxInfo, 493
nas_RxSigInfo, 494
rxInfo, 788
RxSigInfo, 791
isSysForbidden
detailSvcInfo, 218
nas_detailSvcInfo, 428
nas_sysInfoCommon, 504
sysInfoCommon, 912
isSysForbiddenValid
nas_sysInfoCommon, 504
sysInfoCommon, 912
isSysPrlMatch
CDMASysInfo, 173
HDRSysInfo, 304
nas_CDMASysInfo, 418
nas_HDRSysInfo, 445
isSysPrlMatchValid
CDMASysInfo, 173
HDRSysInfo, 304
nas_CDMASysInfo, 418
nas_HDRSysInfo, 445
IsVoiceEnabled
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
Item
GetAudioPathConfigReq, 253
GetAudioVolTLBConfigReq, 257
SetAudioVolTLBConfigReq, 811

Jitter
unpack_qos_swiQosFlow_t, 1107

KeyExchange
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733

LITEQMI_QMI_CBK_PARAM_NOCHANGE
sms.h, 1886
LITEQMI_QMI_CBK_PARAM_RESET
sms.h, 1886
LITEQMI_QMI_CBK_PARAM_SET
sms.h, 1886
LBPTlv
NASQmiCbkNasSystemSelPrefInd, 549
LEN
qaGobiApiDcs.h, 1539
LOCEVENTMASKGNSSSVINFO
loc.h, 1387
LOCEVENTMASKNMEA
loc.h, 1388
LOCEVENTMASKWIFIREQ
loc.h, 1389
LOCEventRegisterReqResp, 372
eventRegister, 374
LOCExtPowerStateReqResp, 374
extPowerState, 374
LOCStartReq, 380
pApplicationInfo, 381
pConfigAltitudeAssumed, 381
pHorizontalAccuracyLvl, 381
pIntermediateReportState, 382
pMinIntervalTime, 382
pRecurrenceType, 382
SessionId, 382
LOCStopReq, 382
sessionId, 382
LPCSTR
SwiDataTypes.h, 1904
LTEAttachProfile
unpack_wds_SLQSGet3GPPConfigItem_t, 1174
LTEAttachProfileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1174
LTEAttachProfileListLen
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 691
unpack_wds_SLQSGet3GPPConfigItem_t, 1174
LTEBandPref
NASLTEBandPreferenceTlv, 536
LTECphyCaInfo
unpack_nas_SlqsGetLTECphyCAInfo_t, 1057
LTEInfo, 385
band, 387
bandwidth, 387
emmConnState, 387
emmState, 387
emmSubState, 387
RXChan, 387
TXChan, 387
LTEInfoInterfreq, 387
freqsLen, 387
InfoInterfreq, 387
ueInIdle, 387
LTEInfoIntrafreq, 387
CellParams, 389
cellReselPriority, 389
cellsLen, 389
earfcn, 389
globalCellId, 389
plmn, 389
sIntraSearch, 389
sNonIntraSearch, 389
servingCellId, 389
tac, 389
threshServingLow, 389
ueInIdle, 389
LTEInfoNeighboringGSM, 390
freqsLen, 390
LteGsmCellInfo, 390
ueInIdle, 390
LTEInfoNeighboringWCDMA, 390
freqsLen, 391
LTEWCDMACellInfo, 391
ueInIdle, 391
LTERSRPThresh, 393

LTERSRPThreshListLen, 393
pLTERSRPThreshList, 393
LTERSRPThreshListLen
LTERSRPThresh, 393
nas_LTERSRPThresh, 457
LTERSRQThresh, 393
LTERSRQThreshListLen, 394
pLTERSRQThreshList, 394
LTERSRQThreshListLen
LTERSRQThresh, 394
nas_LTERSRQThresh, 458
LTERSSIThresh, 394
LTERSSIThreshListLen, 394
pLTERSSIThreshList, 394
LTERSSIThreshListLen
LTERSSIThresh, 394
nas_LTERSSIThresh, 458
LTESNRThresListLen
LTESNRThresh, 397
nas_LTESNRThresh, 461
LTESNRThresh, 397
LTESNRThresListLen, 397
pLTESNRThresList, 397
LTESNRThreshListLen
LTESNRThreshold, 398
nas_LTESNRThreshold, 462
LTESNRThreshold, 397
LTESNRThreshListLen, 398
pLTESNRThreshList, 398
LTESSInfo, 398
rsrp, 399
rsrq, 399
rssi, 399
snr, 399
unpack_nas_SLQSNasGetSigInfo_t, 1073
LTESigRptCfg, 395
avgPeriod, 395
rptRate, 395
LTESigRptConfig, 395
avgPeriod, 396
rptRate, 396
LTESysInfo, 399
cellId, 401
cellIdValid, 401
lac, 401
lacValid, 401
MCC, 401
MNC, 401
networkIdValid, 401
regRejectInfoValid, 401
rejCause, 401
rejectSrvDomain, 401
sysInfoLTE, 401
tac, 401
tacValid, 402
LTEWCDMACellInfo
LTEInfoNeighboringWCDMA, 391
nas_LTEInfoNeighboringWCDMA, 456
Lac
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
lac
GERANInfo, 250
GSMSysInfo, 293
LTESysInfo, 401
nas_GERANInfo, 434
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_UMTSInfo, 511
nas_WCDMASysInfo, 520
UMTSInfo, 981
WCDMASysInfo, 1292
lac1
nas_OperatorPLMNData, 474
OperatorPLMNData, 576
lac2
nas_OperatorPLMNData, 474
OperatorPLMNData, 576
lacValid
GSMSysInfo, 294
LTESysInfo, 401
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_WCDMASysInfo, 520
WCDMASysInfo, 1292
language
CDMABroadcastConfig, 162
sms_CDMABroadcastConfig, 856
lastCallDataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1177
lastCallDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 1176
lastCallRXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1177
lastCallTXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1177
LastErrCode
DataStatusDetail, 214
nas_DataStatusDetail, 426
Latency
unpack_qos_swiQosFlow_t, 1107
Latitude
GPSStateInfo, 287
latitude
pack_loc_SLQSLOCInjectPosition_t, 600
leapSeconds
nas_qaQmi3Gpp2TimeZone, 485
qaQmi3Gpp2TimeZone, 734
leaseState
wds_DHCPLeaseStateTlv, 1295
len
BdsSVInfo, 141
loc_BdsSVInfo, 359
loc_SVInfo, 369
SVInfo, 891

t_Sv, 913
unpack_nas_GetSignalStrengths_t, 1052
length
readTransparentInfo, 774
SMSCAddress, 870
sMSCAddress, 870
SMSEtwsMessage, 872
sMSEtwsMessage, 871
SMSTransferRouteMTMessage, 882
sMSTransferRouteMTMessage, 883
uim_readTransparentInfo, 949
Level
GetM2MAudioVolumeResp, 274
SetM2MAudioVolumeReq, 821
LibPackGPRSRequestedQoS, 328
delayClass, 329
meanThroughputClass, 329
peakThroughputClass, 329
precedenceClass, 329
reliabilityClass, 329
LibPackQosClassID, 351
gDlBitRate, 352
gUlBitRate, 352
maxDlBitRate, 352
maxUlBitRate, 352
QCI, 352
LibPackTFTIDParams, 352
destPortRangeEnd, 353
destPortRangeStart, 354
eValid, 354
filterId, 354
flowLabel, 354
IPSECSPI, 354
ipVersion, 354
nextHeader, 354
pSourceIP, 354
sourceIPMask, 354
srcPortRangeEnd, 354
srcPortRangeStart, 354
tosMask, 354
LibPackUMTSQoS, 354
deliveryErrSDU, 356
grntDownlinkBitrate, 356
grntUplinkBitrate, 356
maxDownlinkBitrate, 356
maxSDUSize, 356
maxUplinkBitrate, 356
qosDeliveryOrder, 356
resBerRatio, 356
sduErrorRatio, 356
trafficClass, 356
trafficPriority, 356
transferDelay, 356
LibPackUMTSReqQoSSigInd, 356
SigInd, 357
UMTSReqQoS, 357
LibPackprofile_3GPP, 340
pAPNClass, 345
pAPNDisabledFlag, 345
pAPNName, 345
pAPNnameSize, 345
pAddrAllocPref, 345
pAuthenticationPref, 345
pGPRSMinimumQoS, 345
pGPRSRequestedQos, 345
pIPv4AddrPref, 345
pIPv6AddPref, 345
pImCnFlag, 345
pPDNInactivTimeout, 345
pPDPtype, 345
pPassword, 345
pPasswordSize, 345
pPcscfAddrUsingDhcp, 345
pPcscfAddrUsingPCO, 345
pPdpAccessConFlag, 345
pPdpContext, 345
pPdpDataCompType, 345
pPdpHdrCompType, 345
pPriDNSIPv4AddPref, 345
pPriDNSIPv6addpref, 345
pPrimaryID, 345
pProfilename, 346
pProfilenameSize, 346
pQosClassID, 346
pSecDNSIPv4AddPref, 346
pSecDNSIPv6addpref, 346
pSecondaryFlag, 346
pTFTID1Params, 346
pTFTID2Params, 346
pUMTSMinQoS, 346
pUMTSMinQosSigInd, 346
pUMTSReqQoS, 346
pUMTSReqQoSSigInd, 346
pUsername, 346
pUsernameSize, 346
LibPackprofile_3GPP2, 346
pAPNClass3GPP2, 350
pAPNEnabled3GPP2, 350
pAllowLinger, 350
pApnString, 350
pApnStringSize, 350
pAppPriority, 350
pAppType, 350
pAuthPassword, 350
pAuthPassword_tSize, 350
pAuthProtocol, 350
pAuthRetryCount, 351
pAuthTimeout, 351
pDataMode, 351
pDataRate, 351
pIpcpAckTimeout, 351
pIpcpCreqRetryCount, 351
pIsPcscfAddressNedded, 351
pLcpAckTimeout, 351
pLcpCreqRetryCount, 351
pNegoDnsSrvrPref, 351

pPDNInactivTimeout3GPP2, 351
pPdnType, 351
pPppSessCloseTimer1x, 351
pPppSessCloseTimerDO, 351
pPriV6DnsAddress, 351
pPrimaryV4DnsAddress, 351
pRATType, 351
pSecV6DnsAddress, 351
pSecondaryV4DnsAddress, 351
pUserId, 351
pUserIdSize, 351
LibpackProfile3GPP, 329
pAPNClass, 334
pAPNDisabledFlag, 334
pAPNName, 334
pAPNnameSize, 334
pAddrAllocPref, 334
pAuthenticationPref, 334
pGPRSMinimumQoS, 334
pGPRSRequestedQos, 334
pIPv4AddrPref, 334
pIPv6AddPref, 334
pImCnFlag, 334
pPDNInactivTimeout, 334
pPDPtype, 334
pPassword, 334
pPasswordSize, 334
pPcscfAddrUsingDhcp, 334
pPcscfAddrUsingPCO, 334
pPdpAccessConFlag, 334
pPdpContext, 334
pPdpDataCompType, 334
pPdpHdrCompType, 334
pPriDNSIPv4AddPref, 334
pPriDNSIPv6addpref, 334
pPrimaryID, 334
pProfilename, 335
pProfilenameSize, 335
pQosClassID, 335
pSecDNSIPv4AddPref, 335
pSecDNSIPv6addpref, 335
pSecondaryFlag, 335
pTFTID1Params, 335
pTFTID2Params, 335
pUMTSMinQoS, 335
pUMTSMinQosSigInd, 335
pUMTSReqQoS, 335
pUMTSReqQoSSigInd, 335
pUsername, 335
pUsernameSize, 335
LibpackProfile3GPP2, 335
pAPNClass3GPP2, 339
pAPNEnabled3GPP2, 339
pAllowLinger, 339
pApnString, 339
pApnStringSize, 339
pAppPriority, 339
pAppType, 339
pAuthPassword, 339
pAuthPasswordSize, 339
pAuthProtocol, 339
pAuthRetryCount, 340
pAuthTimeout, 340
pDataMode, 340
pDataRate, 340
pIpcpAckTimeout, 340
pIpcpCreqRetryCount, 340
pIsPcscfAddressNedded, 340
pLcpAckTimeout, 340
pLcpCreqRetryCount, 340
pNegoDnsSrvrPref, 340
pPDNInactivTimeout3GPP2, 340
pPdnType, 340
pPppSessCloseTimer1x, 340
pPppSessCloseTimerDO, 340
pPriV6DnsAddress, 340
pPrimaryV4DnsAddress, 340
pRATType, 340
pSecV6DnsAddress, 340
pSecondaryV4DnsAddress, 340
pUserId, 340
pUserIdSize, 340
lineCtrlInfo, 357
polarityIncluded, 358
pwrDenialTime, 358
revPolarity, 358
toggleMode, 358
lineValue
pack_voice_SLQSVoiceALSSelectLine_t, 661
voiceALSSelectLineInfo, 1231
linkage
altitudeSrcInfo, 120
altSrcInfo_t, 121
list_type
custSettingList, 206
DMScustSettingList, 225
DMSgetCustomInput, 226
getCustomInput, 260
pack_dms_GetCustFeaturesV2_t, 578
listEntries
FMSPrefImageList, 247
PrefImageList, 718
listSize
FMSImageList, 247
FMSPrefImageList, 247
ImageList, 308
PrefImageList, 718
liteqmi_GetVersion
common.h, 1329
liteqmi_log
common.h, 1329
loc.h
eQMI_LOC_SESS_STATUS_FAILURE, 1389
eQMI_LOC_SESS_STATUS_IN_PROGRESS, 1389
eQMI_LOC_SESS_STATUS_SUCCESS, 1389

eQMI_LOC_SESS_STATUS_TIMEOUT, 1389
loc.h, 1384
LOCEVENTMASKNMEA, 1388
LOCEVENTMASKWIFIREQ, 1389
MAX_TEMP_DATA_LEN, 1389
pack_loc_DeleteAssistData, 1389
pack_loc_EventRegister, 1389
pack_loc_SLQSLOCGetBestAvailPos, 1390
pack_loc_SLQSLOCInjectPosition, 1391
pack_loc_SLQSLOCInjectSensorData, 1391
pack_loc_SLQSLOCInjectUTCTime, 1391
pack_loc_SLQSLOCSetCradleMountConfig, 1392
pack_loc_SetExtPowerState, 1390
pack_loc_SetOperationMode, 1390
pack_loc_Start, 1392
pack_loc_Stop, 1392
unpack_loc_BestAvailPos_Ind, 1393
unpack_loc_DeleteAssistData, 1393
unpack_loc_DeleteAssistData_Ind, 1393
unpack_loc_EngineState_Ind, 1394
unpack_loc_EventRegister, 1394
unpack_loc_GnssSvInfo_Ind, 1394
unpack_loc_PositionRpt_Ind, 1395
unpack_loc_SLQSLOCGetBestAvailPos, 1397
unpack_loc_SLQSLOCInjectPosition, 1397
unpack_loc_SLQSLOCInjectSensorData, 1397
unpack_loc_SLQSLOCInjectUTCTime, 1398
unpack_loc_SLQSLOCSetCradleMountConfig, 1398
unpack_loc_SetExtPowerConfig_Ind, 1395
unpack_loc_SetExtPowerState, 1396
unpack_loc_SetOperationMode, 1396
unpack_loc_SetOperationMode_Ind, 1396
unpack_loc_Start, 1398
unpack_loc_Stop, 1398
loc_BdsSV, 358
id, 358
mask, 358
loc_BdsSVInfo, 358
len, 359
pSV, 359
loc_CellDb, 359
mask, 359
loc_ClkInfo, 359
mask, 360
loc_GnssData, 361
mask, 362
loc_LocApplicationInfo, 363
appNameLength, 364
appProviderLength, 364
appVersionLength, 364
appVersionValid, 364
pAppName, 364
pAppProvider, 364
pAppVersion, 364
loc_SV, 368
id, 369
mask, 369
system, 369
loc_SVInfo, 369
len, 369
pSV, 369
loc_gpsTime, 362
gpsTimeOfWeekMs, 363
gpsWeek, 363
loc_precisionDilution, 364
HDOP, 365
PDOP, 365
VDOP, 365
loc_satelliteInfo, 365
azimuth, 367
elevation, 367
gnssSvId, 367
healthStatus, 367
snr, 367
svInfoMask, 367
svListLen, 367
svStatus, 367
system, 367
validMask, 367
loc_sensorDataUsage, 367
aidingIndicatorMask, 368
usageMask, 368
loc_svUsedforFix, 369
gnssSvUsedList, 370
gnssSvUsedList_len, 370
LocApplicationInfo, 370
appNameLength, 371
appProviderLength, 371
appVersionLength, 371
appVersionValid, 371
pAppName, 371
pAppProvider, 371
pAppVersion, 371
LocDelAssDataReq, 371
pBdsSVInfo, 372
pCellDb, 372
pClkInfo, 372
pGnssData, 372
pSVInfo, 372
LocInjectPositionReq, 374
pAltitudeSrcInfo, 377
pAltitudeWrtEllipsoid, 377
pAltitudeWrtMeanSeaLevel, 377
pHorConfidence, 377
pHorReliability, 378
pHorUncCircular, 378
pLatitude, 378
pLongitude, 378
pPositionSrc, 378
pRawHorConfidence, 378
pRawHorUncCircular, 378
pTimestampAge, 378
pTimestampUtc, 378
pVertConfidence, 378
pVertReliability, 378

pVertUnc, 378
LocInjectSensorDataReq, 378
pAcceleroData, 379
pAcceleroTempData, 379
pAcceleroTimeSrc, 379
pGyroData, 379
pGyroTempData, 379
pGyroTimeSrc, 379
pOpaqueIdentifier, 379
LocSetCradleMountReq, 379
pConfidence, 380
state, 380
localTimeOffset
nas_qaQmi3Gpp2TimeZone, 485
qaQmi3Gpp2TimeZone, 734
Location Service(LOC), 57
logString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1016
logger
common.h, 1327
longName
nas_PLMNNetworkNameData, 483
nasPLMNNameResp, 548
PLMNNetworkNameData, 716
unpack_nas_SLQSGetPLMNName_t, 1059
longNameCI
nasPLMNNameResp, 548
unpack_nas_SLQSGetPLMNName_t, 1059
longNameEn
nasPLMNNameResp, 548
unpack_nas_SLQSGetPLMNName_t, 1059
longNameLen
nas_PLMNNetworkNameData, 483
nasPLMNNameResp, 548
PLMNNetworkNameData, 716
unpack_nas_SLQSGetPLMNName_t, 1059
longNameSB
nasPLMNNameResp, 548
unpack_nas_SLQSGetPLMNName_t, 1059
longNameSpareBits
nas_PLMNNetworkNameData, 483
PLMNNetworkNameData, 716
Longitude
GPSStateInfo, 287
longitude
pack_loc_SLQSLOCInjectPosition_t, 600
loopbackMode
pack_wds_SLQSSSetLoopback_t, 694
loopbackMultiplier
pack_wds_SLQSSSetLoopback_t, 694
LteBandCapability
unpack_dms_SLQSGetBandCapability_t, 1012
LteCQIParm, 382
CQIValueCW0, 383
CQIValueCW1, 383
ValidityCW0, 383
ValidityCW1, 383
lteEARFCN, 383
earfcn0, 383
earfcn1, 384
status, 384
lteEmmDl
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
lteEmmUl
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
lteEsmDl
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
lteEsmUl
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
LteGsmCellInfo
LTEInfoNeighboringGSM, 390
nas_LTEInfoNeighboringGSM, 455
lteGsmCellInfo, 384
cellReselPriority, 385
cells_len, 385
GsmCellInfo, 385
nccPermitted, 385
threshGsmHigh, 385
threshGsmLow, 385
LteNasReleaseInfo
qaGobiApiCbk.h, 1465
LteNasReleaseInfo_s, 391
nas_major, 391
nas_minor, 391
nas_release, 391
ltePCI, 392
earfcn, 392
pci, 392
status, 392
LteQci
unpack_qos_swiQosFlow_t, 1108
lteRsrpDelta
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
lteRsrpinfo
nas_SLQSSignalStrengthsInformation, 500
SLQSSignalStrengthsInformation, 853
lteRsrpinformation, 392
rsrplevel, 393
lteSSInfo, 399
rsrp, 399
rsrq, 399
rssi, 399
snr, 399
LteSccRxInfoResp, 394
pSccRxInfo, 394
lteSnrDelta
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
lteSnrinfo
nas_SLQSSignalStrengthsInformation, 501

SLQSSignalStrengthsInformation, 853
lteSnrinformation, 396
snrlevel, 397
lteWcdmaCellInfo, 402
cellReselPriority, 402
cellsLen, 402
threshXhigh, 402
threshXlow, 403
uarfcn, 403
WCDMACellInfo, 403
ltersrp
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
ltesnr
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063

m_FwBuildId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 893
m_FwImageId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 893
m_PriBuildId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 893
m_PriImageId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 893
m_nCarrierId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 893
m_nFolderId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 893
m_nStorage
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 893
MACIndex
nas_NetworkStatEVDO, 471
NetworkStatEVDO, 565
MAX_BUILD_ID_LEN
dms.h, 1337
MAX_CALL_NO_LEN
qaGobiApiVoice.h, 1801
MAX_CONTENT_LENGTH
qaGobiApiUim.h, 1788
MAX_CUST_ID_LEN
qaGobiApiDms.h, 1548
MAX_FSN_LENGTH
qaGobiApiDms.h, 1548
MAX_ICCID_LENGTH
qaGobiApiUim.h, 1788
uim.h, 1918
MAX_MSE_TWS_MSG
sms.h, 1884
MAX_NO_OF_CALLS
qaGobiApiCbk.h, 1462
qaGobiApiVoice.h, 1801
MAX_NO_OF_FILES
qaGobiApiCbk.h, 1462
MAX_NO_OF_SLOTS
qaGobiApiCbk.h, 1462
qaGobiApiUim.h, 1788
uim.h, 1918
MAX_NO_OF_UUSINFO
qaGobiApiCbk.h, 1462
MAX_PATH_LENGTH
qaGobiApiCbk.h, 1462
qaGobiApiUim.h, 1788
MAX_PILOT_SETS
qaGobiApiNas.h, 1624
MAX_PUK_LENGTH
qaGobiApiUim.h, 1788
MAX_SENSOR_DATA_LEN
loc.h, 1389
MAX_SLOTS_STATUS
qaGobiApiUim.h, 1788
uim.h, 1918
MAX_SMS_LIST_SIZE
sms.h, 1884
MAX_SMS_ROUTES
qaGobiApiSms.h, 1696
MAX_TEMP_DATA_LEN
loc.h, 1389
qaGobiApiLoc.h, 1613
MAXUSSDLENGTH
qaGobiApiCbk.h, 1462
qaGobiApiVoice.h, 1801
MAXVOICEUSSDLENGTH
voice.h, 1937
MCC
SlqsNas3GppNetworkRAT, 87
CDMASysInfo, 173
CDMASysInfoExt, 174
currentPLMN, 198
GSMSysInfo, 294
LTESysInfo, 401
nas_CDMASysInfo, 418
nas_CDMASysInfoExt, 419
nas_currentPLMN, 423
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_QmiNas3GppNetworkInfo, 486
nas_QmiNas3GppNetworkRAT, 487
nas_QmisNasPcsDigit, 487
nas_WCDMASysInfo, 520
SlqsNas3GppNetworkInfo, 842
SlqsNasPcsDigit, 843
unpack_nas_GetServingNetwork_t, 1051
WCDMASysInfo, 1292
MDMCallDuration
ConnectionStatus, 185
connectionStatus, 185
MDMConnStatus
ConnectionStatus, 185
connectionStatus, 185

MEID_MAX_SIZE
dms.h, 1337
MEIDString
unpack_dms_GetDeviceSerialNumbers_t, 996
MIN
unpack_dms_GetVoiceNumber_t, 1003
MINREQBKLEN
common.h, 1327
MMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1114
MNC
SlqsNas3GppNetworkRAT, 87
CDMASysInfo, 173
currentPLMN, 198
GSMSysInfo, 294
LTESysInfo, 401
nas_CDMASysInfo, 418
nas_currentPLMN, 423
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_QmiNas3GppNetworkInfo, 486
nas_QmiNas3GppNetworkRAT, 487
nas_QmisNasPcsDigit, 487
nas_WCDMASysInfo, 520
SlqsNas3GppNetworkInfo, 842
SlqsNasPcsDigit, 843
unpack_nas_GetServingNetwork_t, 1051
WCDMASysInfo, 1292
MNRInfo, 405
mcc, 406
mnc, 406
rat, 406
MPTlv
NASQmiCbkNasSystemSelPrefInd, 549
MSGID_AND_LEN
common.h, 1327
MSGID_DONT_CARE
common.h, 1327
MTMessageInfo
newMTMessageTlv, 566
manString
pack_dms_SLQSSwiSetHostDevInfo_t, 583
unpack_dms_SLQSSwiGetHostDevInfo_t, 1017
manufacturer
unpack_dms_GetManufacturer_t, 999
Mask
getDUNCallInfoReq, 261
pack_wds_SLQSGetDUNCallInfo_t, 687
mask
BdsSV, 140
CellDb, 175
ClkInfo, 181
GnssData, 281
IPv6TrafCls, 328
loc_BdsSV, 358
loc_CellDb, 359
loc_ClkInfo, 360
loc_GnssData, 362
loc_SV, 369
SV, 891
Tos, 927
unpack_qos_IPv6TrafCls_t, 1092
unpack_qos_Tos_t, 1109
max
_SLQSAVMSSettingsPeriodsInfo, 87
max_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1176
WDSSWICurrentChannelRates, 1324
max_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1176
WDSSWICurrentChannelRates, 1324
max_dist
pack_swiloc_SwiLocSetAutoStart_t, 644
SwiLocGetAutoStartResp, 895
SwiLocSetAutoStartReq, 896
unpack_swiloc_SwiLocGetAutoStart_t, 1122
max_dist_reported
SwiLocGetAutoStartResp, 895
unpack_swiloc_SwiLocGetAutoStart_t, 1122
max_time
pack_swiloc_SwiLocSetAutoStart_t, 644
SwiLocGetAutoStartResp, 895
SwiLocSetAutoStartReq, 896
unpack_swiloc_SwiLocGetAutoStart_t, 1122
max_time_reported
SwiLocGetAutoStartResp, 895
unpack_swiloc_SwiLocGetAutoStart_t, 1122
MaxAllowedPktSz
unpack_qos_swiQosFlow_t, 1108
MaxChanRxRate
ChannelRate, 178
dunchannelRate, 232
MaxChanTxRate
ChannelRate, 178
dunchannelRate, 232
maxChannelRXRate
unpack_wds_GetConnectionRate_t, 1164
maxChannelTXRate
unpack_wds_GetConnectionRate_t, 1164
maxDlBitRate
LibPackQosClassID, 352
QosClassID, 769
maxDownlinkBitrate
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
maxImages
FMSImageIDEntries, 246
ImageIDEntries, 308
maxMitigationLevel
mitigationDevList, 405
MaxRXChannelRate
unpack_dms_GetDeviceCap_t, 994

maxRxChannelRate
unpack_dms_GetDeviceCapabilities_t, 995
maxSDUSize
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
maxStorageSize
sms_maxStorageSizeResp, 860
smsMaxStorageSizeResp, 876
MaxTXChannelRate
unpack_dms_GetDeviceCap_t, 994
maxTxChannelRate
unpack_dms_GetDeviceCapabilities_t, 995
maxUlBitRate
LibPackQosClassID, 352
QosClassID, 769
maxUplinkBitrate
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
mcTimeStamp
cdmaMsgDecodingParams, 166
mcc
CSGID, 193
MNRInfo, 406
nas_CSGID, 423
nas_MNRInfo, 467
nas_netSelectionPref, 468
nas_OperatorPLMNData, 474
nasPLMNNameReq, 546
netSelectionPref, 559
OperatorPLMNData, 577
pack_nas_SLQSGetPLMNName_t, 611
PlmnID, 714
unpack_nas_GetHomeNetwork_t, 1048
mccM
minBasedIMSI, 405
nas_minBasedIMSI, 466
mccT
nas_trueIMSI, 509
trueIMSI, 930
mdmCallDurationActive
unpack_wds_SLQSGetDUNCallInfo_t, 1177
MdmConnStatus
DUNCallInfoInd, 231
meanThroughputClass
GPRSQoS, 283
GPRSRequestedQoS, 284
LibPackGPRSRequestedQoS, 329
wds_GPRSQoS, 1300
meid
unpack_dms_GetSerialNumbers_t, 1003
meidLength
_SLQSSwiGetSerialNoExtParams, 97
meidSize
serialNumbersInfo, 801
unpack_dms_GetDeviceSerialNumbers_t, 996
meidString
unpack_dms_SLQSSwiGetSerialNoExt_t, 1018
message
unpack_sms_SLQSGetSMS_t, 1116
message_type
NASOTAMessageTlv, 541
messageClass
sms_routeEntry, 863
smsRouteEntry, 881
messageFailureCode
slqssendsmsparams_s, 846
unpack_sms_SendSMS_t, 1113
messageFormat
pack_sms_SaveSMS_t, 635
pack_sms_SendSMS_t, 635
slqssendasyncsmsparams_s, 845
slqssendsmsparams_s, 846
sms_sendAsyncsmsParams, 865
unpack_sms_SLQSGetSMS_t, 1116
messageID
slqssendsmsparams_s, 846
SMSAsyncRawSend_s, 869
unpack_sms_SendSMS_t, 1113
messageId
cdmaMsgEncodingParams, 168
messageIndex
pack_sms_SLQSGetSMS_t, 638
pack_sms_SLQSModifySMSStatus_t, 640
qmiSmsMessageList, 763
SMSMTMessage, 878
sMSMTMessage, 878
messageLength
cdmaMsgDecodingParams, 166
messageList
unpack_sms_SLQSGetSMSList_t, 1117
messageListSize
unpack_sms_SLQSGetSMSList_t, 1117
messageMode
SMSMemoryInfo, 876
SMSMessageMode, 877
sMSMessageMode, 877
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1120
MessageModeInfo
messageModeTlv, 403
messageModeTlv, 403
MessageModeInfo, 403
TlvPresent, 403
messageSize
pack_sms_SaveSMS_t, 635
pack_sms_SendSMS_t, 635
slqssendasyncsmsparams_s, 845
slqssendsmsparams_s, 846
sms_sendAsyncsmsParams, 865
unpack_sms_SLQSGetSMS_t, 1116
wcdmaMsgEncodingParams, 1288
messageTag

pack_sms_SLQSModifySMSStatus_t, 640
qmiSmsMessageList, 763
unpack_sms_SLQSGetSMS_t, 1116
messageType
sms_routeEntry, 863
smsRouteEntry, 881
messageWaitingInfoContent, 403
activeInd, 404
msgCount, 404
msgType, 404
MicMute
GetAudioProfileResp, 256
GetM2MAudioProfileResp, 273
GetM2MAVMuteResp, 275
SetAudioProfileReq, 810
SetM2MAVMuteReq, 821
min
_SLQSAVMSSettingsPeriodsInfo, 87
minBasedIMSI, 404
imsiM1112, 405
imsiMS1, 405
imsiMS2, 405
mccM, 405
MinPolicedPktSz
unpack_qos_swiQosFlow_t, 1108
minSize
unpack_dms_GetVoiceNumber_t, 1003
minute
nas_timeInfo, 508
nas_UniversalTime, 514
timeInfo, 922
UniversalTime, 991
mipMode
unpack_wds_GetMobileIP_t, 1168
mipStatus
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
mipstatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
mitigationDevID
TmdDeRegNotMitigationLvlReq, 923
TmdGetMitigationLvlReq, 924
TmdMitigationLvlIndReq, 925
TmdRegNotMitigationLvlReq, 926
mitigationDevIDLen
TmdDeRegNotMitigationLvlReq, 923
TmdGetMitigationLvlReq, 924
TmdMitigationLvlIndReq, 925
TmdRegNotMitigationLvlReq, 926
mitigationDevId
mitigationDevList, 405
mitigationDevIdLen
mitigationDevList, 405
MitigationDevInfo
QmiCbkTmdMitiLvlRptInd, 759
mitigationDevList, 405
maxMitigationLevel, 405
mitigationDevId, 405
mitigationDevIdLen, 405
mnc
CSGID, 193
MNRInfo, 406
nas_CSGID, 423
nas_MNRInfo, 467
nas_netSelectionPref, 468
nas_OperatorPLMNData, 474
nasPLMNNameReq, 546
netSelectionPref, 559
OperatorPLMNData, 577
pack_nas_SLQSGetPLMNName_t, 611
PlmnID, 714
unpack_nas_GetHomeNetwork_t, 1048
mncPcsDigits
CSGID, 193
nas_CSGID, 423
mobileCountryCode
SMSEtwsPlmn, 873
sMSEtwsPlmn, 873
mobileIP
pack_wds_SLQSSetWdsEventCallback_t, 693
mobileNetworkCode
SMSEtwsPlmn, 873
sMSEtwsPlmn, 873
mode
callInfo, 153
pack_dms_SetEventReport_t, 581
pack_dms_SetPower_t, 581
pack_loc_SetOperationMode_t, 595
pack_sms_SLQSGetSmsBroadcastConfig_t, 638
pack_sms_SLQSSetSmsBroadcastActivation_t, 641
pack_sms_SLQSSetSmsBroadcastConfig_t, 642
pack_wds_SetMobileIP_t, 683
uim_refreshevent, 951
UIMRefreshEvent, 971
voice_callInfo, 1212
ModePref
NASModePreferenceTlv, 536
modelString
pack_dms_SLQSSwiSetHostDevInfo_t, 583
unpack_dms_SLQSSwiGetHostDevInfo_t, 1017
modelid
unpack_dms_GetModelID_t, 1000
modelid_str
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
modemMode
CommInfo, 184
nas_CommInfo, 422
modemTempNotification
qaGobiApiCbk.h, 1466
ModemTempState
_modemTempNotification, 68
ModemTemperature
_modemTempNotification, 68

modemindex
pack_fms_SetImagesPreference_t, 590
ModifyProfileIn, 406
curProfile, 407
pProfileID, 407
pProfileType, 407
ModifyProfileOut, 407
pExtErrorCode, 407
month
nas_timeInfo, 508
nas_UniversalTime, 514
timeInfo, 922
UniversalTime, 991
msgCount
messageWaitingInfoContent, 404
sms_messageWaitingInfoContent, 860
msgDelFailureCause
SMSAsyncRawSend_s, 869
msgDelFailureType
SMSAsyncRawSend_s, 869
msgProtocol
sms_msgProtocolResp, 861
smsMsgprotocolResp, 878
msgType
messageWaitingInfoContent, 404
sms_messageWaitingInfoContent, 860
msgWaitInfo
getMsgWaitingInfo, 277
msgWaitingInfo, 408
sms_getMsgWaitingInfo, 857
msgWaitingInfo, 407
msgWaitInfo, 408
numInstances, 408
msgid
pack_qmi_t, 631
unpack_qmi_t, 1090
msgtype
common.h, 1328
Mtu
unpack_wds_SLQSGetRuntimeSettings_t, 1179
MultDisc
nas_protocolSubtypeElement, 484
protocolSubtypeElement, 733
multiplier
pktErrRate, 713
unpack_qos_pktErrRate_t, 1092

NAI
unpack_wds_GetMobileIPProfile_t, 1168
NAM_NAME_LENGTH
qaGobiApiNas.h, 1624
NAS_NAM_NAME_LENGTH
nas.h, 1407
NAS_PLMN_LENGTH
nas.h, 1407
NAS_SRV
qaGobiApiCbk.h, 1462
NASBandPreferenceTlv, 521
band_pref, 521
TlvPresent, 521
NASEmergencyModeTlv, 523
EmerMode, 523
TlvPresent, 523
NASGWAcqOrderPrefTlv, 531
GWAcqOrderPref, 531
TlvPresent, 531
NASLTEBandPreferenceTlv, 535
LTEBandPref, 536
TlvPresent, 536
NASLteNasReleaseInfoTlv, 536
nas_major, 536
nas_minor, 536
nas_release, 536
TlvPresent, 536
NASModePreferenceTlv, 536
ModePref, 536
TlvPresent, 536
NASNetSelPreferenceTlv, 536
NetSelPref, 536
TlvPresent, 536
NASOTAMessageTlv, 541
data_buf, 541
data_len, 541
message_type, 541
TlvPresent, 541
NASPRLPreferenceTlv, 548
PRLPref, 548
TlvPresent, 548
NASPhyCaAggPcellInfo, 541
dl_bw_value, 542
freq, 542
iLTEbandValue, 542
pci, 542
TlvPresent, 542
NASPhyCaAggScellDlBw, 542
dl_bw_value, 542
TlvPresent, 543
NASPhyCaAggScellIndType, 543
freq, 544
pci, 544
scell_state, 544
TlvPresent, 544
NASPhyCaAggScellIndex, 543
scell_idx, 543
TlvPresent, 543
NASPhyCaAggScellInfo, 544
dl_bw_value, 545
freq, 545
iLTEbandValue, 545
pci, 545
scell_state, 545
TlvPresent, 545
NASQmiCbkNasSwiOTAMessageInd, 548
nasRelInfoTlv, 549
otaMsgTlv, 549
timeTlv, 549
NASQmiCbkNasSystemSelPrefInd, 549

BPTlv, 549
EMTlv, 549
GWAOPTlv, 549
LBPTlv, 549
MPTlv, 549
NSPTlv, 549
PRLPTlv, 549
RPTlv, 549
SDPTlv, 549
NASRoamPreferenceTlv, 549
RoamPref, 550
TlvPresent, 550
NASServDomainPrefTlv, 550
SrvDomainPref, 550
TlvPresent, 550
NASServingSystemInfo, 550
csAttachState, 551
hdrPersonality, 551
psAttachState, 551
radioInterfaceList, 551
radioInterfaceNo, 551
registrationState, 551
selectedNetwork, 551
NASTimeInfoTlv, 558
time, 558
TlvPresent, 558
NSPTlv
NASQmiCbkNasSystemSelPrefInd, 549
NSSAudioCtrl, 568
downLink, 568
upLink, 568
NUM_OF_SET
qaGobiApiCbk.h, 1462
qaGobiApiSms.h, 1696
NWProfile, 568
pProfSz, 568
pProfValues, 569
tech, 569
NWQoSStatus
unpack_qos_SLQSQosGetNetworkStatus_t, 1095
NWRegStat
_transNWRegInfoNotification, 112
naiSize
unpack_wds_GetMobileIPProfile_t, 1168
namID
airTimer, 117
nasGet3GPP2SubscriptionInfoReq, 523
pack_nas_SLQSNasGet3GPP2Subscription_t, 617
prefVoiceSO, 720
roamTimer, 783
voice_airTimer, 1195
voice_prefVoiceSO, 1227
voice_roamTimer, 1229
namName, 408
namName, 408
namNameLen, 408
namName, 408
nas_namName, 467
namNameLen
namName, 408
nas_namName, 467
Name
unpack_nas_GetServingNetwork_t, 1051
name
unpack_nas_GetHomeNetwork_t, 1048
unpack_wds_GetDefaultProfile_t, 1166
nameLen
remotePartyName, 778
voice_remotePartyName, 1228
namePI
remotePartyName, 778
voice_remotePartyName, 1228
nameSize
unpack_nas_GetHomeNetwork3GPP2_t, 1047
unpack_nas_GetServingNetwork_t, 1051
nameString
pack_dms_SLQSSwiSetOSInfo_t, 584
unpack_dms_SLQSSwiGetOSInfo_t, 1018
namelength
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
unpack_omaDmFotaTlv_t, 1087
namesize
unpack_wds_GetDefaultProfile_t, 1166
nas.h
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6, 1407
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75, 1407
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1407
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1407
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6, 1407
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75, 1407
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE, 1408
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE, 1408

eNAS_LTE_CPHY_SCELL_STATE_DECONFIG-
URED_LITE, 1408
nas.h, 1399
NAS_PLMN_LENGTH, 1407
pack_nas_GetACCOLC, 1408
pack_nas_GetANAAAAuthenticationStatus, 1408
pack_nas_GetCDMANetworkParameters, 1408
pack_nas_GetHomeNetwork, 1409
pack_nas_GetHomeNetwork3GPP2, 1409
pack_nas_GetNetworkPreference, 1409
pack_nas_GetRFInfo, 1409
pack_nas_GetServingNetwork, 1410
pack_nas_GetServingNetworkCapabilities, 1410
pack_nas_GetSignalStrengths, 1410
pack_nas_InitiateDomainAttach, 1411
pack_nas_PerformNetworkScan, 1411
pack_nas_SLQSConfigSigInfo, 1413
pack_nas_SLQSGetErrorRate, 1413
pack_nas_SLQSGetNetworkTime, 1414
pack_nas_SLQSGetOperatorNameData, 1414
pack_nas_SLQSGetPLMNName, 1414
pack_nas_SLQSGetServingSystem, 1415
pack_nas_SLQSGetSignalStrength, 1415
pack_nas_SLQSGetSysInfo, 1415
pack_nas_SLQSGetSysSelectionPref, 1416
pack_nas_SLQSInitiateNetworkRegistration, 1416
pack_nas_SLQSNASSwiGetChannelLock, 1418
pack_nas_SLQSNASSwiSetChannelLock, 1419
pack_nas_SLQSNasConfigSigInfo2, 1416
pack_nas_SLQSNasGet3GPP2Subscription, 1416
pack_nas_SLQSNasGetCellLocationInfo, 1417
pack_nas_SLQSNasGetHDRColorCode, 1417
pack_nas_SLQSNasGetSigInfo, 1417
pack_nas_SLQSNasGetTxRxInfo, 1418
pack_nas_SLQSNasIndicationRegisterExt, 1418
pack_nas_SLQSNasSwiIndicationRegister, 1419
pack_nas_SLQSNasSwiModemStatus, 1419
pack_nas_SLQSSetBandPreference, 1420
pack_nas_SLQSSetSignalStrengthsCallback,
1420
pack_nas_SLQSSetSysSelectionPref, 1420
pack_nas_SLQSSwiGetHDRPersonality, 1421
pack_nas_SLQSSwiGetHDRProtSubtype, 1421
pack_nas_SLQSSwiGetHRPDStats, 1421
pack_nas_SLQSSwiGetLteCQI, 1422
pack_nas_SLQSSwiGetLteSccRxInfo, 1422
pack_nas_SLQSSwiNetworkDebug, 1422
pack_nas_SLQSSwiPSDetach, 1423
pack_nas_SetACCOLC, 1411
pack_nas_SetCDMANetworkParameters, 1412
pack_nas_SetLURejectCallback, 1412
pack_nas_SetNetworkPreference, 1412
pack_nas_SetRFInfoCallback, 1412
pack_nas_SlqsGetLTECphyCAInfo, 1413
unpack_nas_GetACCOLC, 1423
unpack_nas_GetANAAAAuthenticationStatus,
1423
unpack_nas_GetCDMANetworkParameters, 1424
unpack_nas_GetHomeNetwork, 1424
unpack_nas_GetHomeNetwork3GPP2, 1424
unpack_nas_GetNetworkPreference, 1425
unpack_nas_GetRFInfo, 1425
unpack_nas_GetServingNetwork, 1425
unpack_nas_GetServingNetworkCapabilities,
1425
unpack_nas_GetSignalStrengths, 1426
unpack_nas_InitiateDomainAttach, 1426
unpack_nas_PerformNetworkScan, 1426
unpack_nas_SLQSConfigSigInfo, 1429
unpack_nas_SLQSGetErrorRate, 1429
unpack_nas_SLQSGetNetworkTime, 1430
unpack_nas_SLQSGetOperatorNameData, 1430
unpack_nas_SLQSGetPLMNName, 1430
unpack_nas_SLQSGetServingSystem, 1431
unpack_nas_SLQSGetSignalStrength, 1431
unpack_nas_SLQSGetSysInfo, 1431
unpack_nas_SLQSGetSysSelectionPref, 1432
unpack_nas_SLQSInitiateNetworkRegistration,
1432
unpack_nas_SLQSNASSwiGetChannelLock, 1435
unpack_nas_SLQSNASSwiSetChannelLock, 1436
unpack_nas_SLQSNasConfigSigInfo2, 1432
unpack_nas_SLQSNasGet3GPP2Subscription,
1432
unpack_nas_SLQSNasGetCellLocationInfo, 1433
unpack_nas_SLQSNasGetHDRColorCode, 1433
unpack_nas_SLQSNasGetSigInfo, 1433
unpack_nas_SLQSNasGetTxRxInfo, 1434
unpack_nas_SLQSNasIndicationRegisterExt,
1434
unpack_nas_SLQSNasNetworkTimeCallBack_ind,
1434
unpack_nas_SLQSNasSigInfoCallback_ind, 1435
unpack_nas_SLQSNasSwiIndicationRegister,
1435
unpack_nas_SLQSNasSwiModemStatus, 1436
unpack_nas_SLQSNasSwiOTAMessageCallback-
_ind, 1436
unpack_nas_SLQSNasSysInfoCallback_ind, 1437
unpack_nas_SLQSNasTimerCallback_ind, 1437
unpack_nas_SLQSSetBandPreference, 1437
unpack_nas_SLQSSetSignalStrengthsCallback,
1437
unpack_nas_SLQSSetSysSelectionPref, 1438
unpack_nas_SLQSSetSysSelectionPrefCallBack-
_ind, 1438
unpack_nas_SLQSSwiGetHDRPersonality, 1438
unpack_nas_SLQSSwiGetHDRProtSubtype, 1439
unpack_nas_SLQSSwiGetHRPDStats, 1439
unpack_nas_SLQSSwiGetLteCQI, 1439
unpack_nas_SLQSSwiGetLteSccRxInfo, 1440
unpack_nas_SLQSSwiNetworkDebug, 1440
unpack_nas_SLQSSwiPSDetach, 1440
unpack_nas_SetACCOLC, 1427
unpack_nas_SetCDMANetworkParameters, 1427

unpack_nas_SetDataCapabilitiesCallback_ind, 1427
unpack_nas_SetEventReportInd, 1428
unpack_nas_SetLURejectCallback, 1428
unpack_nas_SetNasLTECphyCaIndCallback_ind, 1428
unpack_nas_SetNetworkPreference, 1428
unpack_nas_SetRFInfoCallback, 1428
unpack_nas_SetRoamingIndicatorCallback_ind, 1429
unpack_nas_SetServingSystemCallback_ind, 1429
unpack_nas_SlqsGetLTECphyCAInfo, 1430
nas_ActPilotPNElement, 409
ActSetPilotPN, 409
ActSetPilotPNStrength, 409
nas_AddCDMASysInfo, 409
geoSysIdx, 410
regPrd, 410
nas_AddSysInfo, 410
cellBroadcastCap, 410
geoSysIdx, 410
nas_CDMAChannel, 412
priChA, 413
priChB, 413
secChA, 413
secChB, 413
nas_CDMAECIOThresh, 413
CDMAECIOThreshListLen, 413
pCDMAECIOThreshList, 413
nas_CDMAInfo, 413
baseId, 414
baseLat, 414
baseLong, 414
nid, 414
refpn, 414
sid, 414
nas_CDMARSSIThresh, 414
CDMARSSIThreshListLen, 415
pCDMARSSIThreshList, 415
nas_CDMASysInfo, 415
baseId, 418
baseLat, 418
baseLong, 418
bsInfoValid, 418
bsPRev, 418
bsPRevValid, 418
ccsSupported, 418
ccsSupportedValid, 418
cdmaSysIdValid, 418
isSysPrlMatch, 418
isSysPrlMatchValid, 418
MCC, 418
MNC, 418
networkID, 418
networkIdValid, 418
pRevInUse, 418
pRevInUseValid, 419
packetZone, 418
packetZoneValid, 418
sysInfoCDMA, 419
systemID, 419
nas_CDMASysInfoExt, 419
imsi_11_12, 419
MCC, 419
nas_CSGID, 422
id, 422
mcc, 423
mnc, 423
mncPcsDigits, 423
rat, 423
nas_CallBarringSysInfo, 411
csBarStatus, 411
psBarStatus, 411
nas_CommInfo, 420
imsRegState, 422
modemMode, 422
psState, 422
systemMode, 422
temperature, 422
nas_DRCParams, 429
DRCCover, 430
DRCValue, 430
nas_DataStatusDetail, 424
IPAddress, 426
LastErrCode, 426
nas_DeviceConfigDetail, 428
Chipset, 429
HWVersion, 429
QLIC, 429
Technology, 429
nas_ECIOThresh, 431
ECIOThresListLen, 431
pECIOThresList, 431
nas_GERANInfo, 432
arfcn, 434
bsic, 434
cellID, 434
insNmrCellInfo, 434
lac, 434
nmrInst, 434
plmn, 434
rxLev, 434
timingAdvance, 434
nas_GSMRSSIThresh, 436
GSMRSSIThreshListLen, 436
pGSMRSSIThreshList, 436
nas_GSMSrvStatusInfo, 436
isPrefDataPath, 437
srvStatus, 437
trueSrvStatus, 437
nas_GSMSysInfo, 437
cellId, 439
cellIdValid, 439
dtmSupp, 439
dtmSuppValid, 439

egprsSupp, 440
egprsSuppValid, 440
lac, 440
lacValid, 440
MCC, 440
MNC, 440
networkIdValid, 440
regRejectInfoValid, 440
rejCause, 440
rejectSrvDomain, 440
sysInfoGSM, 440
nas_HDRECIOThresh, 440
HDRECIOThreshListLen, 440
pHDRECIOThreshList, 440
nas_HDRIOThresh, 440
HDRIOThreshListLen, 441
pHDRIOThreshList, 441
nas_HDRRSSIThresh, 441
HDRRSSIThreshListLen, 441
pHDRRSSIThreshList, 441
nas_HDRSINRThresh, 442
HDRSINRThresListLen, 442
pHDRSINRThresList, 442
nas_HDRSINRThreshold, 442
HDRSINRThreshListLen, 443
pHDRSINRThreshList, 443
nas_HDRSysInfo, 443
hdrActiveProt, 445
hdrActiveProtValid, 445
hdrPersonality, 445
hdrPersonalityValid, 445
is856SysId, 445
is856SysIdValid, 445
isSysPrlMatch, 445
isSysPrlMatchValid, 445
sysInfoHDR, 445
nas_IOThresh, 447
IOThresListLen, 447
pIOThresList, 447
nas_LTEInfo, 449
band, 451
bandwidth, 451
emmConnState, 451
emmState, 451
emmSubState, 451
RXChan, 451
TXChan, 451
nas_LTEInfoInterfreq, 452
freqsLen, 452
InfoInterfreq, 452
ueInIdle, 452
nas_LTEInfoIntrafreq, 452
CellParams, 454
cellReselPriority, 454
cellsLen, 454
earfcn, 454
globalCellId, 454
plmn, 454
sIntraSearch, 454
sNonIntraSearch, 454
servingCellId, 454
tac, 454
threshServingLow, 454
ueInIdle, 454
nas_LTEInfoNeighboringGSM, 454
freqsLen, 455
LteGsmCellInfo, 455
ueInIdle, 455
nas_LTEInfoNeighboringWCDMA, 455
freqsLen, 456
ueInIdle, 456
nas_LTERSRPThresh, 457
LTERSRPThreshListLen, 457
pLTERSRPThreshList, 457
nas_LTERSRQThresh, 457
LTERSRQThreshListLen, 458
pLTERSRQThreshList, 458
nas_LTERSSIThresh, 458
LTERSSIThreshListLen, 458
pLTERSSIThreshList, 458
nas_LTESNRThresh, 461
LTESNRThresListLen, 461
pLTESNRThresList, 461
nas_LTESNRThreshold, 461
LTESNRThreshListLen, 462
pLTESNRThreshList, 462
nas_LTESigRptCfg, 458
avgPeriod, 459
rptRate, 459
nas_LTESigRptConfig, 459
avgPeriod, 460
rptRate, 460
nas_LTESysInfo, 462
cellId, 464
cellIdValid, 464
lac, 464
lacValid, 464
MCC, 464
MNC, 464
networkIdValid, 464
regRejectInfoValid, 464
rejCause, 464
rejectSrvDomain, 464
sysInfoLTE, 464
tac, 464
tacValid, 464
nas_MNRInfo, 466
mcc, 467
mnc, 467
rat, 467
nas_NetworkStat1x, 468
ActSetCnt, 470
NeighborSetCnt, 470
pActPilotPNElements, 470
pNeighborSetPilotPN, 470
RX_EC_IO, 470

RX_PWR, 470
SO, 470
State, 470
TX_PWR, 470
nas_NetworkStatEVDO, 470
MACIndex, 471
PER, 472
pSectorID, 472
PilotEnergy, 472
RX_PWR, 472
SNR, 472
SectorIDLen, 472
State, 472
nas_OperatorPLMNData, 473
lac1, 474
lac2, 474
mcc, 474
mnc, 474
PLMNRecID, 474
nas_PLMNNetworkName, 481
numInstance, 481
PLMNNetName, 481
nas_PLMNNetworkNameData, 482
codingScheme, 483
countryInitials, 483
longName, 483
longNameLen, 483
longNameSpareBits, 483
shortName, 483
shortNameLen, 483
shortNameSpareBits, 483
nas_PhyCaAggPcellInfo, 475
dl_bw_value, 475
freq, 475
iLTEbandValue, 475
pci, 476
TlvPresent, 476
nas_PhyCaAggScellDlBw, 476
dl_bw_value, 476
TlvPresent, 476
nas_PhyCaAggScellIndType, 477
freq, 477
pci, 477
scell_state, 477
TlvPresent, 477
nas_PhyCaAggScellIndex, 476
scell_idx, 477
TlvPresent, 477
nas_PhyCaAggScellInfo, 477
dl_bw_value, 480
freq, 480
iLTEbandValue, 480
pci, 480
scell_state, 480
TlvPresent, 480
nas_PilotSetData, 480
NumPilots, 480
pPilotSetInfo, 480
nas_PilotSetParams, 480
PilotPN, 481
PilotStrength, 481
PilotType, 481
nas_QmiNas3GppNetworkInfo, 485
Desription, 486
Forbidden, 486
InUse, 486
MCC, 486
MNC, 486
Preferred, 486
Roaming, 486
nas_QmiNas3GppNetworkRAT, 486
MCC, 487
MNC, 487
RAT, 487
nas_QmisNasPcsDigit, 487
includes_pcs_digit, 487
MCC, 487
MNC, 487
nas_RFInfoTlv, 488
activeBandClass, 488
activeChannel, 488
radioInterface, 488
radioInterfaceSize, 488
TlvPresent, 488
nas_RSRPThresh, 489
pRSRPThresList, 490
RSRPThresListLen, 490
nas_RSRQThresh, 491
pRSRQThresList, 491
RSRQThresListLen, 491
nas_RSSIThresh, 491
pRSSIThresList, 492
RSSIThresListLen, 492
nas_RejectReasonTlv, 488
rejectCause, 488
serviceDomain, 488
TlvPresent, 488
nas_RxSigInfo, 493
isRadioTuned, 494
rsrp, 494
rxChainIndex, 494
rxPower, 494
nas_SLQSSignalStrengthsIndReq, 499
ecioDelta, 499
ecioThresholdList, 499
ecioThresholdListLen, 500
ioDelta, 500
lteRsrpDelta, 500
lteSnrDelta, 500
rsrqDelta, 500
rxSignalStrengthDelta, 500
sinrDelta, 500
sinrThresholdList, 500
sinrThresholdListLen, 500
nas_SLQSSignalStrengthsInformation, 500
ecioInfo, 500

errorRateInfo, 500
io, 500
lteRsrpinfo, 500
lteSnrinfo, 501
rsrqInfo, 501
rxSignalStrengthInfo, 501
sinr, 501
nas_SLQSSignalStrengthsTlv, 501
sSLQSSignalStrengthsInfo, 501
TlvPresent, 501
nas_SccRxInfo, 495
numInstances, 495
rsrq, 495
sigInfo, 496
snr, 496
TlvPresent, 496
nas_SignalStrengthTlv, 498
radioInterface, 499
signalStrength, 499
TlvPresent, 499
nas_SrvStatusInfo, 501
isPrefDataPath, 502
srvStatus, 502
nas_TDSCDMAECIOThresh, 504
nas_TDSCDMARSCPThresh, 505
nas_TDSCDMARSSIThresh, 505
nas_TDSCDMASINRCONFThresh, 506
nas_TDSCDMASINRThresh, 506
nas_UMTSInfo, 510
cellID, 511
ecio, 511
geranInst, 511
GeranInstInfo, 511
lac, 511
plmn, 511
psc, 511
rscp, 511
UMTSInstInfo, 512
uarfcn, 511
umtsInst, 511
nas_UMTSinstInfo, 512
umtsEcio, 512
umtsPsc, 512
umtsRscp, 512
umtsUarfcn, 512
nas_UniversalTime, 513
day, 514
dayOfWeek, 514
hour, 514
minute, 514
month, 514
second, 514
year, 514
nas_WCDMAECIOThresh, 515
nas_WCDMAInfoLTENeighborCell, 516
umtsLTENbrCellLen, 517
wcdmaRRCState, 517
nas_WCDMARSSIThresh, 517
nas_WCDMASysInfo, 517
cellId, 520
cellIdValid, 520
hsCallStatus, 520
hsCallStatusValid, 520
hsInd, 520
hsIndValid, 520
lac, 520
lacValid, 520
MCC, 520
MNC, 520
networkIdValid, 520
psc, 520
pscValid, 520
regRejectInfoValid, 520
rejCause, 521
rejectSrvDomain, 521
sysInfoWCDMA, 521
nas_acqOrderPref, 408
acqOrdeLen, 409
pAcqOrder, 409
nas_callBarStatus, 411
csBarStatus, 412
psBarStatus, 412
nas_cellParams, 419
pci, 420
rsrp, 420
rsrq, 420
rssi, 420
srxlev, 420
nas_currentPLMN, 423
MCC, 423
MNC, 423
netDescr, 423
netDescrLength, 424
nas_dataSrvCapabilities, 424
dataCapabilities, 424
dataCapabilitiesLen, 424
nas_detailSvcInfo, 427
hdrHybrid, 428
hdrSrvStatus, 428
isSysForbidden, 428
srvCapability, 428
srvStatus, 428
nas_dirNum, 429
dirNum, 429
dirNumLen, 429
nas_ecioListElement, 430
ecio, 430
radioIf, 430
nas_errorRateListElement, 431
errorRate, 432
radioIf, 432
nas_geranInstInfo, 434
geranArfcn, 434
geranBsicBcc, 434
geranBsicNcc, 435
geranRssi, 435

nas_gsmCellInfo, 435
arfcn, 435
band1900, 436
bsicId, 436
cellIdValid, 436
rssi, 436
srxlev, 436
nas_homeSIDNID, 445
numInstances, 445
SidNid, 445
nas_infoInterFreq, 446
cell_resel_priority, 446
cellInterFreqParams, 446
cells_len, 446
earfcn, 446
threshXHigh, 447
threshXLow, 447
nas_lteEARFCN, 447
earfcn0, 448
earfcn1, 448
status, 448
nas_lteGsmCellInfo, 448
cellReselPriority, 449
cells_len, 449
GsmCellInfo, 449
nccPermitted, 449
threshGsmHigh, 449
threshGsmLow, 449
nas_ltePCI, 456
earfcn, 456
pci, 456
status, 456
nas_lteRsrpinformation, 456
rsrplevel, 457
nas_lteSnrinformation, 460
snrlevel, 460
nas_lteWcdmaCellInfo, 464
cellReselPriority, 465
cellsLen, 465
threshXhigh, 465
threshXlow, 465
uarfcn, 465
WCDMACellInfo, 465
nas_major
LteNasReleaseInfo_s, 391
NASLteNasReleaseInfoTlv, 536
nas_minBasedIMSI, 465
imsiM1112, 466
imsiMS1, 466
imsiMS2, 466
mccM, 466
nas_minor
LteNasReleaseInfo_s, 391
NASLteNasReleaseInfoTlv, 536
nas_namName, 467
namName, 467
namNameLen, 467
nas_netSelectionPref, 467
mcc, 468
mnc, 468
netReg, 468
nas_nmrCellInfo, 472
nmrArfcn, 473
nmrBsic, 473
nmrCellID, 473
nmrLac, 473
nmrPlmn, 473
nmrRxLev, 473
nas_operatorNameString, 473
PLMNName, 473
nas_operatorPLMNList, 474
numInstance, 475
PLMNData, 475
nas_protocolSubtypeElement, 483
AccessMac, 484
AuthProt, 484
ControlMac, 484
EncryptProt, 484
ForwardMac, 484
IdleState, 484
KeyExchange, 484
MultDisc, 484
PhysicalLayer, 485
ReverseMac, 485
SecProt, 485
VirtStream, 485
nas_qaQmi3Gpp2TimeZone, 485
daylightSavings, 485
leapSeconds, 485
localTimeOffset, 485
nas_release
LteNasReleaseInfo_s, 391
NASLteNasReleaseInfoTlv, 536
nas_roamIndList, 489
numInstances, 489
radioInterface, 489
roamIndicator, 489
nas_rsrqInformation, 490
radioIf, 490
rsrq, 491
nas_rxInfo, 492
ecio, 493
isRadioTuned, 493
phase, 493
rscp, 493
rsrp, 493
rxPower, 493
nas_rxSignalStrengthListElement, 494
radioIf, 495
rxSignalStrength, 495
nas_servSystem, 496
csAttachState, 498
numRadioInterfaces, 498
psAttachState, 498
radioInterface, 498
regState, 498

selNetwork, 498
nas_serviceProviderName, 496
displayCondition, 496
spn, 496
spnLength, 496
nas_sidNid, 498
nid, 498
sid, 498
nas_sysInfoCommon, 502
isSysForbidden, 504
isSysForbiddenValid, 504
roamStatus, 504
roamStatusValid, 504
srvCapability, 504
srvCapabilityValid, 504
srvDomain, 504
srvDomainValid, 504
nas_timeInfo, 507
day, 508
dayLtSavingAdj, 508
dayOfWeek, 508
hour, 508
minute, 508
month, 508
radioInterface, 508
second, 508
timeZone, 508
TlvPresent, 508
year, 508
nas_trueIMSI, 508
imsiT1112, 509
imsiTS1, 509
imsiTS2, 509
imsiTaddrNum, 509
mccT, 509
nas_txInfo, 509
isInTraffic, 510
txPower, 510
nas_umtsLTENbrCell, 512
cellIsTDD, 513
earfcn, 513
pci, 513
rsrp, 513
rsrq, 513
srxlev, 513
nas_wcdmaCellInfo, 514
cpich_ecno, 515
cpich_rscp, 515
psc, 515
srxlev, 515
nas_wcdmaUARFCN, 521
status, 521
uarfcn, 521
nasCellLocationInfoResp, 521
pCDMAInfo, 522
pGERANInfo, 522
pLTEInfoInterfreq, 522
pLTEInfoIntrafreq, 522
pLTEInfoNeighboringGSM, 522
pLTEInfoNeighboringWCDMA, 523
pUMTSCellID, 523
pUMTSInfo, 523
pWCDMAInfoLTENeighborCell, 523
nasGet3GPP2SubscriptionInfoReq, 523
namID, 523
nasGet3GPP2SubscriptionInfoResp, 523
pCDMAChannel, 524
pDirNum, 524
pHomeSIDNID, 524
pMinBasedIMSI, 524
pNAMNameInfo, 524
pTrueIMSI, 524
nasGetHDRColorCodeResp, 524
pColorCode, 525
nasGetLTECphyCa, 525
sPhyCaAggPcellInfo, 525
sPhyCaAggScellDlBw, 525
sPhyCaAggScellIndType, 525
sPhyCaAggScellIndex, 525
sPhyCaAggScellInfo, 525
NasGetLTECphyCaInfo, 525
PhyCaAggPcellInfo, 525
PhyCaAggScellDlBw, 525
PhyCaAggScellIndType, 525
PhyCaAggScellIndex, 525
PhyCaAggScellInfo, 526
nasGetLTECphyCaResp, 526
pPhyCaAggPcellInfo, 526
pPhyCaAggScellDlBw, 526
pPhyCaAggScellIndType, 526
pPhyCaAggScellIndex, 526
pPhyCaAggScellInfo, 526
nasGetSigInfoResp, 526
pCDMASSInfo, 527
pGSMSSInfo, 527
pHDRSSInfo, 527
pLTESSInfo, 527
pTDSCDMASigInfoExt, 527
pTDSCDMASigInfoRscp, 527
pWCDMASSInfo, 527
nasGetSysInfoResp, 527
pAddCDMASysInfo, 529
pAddGSMSysInfo, 529
pAddHDRSysInfo, 529
pAddLTESysInfo, 529
pAddWCDMASysInfo, 529
pCDMASrvStatusInfo, 529
pCDMASysInfo, 529
pGSMCallBarringSysInfo, 529
pGSMCipherDomainSysInfo, 529
pGSMSrvStatusInfo, 529
pGSMSysInfo, 529
pHDRSrvStatusInfo, 529
pHDRSysInfo, 529
pLTESrvStatusInfo, 530
pLTESysInfo, 530

pLTEVoiceSupportSysInfo, 530
pWCDMACallBarringSysInfo, 530
pWCDMACipherDomainSysInfo, 530
pWCDMASrvStatusInfo, 530
pWCDMASysInfo, 530
nasGetTxRxInfoReq, 530
radio_if, 530
nasGetTxRxInfoResp, 530
pRXChain0Info, 531
pRXChain1Info, 531
pTXInfo, 531
nasIndicationRegisterReq, 531
pDDTMInd, 534
pDualStandByPrefInd, 534
pErrorRateInd, 534
pHDRNewUATIAssInd, 534
pHDRSessionCloseInd, 534
pLTECphyCa, 534
pManagedRoamingInd, 534
pNetworkRejectInd, 534
pNetworkTimeInd, 534
pServingSystemInd, 534
pSignalStrengthInd, 534
pSubscriptionInfoInd, 534
pSuppressSysInfoInd, 534
pSysInfoInd, 534
pSystemSelectionInd, 534
nasInitNetworkReg, 534
pChangeDuration, 535
pMNRInfo, 535
pMncPcsDigitStatus, 535
regAction, 535
nasNetworkReject, 537
pCsgId, 539
pPlmnId, 539
radioIf, 539
rejectCause, 539
serviceDomain, 539
nasNetworkTime, 539
pDayltSavAdj, 540
pRadioInterface, 540
pTimeZone, 540
universalTime, 540
nasOperatorNameResp, 540
pNITZInformation, 541
pOperatorNameString, 541
pOperatorPLMNList, 541
pPLMNNetworkName, 541
pSrvcProviderName, 541
nasPLMNNameReq, 545
mcc, 546
mnc, 546
pMncPcsStatus, 546
nasPLMNNameResp, 546
longName, 548
longNameCI, 548
longNameEn, 548
longNameLen, 548
longNameSB, 548
shortName, 548
shortNameCI, 548
shortNameEn, 548
shortNameLen, 548
shortNameSB, 548
spn, 548
spnEncoding, 548
spnLength, 548
nasRelInfoTlv
NASQmiCbkNasSwiOTAMessageInd, 549
nasSigInfo, 551
pCDMASigInfo, 552
pGSMSigInfo, 552
pHDRSigInfo, 552
pLTESigInfo, 552
pRscp, 552
pTDSCDMASigInfoExt, 552
pWCDMASigInfo, 552
nasSwiGetChannelLockResp, 552
pLteEARFCN, 553
pLtePCI, 553
pWcdmaUARFCN, 553
NasSwiIndReg, 553
gsmUmtsDl, 554
gsmUmtsUl, 554
lteEmmDl, 554
lteEmmUl, 554
lteEsmDl, 554
lteEsmUl, 554
pRankIndicatorInd, 554
pTimer, 554
nasSwiSetChannelLockReq, 554
pLteEARFCN, 555
pLtePCI, 555
pWcdmaUARFCN, 555
nasSysInfo, 555
pAddCDMASysInfo, 557
pAddGSMSysInfo, 557
pAddHDRSysInfo, 557
pAddLTESysInfo, 557
pAddWCDMASysInfo, 557
pCDMASrvStatusInfo, 557
pCDMASysInfo, 557
pGSMCallBarringSysInfo, 557
pGSMCipherDomainSysInfo, 557
pGSMSrvStatusInfo, 557
pGSMSysInfo, 557
pHDRSrvStatusInfo, 558
pHDRSysInfo, 558
pLTESrvStatusInfo, 558
pLTESysInfo, 558
pLTEVoiceSupportSysInfo, 558
pSysInfoNoChange, 558
pWCDMACallBarringSysInfo, 558
pWCDMACipherDomainSysInfo, 558
pWCDMASrvStatusInfo, 558
pWCDMASysInfo, 558

nasTimers, 558
t3396_apn, 559
t3396_plmn_id, 559
t3396_val, 559
nccPermitted
lteGsmCellInfo, 385
nas_lteGsmCellInfo, 449
NeighborSetCnt
nas_NetworkStat1x, 470
NetworkStat1x, 563
netDescr
currentPLMN, 198
nas_currentPLMN, 423
netDescrLength
currentPLMN, 198
nas_currentPLMN, 424
netInfoLen
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
netReg
nas_netSelectionPref, 468
netSelectionPref, 559
NetSelPref
NASNetSelPreferenceTlv, 536
netSelectionPref, 559
mcc, 559
mnc, 559
netReg, 559
NetStats, 560
rx_bytes, 560
rx_errors, 560
rx_overflows, 560
rx_packets, 560
tx_bytes, 560
tx_errors, 561
tx_overflows, 561
tx_packets, 561
Network Access Service (NAS), 40
NetworkDebugResp, 561
pDataStatusDetail, 561
pDeviceConfigDetail, 561
pNetworkStat1x, 561
pNetworkStatEVDO, 561
pObjectVer, 561
NetworkID
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
networkID
CDMASysInfo, 173
nas_CDMASysInfo, 418
networkIdValid
CDMASysInfo, 173
GSMSysInfo, 294
LTESysInfo, 401
nas_CDMASysInfo, 418
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_WCDMASysInfo, 520
WCDMASysInfo, 1292
networkInfoLen
unpack_wds_SLQSGetCurrDataSystemStat_t, 1175
NetworkStat1x, 562
ActSetCnt, 563
NeighborSetCnt, 563
pActPilotPNElements, 563
pNeighborSetPilotPN, 563
RX_EC_IO, 563
RX_PWR, 563
SO, 563
State, 563
TX_PWR, 563
NetworkStatEVDO, 564
MACIndex, 565
PER, 565
pSectorID, 565
PilotEnergy, 565
RX_PWR, 565
SNR, 565
SectorIDLen, 565
State, 565
NetworkType
CurrNetworkInfo, 200
currNetworkInfo, 201
wds_currNetworkInfo, 1294
NewMMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1114
newMTMessageTlv, 565
MTMessageInfo, 566
TlvPresent, 566
newPINLen
uim_unblockUIMPIN, 957
unblockUIMPIN, 990
newPINVal
uim_unblockUIMPIN, 957
unblockUIMPIN, 990
newPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 673
voiceSetCallBarringPwdInfo, 1269
newPasswdAgain
pack_voice_SLQSVoiceSetCallBarringPassword_t, 673
voiceSetCallBarringPwdInfo, 1269
newPin
pack_dms_UIMUnblockPIN_t, 588
newPwd
newPwdData, 566
newPwdAgain
newPwdData, 566
newPwdData, 566
newPwd, 566
newPwdAgain, 566
newValue
pack_dms_UIMChangePIN_t, 585
nextHeader

LibPackTFTIDParams, 354
TFTIDParams, 920
nid
CDMAInfo, 164
nas_CDMAInfo, 414
nas_sidNid, 498
sidNid, 832
unpack_nas_GetHomeNetwork_t, 1048
NmeaPort
DcsUsbPortNames, 215
nmrArfcn
nas_nmrCellInfo, 473
nmrCellInfo, 567
nmrBsic
nas_nmrCellInfo, 473
nmrCellInfo, 567
nmrCellID
nas_nmrCellInfo, 473
nmrCellInfo, 567
nmrCellInfo, 566
nmrArfcn, 567
nmrBsic, 567
nmrCellID, 567
nmrLac, 567
nmrPlmn, 568
nmrRxLev, 568
nmrInst
GERANInfo, 250
nas_GERANInfo, 434
nmrLac
nas_nmrCellInfo, 473
nmrCellInfo, 567
nmrPlmn
nas_nmrCellInfo, 473
nmrCellInfo, 568
nmrRxLev
nas_nmrCellInfo, 473
nmrCellInfo, 568
noReplyTimer
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1209
voice_callFWInfo, 1210
Non-service specific APIs (SWI), 51
notifType
voiceSUPSNotification, 1282
Notification
_SwiAvmsEventInd, 100
notification
omaDmNotificationsTlv, 575
unpack_omaDmNotificationsTlv_t, 1088
notificationType
SMSEtwsMessage, 872
sMSEtwsMessage, 871
notused
unpack_dms_SetCrashAction_t, 1004
num_instances
_qaQmi3GPP2BroadcastCfgInfo, 71
_qaQmi3GPPBroadcastCfgInfo, 71
custSettingList, 207
DMScustSettingList, 225
sms_qaQmi3GPP2BroadcastCfgInfo, 862
sms_qaQmi3GPPBroadcastCfgInfo, 862
numApp
slotInf, 836
slotInfo, 838
uim_slotInfo, 956
numCrashes
CrashInfo, 188
crashInformation, 190
numEntries
CurrentImgList, 197
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1015
numFeatures
personalizationStatus, 707
uim_personalizationStatus, 948
numFiles
registerRefresh, 777
uim_registerRefresh, 952
NumFlows
unpack_qos_SLQSSetQosEventCallback_ind_t, 1098
numInstance
nas_operatorPLMNList, 475
nas_PLMNNetworkName, 481
operatorPLMNList, 577
PLMNNetworkName, 715
numInstances
arrAlertingPattern, 128
arrAlertingType, 129
arrAlphaID, 129
arrCalledPartyNum, 130
arrCallEndReason, 131
arrCallInfo, 131
arrConnectPartyNum, 132
arrDiagInfo, 132
arrRedirPartyNum, 133
arrRemotePartyName, 133
arrRemotePartyNum, 134
arrSvcOption, 134
arrUUSInfo, 135
DomainNameList, 229
getCallFWExtInfo, 258
getCallFWInfo, 259
getMsgWaitingInfo, 277
homeSIDNID, 304
msgWaitingInfo, 408
nas_homeSIDNID, 445
nas_roamIndList, 489
nas_SccRxInfo, 495
PCSCFFQDNAddressList, 701
PCSCFIPv4ServerAddressList, 701
roamIndList, 782
SccRxInfo, 796
sms_getMsgWaitingInfo, 857
voice_arrAlertingPattern, 1198

voice_arrAlertingType, 1199
voice_arrAlphaID, 1199
voice_arrCalledPartyNum, 1200
voice_arrCallEndReason, 1200
voice_arrCallInfo, 1201
voice_arrConnectPartyNum, 1201
voice_arrDiagInfo, 1202
voice_arrRedirPartyNum, 1202
voice_arrRemotePartyName, 1203
voice_arrRemotePartyNum, 1204
voice_arrSvcOption, 1204
voice_arrUUSInfo, 1205
voice_getCallFWExtInfo, 1222
voice_getCallFWInfo, 1223
wds_DomainNameList, 1299
wds_PCSCFFQDNAddressList, 1302
wds_PCSCFIPv4ServerAddressList, 1303
numLen
calledPartyInfo, 145
callFWExtInfo, 150
callFWInfo, 151
callingPartyInfo, 154
peerNumberInfo, 706
redirNumInfo, 776
remotePartyNum, 779
voice_callFWExtInfo, 1209
voice_callFWInfo, 1210
voice_peerNumberInfo, 1225
voice_remotePartyNum, 1228
numOfFiles
uim_refreshevent, 951
UIMRefreshEvent, 971
numOfRoutes
sms_setRoutesReq, 867
smsSetRoutesReq, 881
numOpt
DHCPOptionList, 221
wds_DHCPLeaseOptTlv, 1295
wds_DHCPv4OptionList, 1298
WdsDHCPv4OptionList, 1315
wdsDhcpv4OptionList, 1315
numPI
peerNumberInfo, 706
voice_peerNumberInfo, 1225
NumPilots
nas_PilotSetData, 480
PilotSetData, 712
numPlan
calledPartyInfo, 145
callFWExtInfo, 150
callingPartyInfo, 154
connectNumInfo, 187
peerNumberInfo, 706
redirNumInfo, 776
voice_callFWExtInfo, 1209
voice_connectNumInfo, 1217
voice_peerNumberInfo, 1225
numPresInd
connectNumInfo, 187
voice_connectNumInfo, 1217
numQosFlow
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
numRadioInterfaces
nas_servSystem, 498
servSystem, 805
NumRxFilters
unpack_qos_QosFlowInfo_t, 1094
numSI
peerNumberInfo, 706
voice_peerNumberInfo, 1225
numSlot
cardStatus, 156
uim_cardStatus, 940
NumSupUSBComps
unpack_dms_GetUSBComp_t, 1003
NumTxFilters
unpack_qos_QosFlowInfo_t, 1094
numType
calledPartyInfo, 145
callFWExtInfo, 150
callingPartyInfo, 154
connectNumInfo, 187
peerNumberInfo, 706
redirNumInfo, 776
voice_callFWExtInfo, 1209
voice_connectNumInfo, 1218
voice_peerNumberInfo, 1225
number
calledPartyInfo, 145
callFWExtInfo, 150
callFWInfo, 151
callingPartyInfo, 154
ECTNum, 234
peerNumberInfo, 706
redirNumInfo, 776
voice_callFWExtInfo, 1209
voice_callFWInfo, 1210
voice_peerNumberInfo, 1225
numberPlan
callFwdTypeAndPlan, 148
voice_callFwdTypeAndPlan, 1207
numberType
callFwdTypeAndPlan, 148
voice_callFwdTypeAndPlan, 1207
NxtHdrProto
unpack_qos_swiQosFilter_t, 1104

OKtoRefresh
pack_uim_SLQSUIMRefreshOK_t, 656
UIMRefreshOKReq, 973
OMADMCancelSession
qaGobiApiOmadm.h, 1666
OMADMEnabled
unpack_swioma_SLQSOMADMGetSettings_t, 1127
OMADMGetPendingNIA

qaGobiApiOmadm.h, 1667
OMADMGetSessionInfo
qaGobiApiOmadm.h, 1667
OMADMStartSession
qaGobiApiOmadm.h, 1669
OTASPStatus
voiceOTASPStatusInfo, 1265
oddEvenInd
calledPartySubAdd, 146
voice_calledPartySubAdd, 1206
OfflineReason
unpack_dms_GetPower_t, 1002
offset
readTransparentInfo, 774
uim_readTransparentInfo, 949
oldPINLen
changeUIMPIN, 177
uim_changeUIMPIN, 941
oldPINVal
changeUIMPIN, 177
uim_changeUIMPIN, 941
oldPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 673
voiceSetCallBarringPwdInfo, 1269
oldValue
pack_dms_UIMChangePIN_t, 585
omaDmConfig
sessionInfo, 806
sessionInfoExt, 806
omaDmConfigTlv, 569
alertmsg, 569
alertmsglength, 569
state, 569
userInputReq, 569
userInputTimeout, 569
omaDmConfigTlvExt, 570
alertmsg, 571
alertmsglength, 571
state, 571
userInputReq, 571
userInputTimeout, 571
omaDmFota
sessionInfo, 806
sessionInfoExt, 806
omaDmFotaTlv, 571
description, 573
descriptionlength, 573
fwdloadsize, 573
fwloadComplete, 573
namelength, 573
package_name, 573
sessionType, 573
severity, 573
state, 573
updateCompleteStatus, 573
userInputReq, 573
userInputTimeout, 573
version, 573
versionlength, 573
omaDmFotaTlvExt, 573
description, 575
descriptionlength, 575
fumoResultCode, 575
namelength, 575
package_name, 575
packageSize, 575
receivedBytes, 575
reserved, 575
state, 575
userInputTimeout, 575
version, 575
versionlength, 575
omaDmNotifications
sessionInfo, 806
omaDmNotificationsTlv, 575
notification, 575
sessionStatus, 575
opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 602
Open Mobile Alliance Service (OMA), 47
operatingMode
dms_OperatingModeTlv, 223
OperatingModeTlv
unpack_dms_SetEventReport_ind_t, 1006
operation
depersonalizationInformation, 217
pack_pds_SetPDSDefaults_t, 630
uim_depersonalizationInformation, 942
OperationMode
unpack_dms_GetPower_t, 1002
operatorNameString, 575
PLMNName, 576
OperatorPLMNData, 576
lac1, 576
lac2, 576
mcc, 577
mnc, 577
PLMNRecID, 577
operatorPLMNList, 577
numInstance, 577
PLMNData, 577
optCode
DHCPOption, 220
wds_DHCPOpt, 1296
wds_DHCPv4Option, 1297
WdsDHCPv4Option, 1314
wdsDhcpv4Option, 1313
optList
wds_DHCPLeaseOptTlv, 1295
optListData
wds_DHCPLeaseOptTlv, 1295
optVal
wds_DHCPv4Option, 1297
WdsDHCPv4Option, 1314
wdsDhcpv4Option, 1313

optValLen
DHCPOption, 220
wds_DHCPOpt, 1296
wds_DHCPv4Option, 1297
WdsDHCPv4Option, 1314
wdsDhcpv4Option, 1313
OriginateUSSD
qaGobiApiVoice.h, 1802
otaMsgTlv
NASQmiCbkNasSwiOTAMessageInd, 549

p1Status
unpack_dms_UIMGetPINStatus_t, 1023
p1UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1023
p1VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1023
p2Status
unpack_dms_UIMGetPINStatus_t, 1023
p2UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1023
p2VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1023
p3GPP2Pri
swiQosFlow, 906
p3GPP2TimeInfo
GetNetworkTimeResp, 277
unpack_nas_SLQSGetNetworkTime_t, 1058
p3GPPImCn
swiQosFlow, 906
p3GPPResResidualBER
swiQosFlow, 906
p3GPPSigInd
swiQosFlow, 906
p3GPPTimeInfo
GetNetworkTimeResp, 277
unpack_nas_SLQSGetNetworkTime_t, 1058
p3GPPTraHdlPri
swiQosFlow, 906
p3GppNetworkInfoInstances
unpack_nas_PerformNetworkScan_t, 1052
p3GppNetworkInstanceSize
unpack_nas_PerformNetworkScan_t, 1052
p3gppRelease
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 691
pAAASPI
pack_wds_SetMobileIPProfile_t, 685
PACK_WDS_IPV4
wds.h, 1966
PACK_WDS_IPV6
wds.h, 1966
pAMRStatus
pack_voice_SLQSVoiceGetConfig_t, 669
voiceGetConfigReq, 1257
pAPN
_SLQSAVMSSettingsAPNInfo, 86
pAPNClass
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pAPNClass3GPP2
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pAPNDisabledFlag
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pAPNEnabled3GPP2
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pAPNInfo
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAPNName
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
qmiWdsRunTimeSettings, 767
swiPDPRuntimeSettingsResp, 901
pAPNnameSize
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
PASSWORD_LENGTH
qaGobiApiVoice.h, 1801
pAVRXAVCHeadroom
RXAVCList, 787
pAVRXAVCSens
RXAVCList, 787
pAccelAcceptReady
QmiCbkLocSensorStreamingInd, 756
pAccelSamplesAccepted
QmiCbkLocInjectSensorDataInd, 749
pAccelTempAcceptReady
QmiCbkLocSensorStreamingInd, 756
pAccelTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 749
pAcceleroData
LocInjectSensorDataReq, 379
pAcceleroTempData
LocInjectSensorDataReq, 379
pAcceleroTimeSrc
LocInjectSensorDataReq, 379
pAccuracy
unpack_pds_GetPDSDefaults_t, 1088
pAcqOrder
acqOrderPref, 115
nas_acqOrderPref, 409
pAcqOrderPref
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 628
pActPilotPNElements
nas_NetworkStat1x, 470
NetworkStat1x, 563

pAddCDMASysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pAddGSMSysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pAddHDRSysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pAddLTESysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pAddWCDMASysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pAddrAllocPref
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pAddress
pack_wds_SetMobileIPProfile_t, 685
pAirTimer
pack_voice_SLQSVoiceGetConfig_t, 669
voiceGetConfigReq, 1257
pAirTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1156
voiceGetConfigResp, 1259
pAirTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 675
voiceSetConfigReq, 1271
pAirTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1160
voiceSetConfigResp, 1273
pAlertPriority
cdmaMsgDecodingParams, 166
pAlertType
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pAlertingPattern
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pAllowLinger
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pAlphaID
SMSAsyncRawSend_s, 869
pAlphaIDInfo
unpack_voice_SLQSOriginateUSSD_t, 1137
unpack_voice_SLQSVoiceDialCall_t, 1138
unpack_voice_SLQSVoiceGetCallBarring_t, 1143
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1144
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
unpack_voice_SLQSVoiceGetCallWaiting_t, 1148
unpack_voice_SLQSVoiceGetCLIP_t, 1150
unpack_voice_SLQSVoiceGetCLIR_t, 1151
unpack_voice_SLQSVoiceGetCNAP_t, 1152
unpack_voice_SLQSVoiceGetCOLP_t, 1153
unpack_voice_SLQSVoiceGetCOLR_t, 1155
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1158
unpack_voice_SLQSVoiceSetSUPSService_t, 1161
USSResp, 1192
voiceCallInfoResp, 1236
voiceCallResponseParams, 1239
voiceGetCallBarringResp, 1245
voiceGetCallFWResp, 1248
voiceGetCallWaitInfo, 1249
voiceGetCLIPResp, 1250
voiceGetCLIRResp, 1251
voiceGetCNAPResp, 1253
voiceGetCOLPResp, 1254
voiceGetCOLRResp, 1255
voiceSetCallBarringPwdResp, 1270
voiceSetSUPSServiceResp, 1277
voiceSUPSInfo, 1280
pAlphaIdentifier
USSDNoWaitIndicationInfo, 1191
pAltitudeAssumed
QmiCbkLocPositionReportInd, 755
unpack_loc_PositionRpt_Ind_t, 1040
pAltitudeSrcInfo
LocInjectPositionReq, 377
pAltitudeWrtEllipsoid
LocInjectPositionReq, 377
PDSPositionData, 703
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1031
unpack_loc_PositionRpt_Ind_t, 1040
pAltitudeWrtMeanSeaLevel
LocInjectPositionReq, 377
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1031
unpack_loc_PositionRpt_Ind_t, 1040
pAltitudeWrtSealevel
PDSPositionData, 703
pAmrMode
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pAmrOctetAligned
GetIMSVoIPConfigResp, 271

imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pAmrWBMode
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pAmrWBOctetAligned
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pAmrWbEnable
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pApnString
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pApnStringSize
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pApnname
pack_wds_SetDefaultProfile_t, 682
pAppName
loc_LocApplicationInfo, 364
LocApplicationInfo, 371
pAppPriority
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pAppProvider
loc_LocApplicationInfo, 364
LocApplicationInfo, 371
pAppSubType
HDRProtSubtypResp, 298
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1079
pAppType
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pAppVersion
loc_LocApplicationInfo, 364
LocApplicationInfo, 371
pApplication
pack_nas_SetCDMANetworkParameters_t, 607
pApplicationInfo
LOCStartReq, 381
pack_loc_Start_t, 604
pArrAlertingPattern
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrAlertingType
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrAlphaID
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrCallEndReason
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrCallInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
pArrCalledPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrConnectPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrDiagInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrRedirPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrRemotePartyName
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrRemotePartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrSvcOption
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1141
voiceGetAllCallInfo, 1243
voiceSetAllCallStatusCbkInfo, 1268
pArrUUSInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1142
voiceGetAllCallInfo, 1244
pAuth
pack_wds_SLQSStartDataSession_t, 695
pAuthPassword
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pAuthPassword_tSize
LibPackprofile_3GPP2, 350
pAuthPasswordSize
LibpackProfile3GPP2, 339
Profile3GPP2, 730
pAuthProtocol
LibpackProfile3GPP2, 339
LibPackprofile_3GPP2, 350
Profile3GPP2, 730
pAuthRetryCount

LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pAuthTimeout
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pAuthenticateResult
UIMAuthenticateResp, 960
unpack_uim_SLQSUIMAuthenticate_t, 1131
pAuthentication
qmiWdsRunTimeSettings, 767
ssdatasession_params, 888
pAuthenticationPref
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pAutoAnsStatus
unpack_voice_SLQSVoiceSetConfig_t, 1160
voiceSetConfigResp, 1273
pAutoAnswer
pack_voice_SLQSVoiceGetConfig_t, 669
pack_voice_SLQSVoiceSetConfig_t, 675
voiceGetConfigReq, 1257
voiceSetConfigReq, 1272
pAutoAnswerStat
unpack_voice_SLQSVoiceGetConfig_t, 1156
voiceGetConfigResp, 1259
pAutoConnect
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAutoReboot
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAutoSelection
UIMGetConfigurationResp, 964
unpack_uim_SLQSUIMGetConfiguration_t, 1133
pAutosdm
_SLQSOMADMSettings, 93
_SLQSOMADMSettingsReqParams, 94
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 647
pBandPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 628
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pBdsSVInfo
LocDelAssDataReq, 372
pack_loc_Delete_Assist_Data_t, 591
pBearerID
QosFlowInfo, 771
pBearerId
swiPDPRuntimeSettingsResp, 901
pBearerTech
DataBearerTechExt, 210
pBroadcast
pack_nas_SetCDMANetworkParameters_t, 607
pBroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 642
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1117
pBurstDTMFLengths
pack_voice_SLQSVoiceBurstDTMF_t, 663
voiceBurstDTMFInfo, 1233
pCCResType
unpack_voice_SLQSVoiceGetCallBarring_t, 1143
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1144
unpack_voice_SLQSVoiceGetCallWaiting_t, 1148
unpack_voice_SLQSVoiceGetCLIP_t, 1150
unpack_voice_SLQSVoiceGetCLIR_t, 1151
unpack_voice_SLQSVoiceGetCNAP_t, 1152
unpack_voice_SLQSVoiceGetCOLP_t, 1153
unpack_voice_SLQSVoiceGetCOLR_t, 1155
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1158
voiceGetCallBarringResp, 1246
voiceGetCallFWResp, 1248
voiceGetCallWaitInfo, 1249
voiceGetCLIPResp, 1250
voiceGetCLIRResp, 1252
voiceGetCNAPResp, 1253
voiceGetCOLPResp, 1254
voiceGetCOLRResp, 1255
voiceSetCallBarringPwdResp, 1270
pCCResultType
unpack_voice_SLQSVoiceDialCall_t, 1139
unpack_voice_SLQSVoiceSetSUPSService_t, 1161
voiceCallResponseParams, 1239
voiceSetSUPSServiceResp, 1277
pCCSUPSType
unpack_voice_SLQSVoiceDialCall_t, 1139
unpack_voice_SLQSVoiceGetCallBarring_t, 1143
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1144
unpack_voice_SLQSVoiceGetCallWaiting_t, 1148
unpack_voice_SLQSVoiceGetCLIP_t, 1150
unpack_voice_SLQSVoiceGetCLIR_t, 1151
unpack_voice_SLQSVoiceGetCNAP_t, 1152
unpack_voice_SLQSVoiceGetCOLP_t, 1153
unpack_voice_SLQSVoiceGetCOLR_t, 1155
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1158
unpack_voice_SLQSVoiceSetSUPSService_t, 1161
voiceCallResponseParams, 1239
voiceGetCallBarringResp, 1246
voiceGetCallFWResp, 1248
voiceGetCallWaitInfo, 1249
voiceGetCLIPResp, 1250
voiceGetCLIRResp, 1252
voiceGetCNAPResp, 1253
voiceGetCOLPResp, 1254
voiceGetCOLRResp, 1255

voiceSetCallBarringPwdResp, 1270
voiceSetSUPSServiceResp, 1277
pCCSuppsType
unpack_voice_SLQSOriginateUSSD_t, 1137
USSResp, 1192
pCDMABroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 642
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1117
pCDMAChannel
nasGet3GPP2SubscriptionInfoResp, 524
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1070
pCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pCDMAECIOThreshList
CDMAECIOThresh, 163
nas_CDMAECIOThresh, 413
pCDMAFrameErrRate
GetErrRateResp, 266
unpack_nas_SLQSGetErrorRate_t, 1057
pCDMAInfo
nasCellLocationInfoResp, 522
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pCDMARSSIThreshList
CDMARSSIThresh, 169
nas_CDMARSSIThresh, 415
pCDMASSInfo
nasGetSigInfoResp, 527
pCDMASigInfo
nasSigInfo, 552
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1075
pCDMASrvStatusInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pCDMASysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pCLIPResp
unpack_voice_SLQSVoiceGetCLIP_t, 1150
voiceGetCLIPResp, 1250
pCLIPstatus
voiceSUPSInfo, 1280
pCLIRCause
voiceInfoRec, 1262
pCLIRResp
unpack_voice_SLQSVoiceGetCLIR_t, 1151
voiceGetCLIRResp, 1252
pCLIRType
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238
pCLIRstatus
voiceSUPSInfo, 1280
PCMparams, 699
iFaceTab, 700
iFaceTabLen, 700
pCNAPResp
unpack_voice_SLQSVoiceGetCNAP_t, 1152
voiceGetCNAPResp, 1253
pCNAPstatus
voiceSUPSInfo, 1280
pCOLPResp
unpack_voice_SLQSVoiceGetCOLP_t, 1154
voiceGetCOLPResp, 1254
pCOLPstatus
voiceSUPSInfo, 1280
pCOLRResp
unpack_voice_SLQSVoiceGetCOLR_t, 1155
voiceGetCOLRResp, 1255
pCOLRstatus
voiceSUPSInfo, 1280
PCSCFAddrPCO
unpack_wds_SLQSGetRuntimeSettings_t, 1179
PCSCFFQDNAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1179
PCSCFFQDNAddress, 700
fqdnAddr, 700
fqdnLen, 700
PCSCFFQDNAddressList, 700
numInstances, 701
pcsfFQDNAddress, 701
PCSCFIPv4ServerAddressList, 701
numInstances, 701
pscsfIPv4Addr, 701
pCSCFPortName
imsRegMgrConfigInfo, 318
SetRegMgrConfigReq, 824
pCSCFPortNameLen
SetRegMgrConfigReq, 824
pCSGID
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 628
pCUGIndex
voiceSUPSNotification, 1282
pCUGInfo
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238
pCallBarPasswd
voiceSUPSInfo, 1280
pCallBarringPasswd

pack_voice_SLQSVoiceSetSUPSService_t, 677
voiceSetSUPSServiceReq, 1276
pCallEndReason
getDUNCallInfoResp, 264
pCallFWNum
voiceSUPSInfo, 1280
pCallFWTimerVal
voiceSUPSInfo, 1280
pCallForwardingNumber
pack_voice_SLQSVoiceSetSUPSService_t, 677
voiceSetSUPSServiceReq, 1276
pCallFwdInfo
voiceSUPSInfo, 1280
pCallFwdTypeAndPlan
pack_voice_SLQSVoiceSetSUPSService_t, 677
voiceSetSUPSServiceReq, 1276
pCallID
burstDTMFInfo, 142
pack_voice_SLQSVoiceManageCalls_t, 671
pack_voice_SLQSVoiceSendFlash_t, 672
pack_voice_SLQSVoiceStartContDTMF_t, 678
unpack_voice_SLQSVoiceBurstDTMF_t, 1138
unpack_voice_SLQSVoiceDialCall_t, 1138
unpack_voice_SLQSVoiceGetCallBarring_t, 1143
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1144
unpack_voice_SLQSVoiceGetCallWaiting_t, 1148
unpack_voice_SLQSVoiceGetCLIP_t, 1150
unpack_voice_SLQSVoiceGetCLIR_t, 1151
unpack_voice_SLQSVoiceGetCNAP_t, 1152
unpack_voice_SLQSVoiceGetCOLP_t, 1153
unpack_voice_SLQSVoiceGetCOLR_t, 1155
unpack_voice_SLQSVoiceSendFlash_t, 1157
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1158
unpack_voice_SLQSVoiceSetSUPSService_t, 1161
unpack_voice_SLQSVoiceStartContDTMF_t, 1162
voice_burstDTMFInfo, 1205
voiceCallResponseParams, 1239
voiceContDTMFinfo, 1239
voiceFlashInfo, 1241
voiceGetCallBarringResp, 1246
voiceGetCallFWResp, 1248
voiceGetCallWaitInfo, 1249
voiceGetCLIPResp, 1250
voiceGetCLIRResp, 1251
voiceGetCNAPResp, 1253
voiceGetCOLPResp, 1254
voiceGetCOLRResp, 1255
voiceManageCallsReq, 1263
voiceSetCallBarringPwdResp, 1270
voiceSetSUPSServiceResp, 1277
voiceSUPSInfo, 1280
pCallId
pack_voice_SLQSVoiceAnswerCall_t, 662
pack_voice_SLQSVoiceEndCall_t, 665
unpack_voice_SLQSOriginateUSSD_t, 1137
unpack_voice_SLQSVoiceAnswerCall_t, 1137
unpack_voice_SLQSVoiceEndCall_t, 1139
USSResp, 1192
voiceAnswerCall, 1232
pCallInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pCallPartySubAdd
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238
pCallType
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238
pCallWaitInd
voiceInfoRec, 1262
pCallbackAddr
cdmaMsgEncodingParams, 168
pCallbkAddr
cdmaMsgDecodingParams, 166
pCallbkAddrLength
cdmaMsgDecodingParams, 166
pCalledPartyInfo
voiceInfoRec, 1262
pCallerIDInfo
voiceInfoRec, 1262
pCallerNameInfo
voiceInfoRec, 1262
pCallingPartyInfo
voiceInfoRec, 1262
pCardResult
UIMAuthenticateResp, 960
UIMGetFileAttributesResp, 965
UIMReadTransparentResp, 969
unpack_uim_ReadTransparent_t, 1129
unpack_uim_SLQSUIMAuthenticate_t, 1131
unpack_uim_SLQSUIMGetFileAttributes_t, 1134
pCardStatus
UIMGetCardStatusResp, 962
unpack_uim_GetCardStatus_t, 1128
unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind_t, 1135
pCcResultType
unpack_voice_SLQSOriginateUSSD_t, 1137
USSResp, 1192
pCellDb
LocDelAssDataReq, 372
pack_loc_Delete_Assist_Data_t, 591
pChangeDuration
nasInitNetworkReg, 535
pack_nas_SLQSInitiateNetworkRegistration_t, 611
pChannelRate
getDUNCallInfoResp, 264
pChgDuration
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 628
pClkInfo
LocDelAssDataReq, 372
pack_loc_Delete_Assist_Data_t, 591

pCodecSTGain
GetAudioPathConfigResp, 254
SetAudioPathConfigReq, 808
pColorCode
nasGetHDRColorCodeResp, 525
unpack_nas_SLQSNasGetHDRColorCode_t, 1072
pConfidence
LocSetCradleMountReq, 380
pConfigAltitudeAssumed
LOCStartReq, 381
pack_loc_Start_t, 605
pConfigurationMask
pack_uim_SLQSUIMGetConfiguration_t, 653
UIMGetConfigurationReq, 963
pConnectNumInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
voiceInfoRec, 1262
pConnectionRetryTimers
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pConnectionStatus
getDUNCallInfoResp, 264
pContextId
swiPDPRuntimeSettingsResp, 901
pCrashInfo
CrashInfoParams, 189
pCrashString
CrashInfo, 188
pCreateProfileOut
unpack_wds_SLQSCreateProfile_t, 1172
pCsgId
nasNetworkReject, 539
pCurAMRConfig
unpack_voice_SLQSVoiceGetConfig_t, 1156
voiceGetConfigResp, 1259
pCurDataBearerTechnology
dataBearers, 207
pCurPrefVoiceSO
unpack_voice_SLQSVoiceGetConfig_t, 1156
voiceGetConfigResp, 1259
pCurProfile
pack_wds_SLQSCreateProfile_t, 686
pCurVoiceDomainPref
unpack_voice_SLQSVoiceGetConfig_t, 1156
voiceGetConfigResp, 1259
pCurVoicePrivacyPref
unpack_voice_SLQSVoiceGetConfig_t, 1157
voiceGetConfigResp, 1259
pCurrChannelRateInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pCurrDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pCurrImgInfo
CurrentImgList, 197
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1015
pCurrNetworkInfo
CurrDataSysStat, 196
pCurrPrefDataSysInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pCurrTTYMode
unpack_voice_SLQSVoiceGetConfig_t, 1156
voiceGetConfigResp, 1259
pCurrentChannelRXRate
WdsConnectionRateElmnts, 1310
pCurrentChannelTXRate
WdsConnectionRateElmnts, 1310
pCurrentPersonality
HDRPersonalityInd, 297
HDRPersonalityResp, 297
unpack_nas_SLQSSwiGetHDRPersonality_t, 1078
pCurrentPrsnlty
HDRProtSubtypResp, 298
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1079
pCurrentmitigationLvl
TmdGetMitigationLvlResp, 925
pCustSettingInfo
DMSgetCustomFeatureV2, 226
getCustomFeatureV2, 259
pCustSettingList
DMSgetCustomFeatureV2, 226
getCustomFeatureV2, 259
pCustomSCP
pack_nas_SetCDMANetworkParameters_t, 607
pCwtMute
SetM2MAudioProfileReq, 820
SetM2MAVMuteReq, 821
pDDTMInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pDHCPRelayEnabled
custFeaturesInfo, 203
custFeaturesSetting, 205
PDOP
loc_precisionDilution, 365
precisionDilution_s, 718
PDPType
unpack_wds_SLQSGetRuntimeSettings_t, 1179
pDRCParams
GetHRPDStatsResp, 267
unpack_nas_SLQSSwiGetHRPDStats_t, 1079
PDS_SRV
qaGobiApiCbk.h, 1462
PDSInjectTimeReference
qaGobiApiPds.h, 1675
PDSPosMethodStateReq, 703
pWifiState, 704
pXtraDataState, 704
pXtraTimeState, 704
PDSPositionData, 701
pAltitudeWrtEllipsoid, 703
pAltitudeWrtSealevel, 703

pHorizontalConfidence, 703
pHorizontalUncCircular, 703
pLatitude, 703
pLongitude, 703
pPositionSource, 703
pTimeStamp, 703
pTimeType, 703
pVerticalConfidence, 703
pVerticalUnc, 703
pDTMFTXGain
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 808
pDataBearer
QosEventInfo, 770
slqsWdsEventInfo, 854
unpack_wds_GetDataBearerTechnology_t, 1165
pDataBearerTech
getDUNCallInfoResp, 264
pDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pDataCallStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pDataMode
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pDataRate
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
swiQosFlow, 906
pDataSrc
voiceSUPSInfo, 1280
pDataStatusDetail
NetworkDebugResp, 561
unpack_nas_SLQSSwiNetworkDebug_t, 1081
pDataSystemStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pDate
_SLQSOMADMSessionInfo, 91
pDateLength
_SLQSOMADMSessionInfo, 91
pDayltSavAdj
nasNetworkTime, 540
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1074
pDefaultPDNEnabled
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 691
pDeferTime
pack_swioma_SLQSOMADMSendSelection_t, 646
pDesription
QmiNas3GppNetworkInfo, 762
pDestAddr
cdmaMsgEncodingParams, 168
wcdmaMsgEncodingParams, 1288
pDestSMSContent
getDyingGaspCfg, 265
pack_dms_SLQSSwiSetDyingGaspCfg_t, 583
packgetDyingGaspCfg, 699
setDyingGaspCfg, 813
pDestSMSNum
getDyingGaspCfg, 265
pack_dms_SLQSSwiSetDyingGaspCfg_t, 583
packgetDyingGaspCfg, 699
setDyingGaspCfg, 813
pDetachAction
pack_nas_SLQSSwiPSDetach_t, 629
PSDetachReq, 734
pDevCrashStatus
CrashInfoParams, 189
pDeviceConfigDetail
NetworkDebugResp, 561
unpack_nas_SLQSSwiNetworkDebug_t, 1081
pDiagInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pDigitBuff
burstDTMFInfo, 142
voice_burstDTMFInfo, 1205
pDirNum
nasGet3GPP2SubscriptionInfoResp, 524
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1070
pDisableIMSI
custFeaturesInfo, 203
pDispInfo
voiceInfoRec, 1262
pDisplayMode
cdmaMsgDecodingParams, 166
pDomainList
qmiWdsRunTimeSettings, 767
pDormancyStatus
getDUNCallInfoResp, 264
slqsWdsEventInfo, 854
pDormancyStatusInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pDualStandByPrefInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pECIOThresList
ECIOThresh, 233
nas_ECIOThresh, 431
pECIOThresh
pack_nas_SLQSConfigSigInfo_t, 610
sigInfo, 833
pECMode
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pECTNum
voiceSUPSNotification, 1282

PER
nas_NetworkStatEVDO, 472
NetworkStatEVDO, 565
pESNString
serialNumbersInfo, 801
pEVDOPageMonPerChangeInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pEarMute
SetM2MAudioProfileReq, 820
pEmerMode
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 628
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pEmergencyCategory
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238
pEnableNotification
pack_wds_DHCPv4ClientLeaseChange_t, 679
WdsClientLeaseChange, 1309
pEnabled
pack_wds_SetMobileIPProfile_t, 685
pEnabledStatus
unpack_pds_GetPDSState_t, 1089
pEncodingAlphabet
cdmaMsgEncodingParams, 168
pEncryptData
pack_uim_ReadTransparent_t, 650
UIMReadTransparentReq, 968
pEncryptedData
UIMReadTransparentResp, 969
unpack_uim_ReadTransparent_t, 1129
pEncryptedPIN1
pack_uim_VerifyPin_t, 659
UIMPinResp, 966
UIMVerifyPinReq, 980
unpack_uim_ChangePin_t, 1128
unpack_uim_SetPinProtection_t, 1130
unpack_uim_UnblockPin_t, 1136
unpack_uim_VerifyPin_t, 1136
pError
USSDNoWaitIndicationInfo, 1191
pErrorCodeStr
imsaRatStatusInfo, 311
pErrorMask
CATEventDataType, 159
pErrorRateInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pEspSpi
swiQosFilter, 903
pEtwsMessageInfo
SMSEventInfo_s, 874
pEtwsPlmnInfo
SMSEventInfo_s, 874
pExtDispInfo
voiceInfoRec, 1262
pExtDispRecInfo
voiceInfoRec, 1262
pExtErrCode
_GetProfileSettingOut, 65
UnPackGetProfileSettingOut, 1187
pExtErrorCode
CreateProfileOut, 192
ModifyProfileOut, 407
unpack_wds_SLQSModifyProfile_t, 1180
pFOTAUpdate
_SLQSOMADMSettings, 93
pFOTAdownload
_SLQSOMADMSettings, 93
pFailCause
unpack_voice_SLQSVoiceGetCallBarring_t, 1143
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1144
unpack_voice_SLQSVoiceGetCallWaiting_t, 1148
unpack_voice_SLQSVoiceGetCLIP_t, 1150
unpack_voice_SLQSVoiceGetCLIR_t, 1151
unpack_voice_SLQSVoiceGetCNAP_t, 1152
unpack_voice_SLQSVoiceGetCOLP_t, 1154
unpack_voice_SLQSVoiceGetCOLR_t, 1155
unpack_voice_SLQSVoiceManageCalls_t, 1157
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1159
unpack_voice_SLQSVoiceSetSUPSService_t, 1161
voiceGetCallBarringResp, 1246
voiceGetCallFWResp, 1248
voiceGetCallWaitInfo, 1249
voiceGetCLIPResp, 1250
voiceGetCLIRResp, 1252
voiceGetCNAPResp, 1253
voiceGetCOLPResp, 1254
voiceGetCOLRResp, 1255
voiceManageCallsResp, 1264
voiceSetCallBarringPwdResp, 1270
voiceSetSUPSServiceResp, 1277
voiceSUPSInfo, 1280
pFailErrorCode
imsaPdpStatusInfo, 310
pFailureCause
USSDNoWaitIndicationInfo, 1191
pFailureReason
unpack_wds_SLQSStartDataSession_t, 1185
pFile
ERIFileparams, 236
pFileAttributes
UIMGetFileAttributesResp, 965
unpack_uim_SLQSUIMGetFileAttributes_t, 1134
pFileSize
ERIFileparams, 236
pFixId
QmiCbkLocPositionReportInd, 755
unpack_loc_PositionRpt_Ind_t, 1040
pFlag
RXPCMIIRFltr, 790

TXPCMIIRFltr, 933
pFlashPayLd
pack_voice_SLQSVoiceSendFlash_t, 672
voiceFlashInfo, 1241
pFlashType
pack_voice_SLQSVoiceSendFlash_t, 672
voiceFlashInfo, 1241
pFollowOnDC
slqssendasyncsmsparams_s, 845
sms_sendAsyncsmsParams, 865
pForbidden
QmiNas3GppNetworkInfo, 762
pForceOnDC
slqssendasyncsmsparams_s, 845
sms_sendAsyncsmsParams, 865
pForceRev0
pack_nas_SetCDMANetworkParameters_t, 607
pFwAutoCheck
_SLQSOMADMSettings, 93
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 647
pFwAutoSDM
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pFwAutoUpdate
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pFwAutodownload
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pGCDumpString
CrashInfo, 188
pGERANInfo
nasCellLocationInfoResp, 522
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pGPRSGrantedQoS
qmiWdsRunTimeSettings, 767
pGPRSMinimumQoS
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pGPRSRequestedQos
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pGPSEnable
custFeaturesSetting, 205
pGPSLPM
custFeaturesInfo, 203
custFeaturesSetting, 205
pGPSSel
custFeaturesInfo, 203
custFeaturesSetting, 205
pGSMBER
GetErrRateResp, 266
unpack_nas_SLQSGetErrorRate_t, 1057
pGSMCallBarringSysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pGSMCipherDomainSysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pGSMRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pGSMRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pGSMRSSIThreshList
GSMRSSIThresh, 290
nas_GSMRSSIThresh, 436
pGSMSSInfo
nasGetSigInfoResp, 527
pGSMSigInfo
nasSigInfo, 552
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1075
pGSMSrvStatusInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pGSMSysInfo
nasGetSysInfoResp, 529
nasSysInfo, 557
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pGWAcqOrderPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 628
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pGWAddressV4
qmiWdsRunTimeSettings, 767
pGenerator
GetM2MAudioProfileReq, 272
SetM2MAudioProfileReq, 820
pGetCallFWExtInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1144
voiceGetCallFWResp, 1248
pGetCallFWInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1144
voiceGetCallFWResp, 1248
pGetCustomInput
DMSgetCustomFeatureV2, 226
getCustomFeatureV2, 259
pGetDyingGaspCfg
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1014
pGetDyingGaspStatistics

unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1014
pGetIndicationRegInfo
unpack_sms_SLQSGetIndicationRegister_t, 1115
pGetMsgWaitingInfoResp
unpack_sms_SLQSGetMessageWaiting_t, 1115
pGetTransLayerInfo
unpack_sms_SLQSGetTransLayerInfo_t, 1118
pGetTransNWRegInfo
unpack_sms_SLQSGetTransNWRegInfo_t, 1118
pGnssData
LocDelAssDataReq, 372
pack_loc_Delete_Assist_Data_t, 591
pGpsTime
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1031
unpack_loc_PositionRpt_Ind_t, 1040
pGyroAcceptReady
QmiCbkLocSensorStreamingInd, 756
pGyroData
LocInjectSensorDataReq, 379
pGyroSamplesAccepted
QmiCbkLocInjectSensorDataInd, 749
pGyroTempAcceptReady
QmiCbkLocSensorStreamingInd, 756
pGyroTempData
LocInjectSensorDataReq, 379
pGyroTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 749
pGyroTimeSrc
LocInjectSensorDataReq, 379
pHA2002bis
pack_wds_SetMobileIPParameters_t, 684
pHAAuthenticator
pack_wds_SetMobileIPParameters_t, 684
pHASPI
pack_wds_SetMobileIPProfile_t, 685
pHDRECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pHDRECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pHDRECIOThreshList
HDRECIOThresh, 296
nas_HDRECIOThresh, 440
pHDRIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pHDRIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pHDRIOThreshList
HDRIOThresh, 296
nas_HDRIOThresh, 441
pHDRNewUATIAssInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pHDRPackErrRate
GetErrRateResp, 266
unpack_nas_SLQSGetErrorRate_t, 1057
pHDRRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pHDRRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pHDRRSSIThreshList
HDRRSSIThresh, 299
nas_HDRRSSIThresh, 441
pHDRSINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pHDRSINRThresList
HDRSINRThresh, 299
nas_HDRSINRThresh, 442
pHDRSINRThresh
pack_nas_SLQSConfigSigInfo_t, 610
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
sigInfo, 833
pHDRSINRThreshList
HDRSINRThreshold, 300
nas_HDRSINRThreshold, 443
pHDRSSInfo
nasGetSigInfoResp, 527
pHDRSessionCloseInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pHDRSigInfo
nasSigInfo, 552
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1075
pHDRSrvStatusInfo
nasGetSysInfoResp, 529
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pHDRSysInfo
nasGetSysInfoResp, 529
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pHaltSubscription
UIMGetConfigurationResp, 964
unpack_uim_SLQSUIMGetConfiguration_t, 1133
pHeading
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1031
unpack_loc_PositionRpt_Ind_t, 1040
pHeadingUnc
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1031

unpack_loc_PositionRpt_Ind_t, 1041
pHomeSIDNID
nasGet3GPP2SubscriptionInfoResp, 524
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1070
pHorCirConf
QmiCbkLocBestAvailPosInd, 744
unpack_loc_BestAvailPos_Ind_t, 1031
pHorConfidence
LocInjectPositionReq, 377
QmiCbkLocPositionReportInd, 755
unpack_loc_PositionRpt_Ind_t, 1041
pHorEllpConf
QmiCbkLocBestAvailPosInd, 744
unpack_loc_BestAvailPos_Ind_t, 1031
pHorReliability
LocInjectPositionReq, 378
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1031
unpack_loc_PositionRpt_Ind_t, 1041
pHorUncCircular
LocInjectPositionReq, 378
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pHorUncEllipseOrientAzimuth
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pHorUncEllipseSemiMajor
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pHorUncEllipseSemiMinor
QmiCbkLocBestAvailPosInd, 744
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pHorizontalAccuracyLvl
LOCStartReq, 381
pack_loc_Start_t, 605
pHorizontalConfidence
PDSPositionData, 703
pHorizontalUncCircular
PDSPositionData, 703
pHotSwapStatus
UIMGetCardStatusResp, 962
unpack_uim_GetCardStatus_t, 1129
pHwConfig
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 694
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1183
WdsDHCPv4Config, 1312
PI
calledPartyInfo, 146
callerIDInfo, 147
callFWExtInfo, 150
callingPartyInfo, 154
redirNumInfo, 776
voice_callFWExtInfo, 1209
PIFACEId
SetM2MAudioAVCFGReq, 818
pIMCNflag
qmiWdsRunTimeSettings, 767
pIMEIString
serialNumbersInfo, 801
pIMSDomain
GetIMSUserConfigParams, 269
imsUserConfigInfo, 320
SetIMSUserConfigReq, 814
pIMSDomainLen
GetIMSUserConfigParams, 269
SetIMSUserConfigReq, 814
pIMSTestMode
GetRegMgrConfigParams, 278
imsRegMgrConfigInfo, 318
SetRegMgrConfigReq, 824
pIOThresList
IOThresh, 325
nas_IOThresh, 447
pIOThresh
pack_nas_SLQSConfigSigInfo_t, 610
sigInfo, 833
pIPAddressV4
QmiWdsIpAddressInfo, 764
qmiWdsRunTimeSettings, 767
pIPAddressV6
QmiWdsIpAddressInfo, 764
pIPFamSupport
custFeaturesInfo, 203
pIPFamily
GetInstIDResp, 271
pIPFamilyPreference
qmiWdsRunTimeSettings, 767
pIPV6AddrInfo
qmiWdsRunTimeSettings, 767
pIPV6GWAddrInfo
qmiWdsRunTimeSettings, 767
pIPv4Addr
WdsDHCPv4ClientLeaseInd, 1311
pIPv4AddrPref
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pIPv4Address
swiPDPRuntimeSettingsResp, 901
pIPv4DstAddr
swiQosFilter, 903
pIPv4GWAddress
swiPDPRuntimeSettingsResp, 901
pIPv4SrcAddr
swiQosFilter, 903

pIPv6AddPref
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pIPv6Address
swiPDPRuntimeSettingsResp, 901
pIPv6DstAddr
swiQosFilter, 903
pIPv6GWAddress
swiPDPRuntimeSettingsResp, 901
pIPv6Label
swiQosFilter, 903
pIPv6SrcAddr
swiQosFilter, 903
pIPv6TrafCls
swiQosFilter, 903
pIPv6prefixlen
QmiWdsIpAddressInfo, 764
pId
swiQosFilter, 903
pIds
swiQosIds, 907
pIgnoreHotSwapSwitch
pack_uim_SLQSUIMPowerUp_t, 655
UIMPowerUpReq, 967
pImCnFlag
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pImageList
pack_fms_SetImagesPreference_t, 590
unpack_fms_GetImagesPreference_t, 1025
pImeiSvnString
serialNumbersInfo, 801
pImgType
FirmwareUpdatStat, 243
pImsRegErrCode
IMSARegistrationStatus, 312
pImsRegStatus
IMSARegistrationStatus, 312
imsaRegStatusInfo, 312
pInUse
QmiNas3GppNetworkInfo, 762
pIndFieldsList
IMSASupportedFieldsResp, 315
pIndicationToken
pack_uim_ChangePin_t, 649
pack_uim_ReadTransparent_t, 650
pack_uim_SetPinProtection_t, 651
pack_uim_SLQSUIMAuthenticate_t, 652
pack_uim_SLQSUIMGetFileAttributes_t, 654
pack_uim_UnblockPin_t, 658
pack_uim_VerifyPin_t, 659
UIMAuthenticateReq, 959
UIMAuthenticateResp, 960
UIMChangePinReq, 960
UIMGetFileAttributesReq, 964
UIMGetFileAttributesResp, 965
UIMPinResp, 966
UIMReadTransparentReq, 968
UIMReadTransparentResp, 969
UIMSetPinProtectionReq, 975
UIMUnblockPinReq, 979
UIMVerifyPinReq, 980
unpack_uim_ChangePin_t, 1128
unpack_uim_ReadTransparent_t, 1129
unpack_uim_SetPinProtection_t, 1130
unpack_uim_SLQSUIMAuthenticate_t, 1131
unpack_uim_SLQSUIMGetFileAttributes_t, 1134
unpack_uim_UnblockPin_t, 1136
unpack_uim_VerifyPin_t, 1136
pInfo
pack_voice_AnswerUSSD_t, 660
pack_voice_OriginateUSSD_t, 660
pInstanceID
GetInstIDResp, 271
pInstanceSize
QmiNasPerformNetworkScanResp, 763
pInstances
QmiNasPerformNetworkScanResp, 763
pInstancesSize
QmiNasGetRFBandInfoResp, 762
pIntermediateReportState
LOCStartReq, 382
pack_loc_Start_t, 605
pInterval
unpack_pds_GetPDSDefaults_t, 1088
pIpcpAckTimeout
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pIpcpCreqRetryCount
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pIsPcscfAddressNedded
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pIsUDHPresent
wcdmaLongMsgDecodingParams, 1285
pIsVoiceEnabled
custFeaturesInfo, 203
custFeaturesSetting, 205
pJitter
swiQosFlow, 906
pKeyReferenceID
pack_uim_ChangePin_t, 649
pack_uim_SetPinProtection_t, 651
pack_uim_UnblockPin_t, 658
pack_uim_VerifyPin_t, 659
UIMChangePinReq, 960
UIMSetPinProtectionReq, 975
UIMUnblockPinReq, 979
UIMVerifyPinReq, 980
PLMN_LENGTH

qaGobiApiNas.h, 1625
PLMNData
nas_operatorPLMNList, 475
operatorPLMNList, 577
PLMNName
nas_operatorNameString, 473
operatorNameString, 576
PLMNNetName
nas_PLMNNetworkName, 481
PLMNNetworkName, 715
PLMNNetworkName, 714
numInstance, 715
PLMNNetName, 715
PLMNNetworkNameData, 715
codingScheme, 716
countryInitials, 716
longName, 716
longNameLen, 716
longNameSpareBits, 716
shortName, 716
shortNameLen, 716
shortNameSpareBits, 717
PLMNRecID
nas_OperatorPLMNData, 474
OperatorPLMNData, 577
pLTEAttachProfile
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 691
pLTEAttachProfileList
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 691
pLTEBandPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 628
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pLTECphyCa
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pLTEInfo
swiModemStatusResp, 897
unpack_nas_SLQSNasSwiModemStatus_t, 1076
pLTEInfoInterfreq
nasCellLocationInfoResp, 522
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pLTEInfoIntrafreq
nasCellLocationInfoResp, 522
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pLTEInfoNeighboringGSM
nasCellLocationInfoResp, 522
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pLTEInfoNeighboringWCDMA
nasCellLocationInfoResp, 523
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pLTERSRPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pLTERSRPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pLTERSRPThreshList
LTERSRPThresh, 393
nas_LTERSRPThresh, 457
pLTERSRQDelta
pack_nas_SLQSNasConfigSigInfo2_t, 615
setSignalStrengthInfo, 828
pLTERSRQThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 828
pLTERSRQThreshList
LTERSRQThresh, 394
nas_LTERSRQThresh, 458
pLTERSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pLTERSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pLTERSSIThreshList
LTERSSIThresh, 394
nas_LTERSSIThresh, 458
pLTESNRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pLTESNRThresList
LTESNRThresh, 397
nas_LTESNRThresh, 461
pLTESNRThresh
pack_nas_SLQSConfigSigInfo_t, 610
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
sigInfo, 833
pLTESNRThreshList
LTESNRThreshold, 398
nas_LTESNRThreshold, 462
pLTESSInfo
nasGetSigInfoResp, 527
pLTESigInfo
nasSigInfo, 552
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1075
pLTESigRptCfg
pack_nas_SLQSConfigSigInfo_t, 610
sigInfo, 833
pLTESigRptConfig
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pLTESrvStatusInfo
nasGetSysInfoResp, 530
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084

pLTESysInfo
nasGetSysInfoResp, 530
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1065
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pLTEVoiceSupportSysInfo
nasGetSysInfoResp, 530
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1066
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pLanguage
cdmaMsgDecodingParams, 166
pLastBearerTech
DataBearerTechExt, 210
pLastCallDataBearerTech
getDUNCallInfoResp, 264
pLastCallDataBearerTechnology
dataBearers, 207
pLastCallRXOKBytesCnt
getDUNCallInfoResp, 264
pLastCallTXOKBytesCnt
getDUNCallInfoResp, 264
pLatency
swiQosFlow, 906
pLatitude
LocInjectPositionReq, 378
PDSPositionData, 703
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pLcpAckTimeout
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pLcpCreqRetryCount
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pLeapSeconds
QmiCbkLocPositionReportInd, 755
unpack_loc_PositionRpt_Ind_t, 1041
pLeaseState
WdsDHCPv4ClientLeaseInd, 1311
pLineCtrlInfo
voiceInfoRec, 1262
pLinktimer
pack_sms_SendSMS_t, 635
slqssendasyncsmsparams_s, 845
slqssendsmsparams_s, 846
sms_sendAsyncsmsParams, 865
pLogString
FirmwareUpdatStat, 243
pLogStringLen
FirmwareUpdatStat, 244
pLongitude
LocInjectPositionReq, 378
PDSPositionData, 703
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pLoopbackMode
WDSSetLoopbackData, 1323
pLoopbackMultiplier
WDSSetLoopbackData, 1323
pLteBandCapability
BandCapabilityResp, 140
pLteEARFCN
nasSwiGetChannelLockResp, 553
nasSwiSetChannelLockReq, 555
pack_nas_SLQSNASSwiSetChannelLock_t, 622
unpack_nas_SLQSNASSwiGetChannelLock_t, 1075
pLteNasRelInfo
SwiOTAMsg_s, 898
pLtePCI
nasSwiGetChannelLockResp, 553
nasSwiSetChannelLockReq, 555
pack_nas_SLQSNASSwiSetChannelLock_t, 622
unpack_nas_SLQSNASSwiGetChannelLock_t, 1075
pLteQci
swiQosFlow, 906
pMCC
QmiNas3GppNetworkInfo, 762
unpack_nas_GetHomeNetwork3GPP2_t, 1047
pMEIDString
serialNumbersInfo, 801
pMICGainSelect
GetAudioPathConfigResp, 255
pMIPStatusInd
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pMNAAA
pack_wds_SetMobileIPProfile_t, 685
pMNC
QmiNas3GppNetworkInfo, 762
unpack_nas_GetHomeNetwork3GPP2_t, 1047
pMNCIncPCSDigStat
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 629
pMNHA
pack_wds_SetMobileIPProfile_t, 685
pMNRInfo
nasInitNetworkReg, 535
pack_nas_SLQSInitiateNetworkRegistration_t, 612
pMTMessageInfo
SMSEventInfo_s, 875
pMagneticDeviation
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pManString
_SLQSSwiGetHostDevInfoParams, 96

_SLQSSwiSetHostDevInfoParams, 98
pManagedRoamingInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pMaxAllowedPktSz
swiQosFlow, 906
pMaxChannelRXRate
WdsConnectionRateElmnts, 1310
pMaxChannelTXRate
WdsConnectionRateElmnts, 1310
pMaxStorageSizeReq
pack_sms_SLQSSmsGetMaxStorageSize_t, 643
pMaxStorageSizeResp
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1119
pMdmCallDurationActive
getDUNCallInfoResp, 264
pMeidString
_SLQSSwiGetSerialNoExtParams, 97
pMessage
cdmaMsgDecodingParams, 167
cdmaMsgEncodingParams, 168
pack_sms_SaveSMS_t, 635
pack_sms_SendSMS_t, 635
slqssendasyncsmsparams_s, 845
slqssendsmsparams_s, 846
sms_sendAsyncsmsParams, 865
wcdmaLongMsgDecodingParams, 1285
wcdmaMsgDecodingParams, 1287
pMessageID
cdmaMsgDecodingParams, 167
pMessageIndex
pack_sms_SLQSDeleteSMS_t, 637
unpack_sms_SaveSMS_t, 1113
pMessageMode
pack_sms_SLQSDeleteSMS_t, 637
pack_sms_SLQSGetSMS_t, 638
pack_sms_SLQSGetSMSList_t, 639
pack_sms_SLQSModifySMSStatus_t, 640
sms_maxStorageSizeReq, 859
smsMaxStorageSizeReq, 875
pMessageModeInfo
SMSEventInfo_s, 874
pMessageProtocol
unpack_sms_SLQSSmsGetMessageProtocol_t, 1119
pMessageSize
cdmaMsgEncodingParams, 168
pMessageTag
pack_sms_SLQSDeleteSMS_t, 637
pMicMute
SetM2MAudioProfileReq, 820
pMinBasedIMSI
nasGet3GPP2SubscriptionInfoResp, 524
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1070
pMinIntervalTime
LOCStartReq, 382
pack_loc_Start_t, 605
pMinPolicedPktSz
swiQosFlow, 907
pMinSessionExpiryTimer
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pMitigationDevList
TmdGetMitigationDevListResp, 923
pMitigationDevListLen
TmdGetMitigationDevListResp, 923
pMncPcsDigitStatus
nasInitNetworkReg, 535
pack_nas_SLQSInitiateNetworkRegistration_t, 612
pMncPcsStatus
nasPLMNNameReq, 546
pack_nas_SLQSGetPLMNName_t, 611
pMode
pack_wds_SetMobileIPParameters_t, 684
pModePref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 629
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pModelString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pMtu
qmiWdsRunTimeSettings, 767
pNAI
pack_wds_SetMobileIPProfile_t, 685
pNAMNameInfo
nasGet3GPP2SubscriptionInfoResp, 524
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1070
pNID
unpack_nas_GetHomeNetwork3GPP2_t, 1048
pNITZInformation
nasOperatorNameResp, 541
unpack_nas_SLQSGetOperatorNameData_t, 1059
pNSEnable
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pNSSAudioCtrl
voiceInfoRec, 1262
pNSSRelease
voiceInfoRec, 1262
pNamID
pack_voice_SLQSVoiceGetConfig_t, 669
voiceGetConfigReq, 1257
pName
pack_wds_SetDefaultProfile_t, 682
unpack_nas_GetHomeNetwork3GPP2_t, 1047
pNameString
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
pNegoDnsSrvrPref
LibpackProfile3GPP2, 340

LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pNeighborSetPilotPN
nas_NetworkStat1x, 470
NetworkStat1x, 563
pNetSelPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 629
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pNetworkInfo
_slqsNetworkScanInfo, 88
pNetworkInfoInstances
_slqsNetworkScanInfo, 88
pNetworkInfoLen
CurrDataSysStat, 196
pNetworkRejectInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pNetworkStat1x
NetworkDebugResp, 561
unpack_nas_SLQSSwiNetworkDebug_t, 1081
pNetworkStatEVDO
NetworkDebugResp, 561
unpack_nas_SLQSSwiNetworkDebug_t, 1081
pNetworkTimeInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pNewImsRegStatus
IMSARegistrationStatus, 312
pNewPwdData
voiceSUPSInfo, 1280
pNotifStore
_SLQSAVMSSetSettingsReqPara, 85
pNotificationStore
_SLQSAVMSSettings, 85
pNumSupUSBComps
USBCompParams, 1190
pNumberOfPhySlot
UIMGetSlotsStatusResp, 965
unpack_uim_SLQSUIMGetSlotsStatus_t, 1134
pNw2DescDisp
unpack_nas_GetHomeNetwork3GPP2_t, 1048
pNw2DescEnc
unpack_nas_GetHomeNetwork3GPP2_t, 1048
pNw2DescLen
unpack_nas_GetHomeNetwork3GPP2_t, 1048
pNw2MCC
unpack_nas_GetHomeNetwork3GPP2_t, 1048
pNw2MNC
unpack_nas_GetHomeNetwork3GPP2_t, 1048
pNw2Name
unpack_nas_GetHomeNetwork3GPP2_t, 1048
pNxtHdrProto
swiQosFilter, 903
pOMADMEnabled
_SLQSAVMSSettings, 86
_SLQSOMADMSettings, 93
PORTNAM_LEN
qaGobiApiDcs.h, 1539
pOTASPStatus
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1142
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
voiceGetAllCallInfo, 1244
pObjectVer
NetworkDebugResp, 561
unpack_nas_SLQSSwiNetworkDebug_t, 1081
pOffLength
voiceDTMFEventInfo, 1240
pOnLength
voiceDTMFEventInfo, 1240
pOpaqueIdentifier
LocInjectSensorDataReq, 379
QmiCbkLocInjectSensorDataInd, 749
pOperation
unpack_pds_GetPDSDefaults_t, 1088
pOperatorNameString
nasOperatorNameResp, 541
unpack_nas_SLQSGetOperatorNameData_t, 1059
pOperatorPLMNList
nasOperatorNameResp, 541
unpack_nas_SLQSGetOperatorNameData_t, 1059
pOptList
wds_DHCPv4OptionList, 1298
WdsDHCPv4ClientLeaseInd, 1311
WdsDHCPv4OptionList, 1315
wdsDhcpv4OptionList, 1315
pOptVal
DHCPOption, 220
wds_DHCPOpt, 1296
pOptions
DHCPOptionList, 221
pPCMParams
SetM2MAudioAVCFGReq, 818
pPCSCFAddrPCO
qmiWdsRunTimeSettings, 767
pPCSCFFQDNAddrList
qmiWdsRunTimeSettings, 767
pPCSCFPort
GetRegMgrConfigParams, 278
pPCSDigitInfo
_slqsNetworkScanInfo, 88
pPCSDigitInstances
_slqsNetworkScanInfo, 88
pPCSInstance
unpack_nas_PerformNetworkScan_t, 1052
pPCSInstanceSize
unpack_nas_PerformNetworkScan_t, 1052
pPDNInactivTimeout
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPDNInactivTimeout3GPP2
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351

Profile3GPP2, 731
pPDPType
qmiWdsRunTimeSettings, 767
pPDPtype
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPDUMessage
wcdmaMsgEncodingParams, 1288
pPLMNNetworkName
nasOperatorNameResp, 541
unpack_nas_SLQSGetOperatorNameData_t, 1059
pPRLPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 629
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pPRLPreference
dmsCurrentPRLInfo, 224
pPRLVersion
dmsCurrentPRLInfo, 224
pPWD
_SLQSAVMSSettingsAPNInfo, 86
pPacketsCountRX
QosEventInfo, 770
slqsWdsEventInfo, 854
pPacketsCountTX
QosEventInfo, 770
slqsWdsEventInfo, 854
pPartNum
wcdmaLongMsgDecodingParams, 1286
pPass
pack_wds_SLQSStartDataSession_t, 695
pPassword
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
pack_wds_SetDefaultProfile_t, 682
Profile3GPP, 725
ssdatasession_params, 888
pPasswordSize
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPcscfAddrUsingDhcp
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPcscfAddrUsingPCO
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPdnType
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pPdpAccessConFlag
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPdpContext
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPdpDataCompType
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPdpHdrCompType
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPdpStatusConfig
IMSAIndRegisterInfo, 309
pPeroidInfo
_SLQSAVMSSetSettingsReqPara, 85
pPeroidsInfo
_SLQSAVMSSettings, 86
pPersonalityListLength
HDRPersonalityInd, 297
HDRPersonalityResp, 297
HDRProtSubtypResp, 298
unpack_nas_SLQSSwiGetHDRPersonality_t, 1078
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1079
pPersonalizationStatus
UIMGetConfigurationResp, 964
unpack_uim_SLQSUIMGetConfiguration_t, 1133
pPhoneCtxtURI
GetIMSSMSConfigParams, 268
imsSMSConfigInfo, 320
SetIMSSMSConfigReq, 813
pPhoneCtxtURILen
GetIMSSMSConfigParams, 268
SetIMSSMSConfigReq, 813
pPhyCaAggPcellInfo
nasGetLTECphyCaResp, 526
pPhyCaAggScellDlBw
nasGetLTECphyCaResp, 526
pPhyCaAggScellIndType
nasGetLTECphyCaResp, 526
pPhyCaAggScellIndex
nasGetLTECphyCaResp, 526
pPhyCaAggScellInfo
nasGetLTECphyCaResp, 526
pPilotSetData
GetHRPDStatsResp, 267
unpack_nas_SLQSSwiGetHRPDStats_t, 1080
pPilotSetInfo
nas_PilotSetData, 480
PilotSetData, 712
pPkgDescLength
_SLQSOMADMSessionInfo, 91
pPkgDescription
_SLQSOMADMSessionInfo, 91
pPkgName
_SLQSOMADMSessionInfo, 91

pPkgNameLength
_SLQSOMADMSessionInfo, 91
pPktErrRate
swiQosFlow, 907
pPlasmaIDString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pPlmnId
nasNetworkReject, 539
pPollingTimer
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 86
pPositionSource
PDSPositionData, 703
pPositionSrc
LocInjectPositionReq, 378
pPppSessCloseTimer1x
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pPppSessCloseTimerDO
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pPrDNSIPv4Address
swiPDPRuntimeSettingsResp, 901
pPrDNSIPv6Address
swiPDPRuntimeSettingsResp, 901
pPrPCSCFIPv4Address
swiPDPRuntimeSettingsResp, 901
pPrPCSCFIPv6Address
swiPDPRuntimeSettingsResp, 901
pPrecedence
swiQosFilter, 903
pPrecisionDilution
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pPrefNetwork
CurrDataSysStat, 196
pPrefVoiceDomain
pack_voice_SLQSVoiceSetConfig_t, 675
voiceSetConfigReq, 1272
pPrefVoicePrivacy
pack_voice_SLQSVoiceGetConfig_t, 669
voiceGetConfigReq, 1257
pPrefVoiceSO
pack_voice_SLQSVoiceGetConfig_t, 669
pack_voice_SLQSVoiceSetConfig_t, 675
voiceGetConfigReq, 1257
voiceSetConfigReq, 1272
pPrefVoiceSOStatus
unpack_voice_SLQSVoiceSetConfig_t, 1160
voiceSetConfigResp, 1273
pPreferred
QmiNas3GppNetworkInfo, 762
pPriCSCFPort
imsRegMgrConfigInfo, 318
SetRegMgrConfigReq, 824
pPriCSCFPortName
GetRegMgrConfigParams, 278
pPriCSCFPortNameLen
GetRegMgrConfigParams, 278
pPriDNSIPv4AddPref
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPriDNSIPv6addpref
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPriV6DnsAddress
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pPrimaryDNSV4
qmiWdsRunTimeSettings, 768
pPrimaryDNSV6
qmiWdsRunTimeSettings, 768
pPrimaryHA
pack_wds_SetMobileIPProfile_t, 685
pPrimaryID
LibpackProfile3GPP, 334
LibPackprofile_3GPP, 345
Profile3GPP, 725
pPrimaryV4DnsAddress
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pPriority
cdmaMsgDecodingParams, 167
cdmaMsgEncodingParams, 168
pPrivacy
cdmaMsgDecodingParams, 167
pProfSz
NWProfile, 568
pProfValues
NWProfile, 569
pProfileID
CreateProfileIn, 191
ModifyProfileIn, 407
qmiWdsRunTimeSettings, 768
unpack_wds_SLQSCreateProfile_t, 1172
pProfileId
pack_wds_SLQSCreateProfile_t, 686
pack_wds_SLQSModifyProfile_t, 690
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 693
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 694
WdsDHCPv4ClientLeaseInd, 1311
WdsDHCPv4Config, 1312
pProfileId3GPP
ssdatasession_params, 888
pProfileId3GPP2
ssdatasession_params, 888
swiQosFlow, 907

pProfileIndex
CreateProfileOut, 192
pProfileList
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 691
pProfileName
qmiWdsRunTimeSettings, 768
pProfileSettings
unpack_wds_SLQSGetProfileSettings_t, 1177
pProfileType
CreateProfileIn, 191
CreateProfileOut, 192
ModifyProfileIn, 407
pack_wds_SLQSCreateProfile_t, 686
pack_wds_SLQSModifyProfile_t, 690
pProfilename
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pProfilenameSize
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pProtoSubTypElmnt
HDRProtSubtypResp, 298
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1079
pProtocol
pack_nas_SetCDMANetworkParameters_t, 608
pProtocolSubtypeElement
HDRPersonalityInd, 297
HDRPersonalityResp, 297
unpack_nas_SLQSSwiGetHDRPersonality_t, 1078
pQFlowState
QosFlowInfo, 771
pQmiInterfaceInfo
_packetSrvStatus, 70
slqsSessionStateInfo, 847
slqsWdsEventInfo, 854
pQosClassID
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pRAT
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 629
pRATINstance
unpack_nas_PerformNetworkScan_t, 1052
pRATInfo
_slqsNetworkScanInfo, 88
pRATInstanceSize
unpack_nas_PerformNetworkScan_t, 1052
pRATInstances
_slqsNetworkScanInfo, 88
pRATStatus
imsaRatStatusInfo, 311
pRATType
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
PREFERRED_INDEX
qaNasPerformNetworkScan.h, 1862
pRFBandInfoElements
QmiNasGetRFBandInfoResp, 762
PRI_UPDATE_FAIL
qaGobiApiFms.h, 1588
PRIString
unpack_dms_GetFirmwareRevision_t, 997
unpack_dms_GetFirmwareRevisions_t, 998
PRLInd
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
PRLPTlv
NASQmiCbkNasSystemSelPrefInd, 549
PRLPref
NASPRLPreferenceTlv, 548
pRMAutoConnect
custFeaturesInfo, 203
pRSRPThresList
nas_RSRPThresh, 490
RSRPThresh, 784
pRSRPThresh
pack_nas_SLQSConfigSigInfo_t, 610
sigInfo, 833
pRSRQThresList
nas_RSRQThresh, 491
RSRQThresh, 785
pRSRQThresh
pack_nas_SLQSConfigSigInfo_t, 610
sigInfo, 833
pRSSIThresList
nas_RSSIThresh, 492
RSSIThresh, 786
pRSSIThresh
pack_nas_SLQSConfigSigInfo_t, 610
sigInfo, 833
pRTPRTCPInactTimer
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pRXAGCList
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pRXAIG
RXAGCList, 787
pRXAVCAGCSwitch
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pRXAVCList
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pRXChain0Info
nasGetTxRxInfoResp, 531
unpack_nas_SLQSNasGetTxRxInfo_t, 1073
pRXChain1Info
nasGetTxRxInfoResp, 531

unpack_nas_SLQSNasGetTxRxInfo_t, 1073
pRXComprSlope
RXAGCList, 787
pRXComprThres
RXAGCList, 787
pRXDroppedCount
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1318
pRXExpSlope
RXAGCList, 787
pRXExpThres
RXAGCList, 787
pRXOKBytesCount
getDUNCallInfoResp, 264
pRXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1318
pRXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1318
pRXPCMIIRFltr
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pRXPacketErrors
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1318
pRXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pRXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pRXStaticGain
RXAGCList, 787
pRXTotalBytes
unpack_wds_GetByteTotals_t, 1163
WdsByteTotalsElmnts, 1308
pRadioInterface
nasNetworkTime, 540
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1074
pRankIndicatorInd
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
pRatHandoverStatusConfig
IMSAIndRegisterInfo, 309
pRawHorConfidence
LocInjectPositionReq, 378
pRawHorUncCircular
LocInjectPositionReq, 378
pReRegPeriod
pack_wds_SetMobileIPParameters_t, 684
pReRegTraffic
pack_wds_SetMobileIPParameters_t, 684
pReadAcknowledgementReq
cdmaMsgDecodingParams, 167
pReadResult
UIMReadTransparentResp, 969
unpack_uim_ReadTransparent_t, 1129
pReason
voiceSUPSInfo, 1280
pReasonMask
unpack_dms_GetOfflineReason_t, 1001
pRecurrenceType
LOCStartReq, 382
pack_loc_Start_t, 605
pRedirNumInfo
voiceInfoRec, 1262
pRefData
FirmwareUpdatStat, 244
pRefString
FirmwareUpdatStat, 244
pRefStringLen
FirmwareUpdatStat, 244
pReferenceNum
wcdmaLongMsgDecodingParams, 1286
pRefreshEvent
UIMRefreshGetLastEventResp, 972
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 1135
pRegCallStatInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 857
sms_setIndicationReg, 866
pRegDTMFEvents
pack_voice_SLQSVoiceIndicationRegister_t, 670
voiceIndicationRegisterInfo, 1260
pRegInd
_getTransLayerInfoResp, 66
sms_getTransLayerInfo, 858
pRegMgrConfigEvents
imsCfgIndRegisterInfo, 317
pRegStatus
_getTransNWRegInfoResp, 67
sms_getTransNWRegInfo, 858
pRegStatusConfig
IMSAIndRegisterInfo, 309
pRegStatusErrorCode
imsaRegStatusInfo, 312
pRegTransLayerInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 857
sms_setIndicationReg, 866
pRegTransNWRegInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 857
sms_setIndicationReg, 866
pRegVoicePrivacyEvents
pack_voice_SLQSVoiceIndicationRegister_t, 670
voiceIndicationRegisterInfo, 1260
pRejectReason
pack_swioma_SLQSOMADMSendSelection_t, 646

pRelValidity
cdmaMsgEncodingParams, 168
pRelativeValidity
cdmaMsgDecodingParams, 167
pRemainingRetries
UIMDepersonalizationResp, 961
UIMPinResp, 966
unpack_uim_ChangePin_t, 1128
unpack_uim_SetPinProtection_t, 1130
unpack_uim_SLQSUIMDepersonalization_t, 1132
unpack_uim_UnblockPin_t, 1136
unpack_uim_VerifyPin_t, 1136
pRemotePartyName
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pRemotePartyNum
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pReportChannelRate
getDUNCallInfoReq, 261
pack_wds_SLQSGetDUNCallInfo_t, 687
pReportConnStatus
getDUNCallInfoReq, 261
pack_wds_SLQSGetDUNCallInfo_t, 687
pReportDataBearerTech
getDUNCallInfoReq, 261
pack_wds_SLQSGetDUNCallInfo_t, 687
pReportDormStatus
getDUNCallInfoReq, 261
pack_wds_SLQSGetDUNCallInfo_t, 687
pReqFieldsList
IMSASupportedFieldsResp, 315
pReqMitigationLvl
TmdGetMitigationLvlResp, 925
pReqSettings
pack_wds_SLQSGetRuntimeSettings_t, 689
pRequestOptionList
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 694
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1184
WdsDHCPv4Config, 1312
pRequestedTag
pack_sms_SLQSGetSMSList_t, 639
pRespData
USSDRespFNetwork, 1191
pRespFieldsList
IMSASupportedFieldsResp, 315
pRetryCount
_SLQSOMADMSessionInfo, 91
pRetryInterval
pack_wds_SetMobileIPParameters_t, 684
pRetryLimit
pack_wds_SetMobileIPParameters_t, 684
pRetryMessage
slqssendasyncsmsparams_s, 845
sms_sendAsyncsmsParams, 865
pRetryMessageId
slqssendasyncsmsparams_s, 845
sms_sendAsyncsmsParams, 865
pRevInUse
CDMASysInfo, 174
nas_CDMASysInfo, 418
pRevInUseValid
CDMASysInfo, 174
nas_CDMASysInfo, 419
pRevTunneling
pack_wds_SetMobileIPProfile_t, 685
pRingBackTimer
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pRingingTimer
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pRoamPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 629
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pRoamTimer
pack_voice_SLQSVoiceGetConfig_t, 669
voiceGetConfigReq, 1257
pRoamTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1157
voiceGetConfigResp, 1259
pRoamTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 675
voiceSetConfigReq, 1272
pRoamTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1160
voiceSetConfigResp, 1273
pRoaming
pack_nas_SetCDMANetworkParameters_t, 608
QmiNas3GppNetworkInfo, 762
pRscp
nasSigInfo, 552
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1075
pRxFilter
swiQosModifyReq, 908
swiQosReq, 909
pRxFlow
swiQosGranted, 907
swiQosModifyReq, 908
swiQosReq, 909
pRxQFilter
QosFlowInfo, 771
pRxQFlowGranted
QosFlowInfo, 771
PSDetachReq, 733
pDetachAction, 734
PSDomain
unpack_nas_GetServingNetwork_t, 1051
pSID
unpack_nas_GetHomeNetwork3GPP2_t, 1048

pSIPConfigEvents
imsCfgIndRegisterInfo, 317
pSIPLocalPort
GetSIPConfigResp, 279
imsSIPConfigInfo, 319
SetSIPConfigReq, 830
pSMSAttemptedFlag
getDyingGaspStatistics, 265
packgetDyingGaspStatistics, 699
pSMSCAddress
pack_sms_SetSMSCAddress_t, 636
unpack_sms_GetSMSCAddress_t, 1112
pSMSCAddressInfo
SMSEventInfo_s, 875
pSMSCType
pack_sms_SetSMSCAddress_t, 636
unpack_sms_GetSMSCAddress_t, 1113
pSMSConfigEvents
imsCfgIndRegisterInfo, 317
pSMSFormat
GetIMSSMSConfigParams, 268
imsSMSConfigInfo, 320
SetIMSSMSConfigReq, 813
pSMSOnIMSInfo
SMSEventInfo_s, 875
pSMSOverIPNwInd
GetIMSSMSConfigParams, 268
imsSMSConfigInfo, 320
SetIMSSMSConfigReq, 813
pSMSSupport
custFeaturesInfo, 203
pSMSSvcRAT
imsaSvcStatusInfo, 316
pSMSSvcStatus
imsaSvcStatusInfo, 316
pSPC
pack_nas_SetCDMANetworkParameters_t, 608
pack_wds_SetMobileIPParameters_t, 684
pSV
BdsSVInfo, 141
loc_BdsSVInfo, 359
loc_SVInfo, 369
SVInfo, 892
pSVInfo
LocDelAssDataReq, 372
pack_loc_Delete_Assist_Data_t, 591
pSWVerString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pSarRFState
unpack_sar_SLQSGetRfSarState_t, 1112
pSatelliteInfo
gnssSvInfoNotification, 282
unpack_loc_GnssSvInfo_Ind_t, 1035
pScAddr
wcdmaLongMsgDecodingParams, 1286
wcdmaMsgDecodingParams, 1287
pScAddrLength
wcdmaLongMsgDecodingParams, 1286
wcdmaMsgDecodingParams, 1287
pScanResult
_slqsNetworkScanInfo, 88
unpack_nas_PerformNetworkScan_t, 1052
pSccRxInfo
LteSccRxInfoResp, 394
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1081
pScrAmrEnable
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pScrAmrWbEnable
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pSeDNSIPv4Address
swiPDPRuntimeSettingsResp, 901
pSeDNSIPv6Address
swiPDPRuntimeSettingsResp, 901
pSePCSCFIPv4Address
swiPDPRuntimeSettingsResp, 901
pSePCSCFIPv6Address
swiPDPRuntimeSettingsResp, 901
pSecDNSIPv4AddPref
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pSecDNSIPv6addpref
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pSecV6DnsAddress
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pSecondaryDNSV4
qmiWdsRunTimeSettings, 768
pSecondaryDNSV6
qmiWdsRunTimeSettings, 768
pSecondaryFlag
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pSecondaryHA
pack_wds_SetMobileIPProfile_t, 686
pSecondaryV4DnsAddress
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pSectorID
nas_NetworkStatEVDO, 472
NetworkStatEVDO, 565
pSendSmsParams
pack_sms_SLQSSendAsyncSMS_t, 640
pSenderAddr
cdmaMsgDecodingParams, 167
wcdmaLongMsgDecodingParams, 1286

wcdmaMsgDecodingParams, 1287
pSenderAddrLength
cdmaMsgDecodingParams, 167
wcdmaLongMsgDecodingParams, 1286
wcdmaMsgDecodingParams, 1287
pSensorDataUsage
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pServerAddrList
qmiWdsRunTimeSettings, 768
pServiceClass
pack_voice_SLQSVoiceSetSUPSService_t, 677
voiceSetSUPSServiceReq, 1276
pServiceOption
slqssendasyncsmsparams_s, 845
sms_sendAsyncsmsParams, 865
pServiceStatusConfig
IMSAIndRegisterInfo, 309
pServingSystemInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pSessionExpiryTimer
GetIMSVoIPConfigResp, 271
imsVoIPConfigInfo, 323
SetIMSVoIPConfigReq, 817
pSessionIDv4
GetSessionIDResp, 278
pSessionIDv6
GetSessionIDResp, 278
pSessionState
_SLQSOMADMSessionInfo, 91
pSessionType
_SLQSOMADMSessionInfo, 91
pSetIndicationRegReq
pack_sms_SLQSSetIndicationRegister_t, 641
pSetRoutesReq
pack_sms_SLQSSmsSetRoutes_t, 643
pSettingResp
GetIMSSMSConfigParams, 268
GetIMSUserConfigParams, 269
GetIMSVoIPConfigResp, 271
GetRegMgrConfigParams, 278
GetSIPConfigResp, 279
SetIMSSMSConfigResp, 814
SetIMSUserConfigResp, 815
SetIMSVoIPConfigResp, 818
SetRegMgrConfigResp, 824
SetSIPConfigResp, 831
pSeverity
_SLQSOMADMSessionInfo, 91
pSigCompEnabled
GetSIPConfigResp, 279
imsSIPConfigInfo, 319
SetSIPConfigReq, 830
pSigIndReq
pack_nas_SLQSSetSignalStrengthsCallback_t, 624
pSignalInfo
voiceInfoRec, 1262
pSignalStrengthInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pSmsOnIms
slqssendasyncsmsparams_s, 845
slqssendsmsparams_s, 846
sms_sendAsyncsmsParams, 865
pSmsServiceRat
IMSAServiceStatus, 314
pSmsServiceStatus
IMSAServiceStatus, 314
pSmsStorage
unpack_sms_SLQSSwiGetSMSStorage_t, 1120
pSource
_SLQSOMADMSessionInfo, 91
pSourceIP
LibPackTFTIDParams, 354
TFTIDParams, 920
pSourceLength
_SLQSOMADMSessionInfo, 91
pSpeedHorizontal
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pSpeedUnc
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pSpeedVertical
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pSpeedVerticalUnc
QmiCbkLocBestAvailPosInd, 745
unpack_loc_BestAvailPos_Ind_t, 1032
pSrcRAT
imsaRatStatusInfo, 311
pSrvDomainPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 629
unpack_nas_SLQSGetSysSelectionPref_t, 1069
pSrvOpt
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pSrvRegRestriction
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 629
pSrvcProviderName
nasOperatorNameResp, 541
unpack_nas_SLQSGetOperatorNameData_t, 1059

pStage0Val
RXPCMIIRFltr, 790
TXPCMIIRFltr, 933
pStage1Val
RXPCMIIRFltr, 790
TXPCMIIRFltr, 933
pStage2Val
RXPCMIIRFltr, 790
TXPCMIIRFltr, 933
pStage3Val
RXPCMIIRFltr, 790
TXPCMIIRFltr, 933
pStage4Val
RXPCMIIRFltr, 790
TXPCMIIRFltr, 933
pStageCnt
RXPCMIIRFltr, 790
TXPCMIIRFltr, 933
pStatMask
pack_wds_GetPacketStatistics_t, 680
WdsPktStatisticsReq, 1319
pStatus
_SLQSOMADMSessionInfo, 91
pSubnetMaskV4
qmiWdsRunTimeSettings, 768
pSubscribeTimer
GetSIPConfigResp, 279
imsSIPConfigInfo, 319
SetSIPConfigReq, 830
pSubscriptionInfoInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pSupUSBComps
USBCompParams, 1190
pSupportedMsgList
IMSASupportedMsgInfo, 315
pSuppressSysInfoInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pSuppsNotifEvents
pack_voice_SLQSVoiceIndicationRegister_t, 670
voiceIndicationRegisterInfo, 1260
pSvUsedforFix
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pSvcClass
pack_voice_SLQSVoiceGetCallBarring_t, 666
pack_voice_SLQSVoiceGetCallForwardingStatus-
_t, 667
pack_voice_SLQSVoiceGetCallWaiting_t, 668
unpack_voice_SLQSVoiceGetCallBarring_t, 1143
unpack_voice_SLQSVoiceGetCallWaiting_t, 1148
voiceGetCallBarringReq, 1244
voiceGetCallBarringResp, 1246
voiceGetCallFWReq, 1246
voiceGetCallWaitInfo, 1249
voiceSUPSInfo, 1280
pSvcType
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238
pSwiGetResetInd
dmsIndicationRegisterReq, 227
pSysInfoInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pSysInfoNoChange
nasSysInfo, 558
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pSystemSelectionInd
nasIndicationRegisterReq, 534
pack_nas_SLQSNasIndicationRegisterExt_t, 620
pTCPDstPort
swiQosFilter, 903
pTCPSrcPort
swiQosFilter, 903
pTDSCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMAECIOThreshList
nas_TDSCDMAECIOThresh, 504
TDSCDMAECIOThresh, 913
pTDSCDMARSCPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMARSCPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMARSCPThreshList
nas_TDSCDMARSCPThresh, 505
TDSCDMARSCPThresh, 914
pTDSCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMARSSIThreshList
nas_TDSCDMARSSIThresh, 505
TDSCDMARSSIThresh, 914
pTDSCDMASINRCONFThresh
sigInfo, 833
pTDSCDMASINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMASINRThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pTDSCDMASINRThreshList
nas_TDSCDMASINRThresh, 506
TDSCDMASINRThresh, 917
pTDSCDMASigInfoExt

nasGetSigInfoResp, 527
nasSigInfo, 552
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1075
pTDSCDMASigInfoRscp
nasGetSigInfoResp, 527
pTFTID1Params
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pTFTID2Params
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pTTYConfigStatus
unpack_voice_SLQSVoiceSetConfig_t, 1160
voiceSetConfigResp, 1273
pTTYMode
pack_voice_SLQSVoiceGetConfig_t, 669
pack_voice_SLQSVoiceSetConfig_t, 675
voiceGetConfigReq, 1257
voiceSetConfigReq, 1272
pTXAGCList
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pTXAIG
TXAGCList, 931
pTXAVCSwitch
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pTXComprSlope
TXAGCList, 931
pTXComprThres
TXAGCList, 931
pTXDroppedCount
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pTXExpSlope
TXAGCList, 931
pTXExpThres
TXAGCList, 931
pTXGain
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pTXInfo
nasGetTxRxInfoResp, 531
unpack_nas_SLQSNasGetTxRxInfo_t, 1073
pTXOKBytesCount
getDUNCallInfoResp, 264
pTXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pTXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pTXPCMIIRFltr
GetAudioPathConfigResp, 255
SetAudioPathConfigReq, 809
pTXPacketErrors
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pTXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pTXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1170
WdsPktStatisticsElmnts, 1319
pTXStaticGain
TXAGCList, 931
pTXTotalBytes
unpack_wds_GetByteTotals_t, 1163
WdsByteTotalsElmnts, 1308
pTdsBandCapability
BandCapabilityResp, 140
pTdscdmaBandPref
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 629
pTech
pack_wds_SLQSStartDataSession_t, 695
pTechnology
qmiWdsRunTimeSettings, 768
ssdatasession_params, 888
pTechnologyMask
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pTextMsg
cdmaMsgDecodingParams, 167
cdmaMsgEncodingParams, 169
wcdmaLongMsgDecodingParams, 1286
wcdmaMsgDecodingParams, 1287
wcdmaMsgEncodingParams, 1288
pTextMsgLength
cdmaMsgDecodingParams, 167
wcdmaLongMsgDecodingParams, 1286
wcdmaMsgDecodingParams, 1287
pTgtRAT
imsaRatStatusInfo, 311
pTime
_SLQSOMADMSessionInfo, 91
SwiOTAMsg_s, 898
pTimeLength
_SLQSOMADMSessionInfo, 91
pTimeSrc
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pTimeStamp
getDyingGaspStatistics, 265
packgetDyingGaspStatistics, 699
PDSPositionData, 703
pTimeType
PDSPositionData, 703
pTimeUnc

QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pTimeZone
nasNetworkTime, 540
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1074
pTimeout
unpack_pds_GetPDSDefaults_t, 1088
pTimer
NasSwiIndReg, 554
pack_nas_SLQSNasSwiIndicationRegister_t, 621
pTimerSIPReg
GetSIPConfigResp, 279
imsSIPConfigInfo, 319
SetSIPConfigReq, 830
pTimerT1
GetSIPConfigResp, 279
imsSIPConfigInfo, 319
SetSIPConfigReq, 830
pTimerT2
GetSIPConfigResp, 279
imsSIPConfigInfo, 319
SetSIPConfigReq, 830
pTimerTf
GetSIPConfigResp, 279
imsSIPConfigInfo, 319
SetSIPConfigReq, 830
pTimerVal
pack_voice_SLQSVoiceSetSUPSService_t, 677
voiceSetSUPSServiceReq, 1276
pTimestampAge
LocInjectPositionReq, 378
pTimestampUtc
LocInjectPositionReq, 378
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pTokenBucket
swiQosFlow, 907
pTos
swiQosFilter, 903
pTotalBytesRX
QosEventInfo, 770
slqsWdsEventInfo, 855
pTotalBytesTX
QosEventInfo, 770
slqsWdsEventInfo, 855
pTotalNum
wcdmaLongMsgDecodingParams, 1286
pTrackingStatus
unpack_pds_GetPDSState_t, 1089
pTrafficClass
swiQosFlow, 907
pTranDstPort
swiQosFilter, 903
pTranSrcPort
swiQosFilter, 903
pTransLayerInfo
_getTransLayerInfoResp, 66
_transLayerInfoNotification, 112
sms_getTransLayerInfo, 858
pTransferRouteMTMessageInfo
SMSEventInfo_s, 875
pTransferStatInd
getDUNCallInfoReq, 261
pack_wds_SLQSGetDUNCallInfo_t, 688
pack_wds_SLQSWdsSetEventReport_t, 697
wdsSetEventReportReq, 1323
pTransferStatusReport
sms_setRoutesReq, 867
smsSetRoutesReq, 881
pTrueIMSI
nasGet3GPP2SubscriptionInfoResp, 524
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1070
pTxFilter
swiQosModifyReq, 908
swiQosReq, 909
pTxFlow
swiQosGranted, 907
swiQosModifyReq, 908
swiQosReq, 909
pTxQFilter
QosFlowInfo, 771
pTxQFlowGranted
QosFlowInfo, 771
pTypeCode
USSDRespFNetwork, 1191
pUATI
GetHRPDStatsResp, 267
unpack_nas_SLQSSwiGetHRPDStats_t, 1080
pUDPDstPort
swiQosFilter, 903
pUDPSrcPort
swiQosFilter, 903
pUMTSCellID
nasCellLocationInfoResp, 523
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pUMTSGrantedQoS
qmiWdsRunTimeSettings, 768
pUMTSInfo
nasCellLocationInfoResp, 523
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pUMTSMinQoS
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pUMTSMinQosSigInd
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726

pUMTSReqQoS
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pUMTSReqQoSSigInd
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pUSBComp
USBCompConfig, 1188
USBCompParams, 1190
pUSSDData
USSDNoWaitIndicationInfo, 1191
pUSSDInfo
unpack_voice_SLQSOriginateUSSD_t, 1137
USSResp, 1193
pUSSInfo
voiceSUPSInfo, 1280
pUTSvcRAT
imsaSvcStatusInfo, 316
pUTSvcStatus
imsaSvcStatusInfo, 316
pUUSInFo
pack_voice_SLQSVoiceDialCall_t, 665
voiceCallRequestParams, 1238
pUUSInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
pUimSlotsStatus
UIMGetSlotsStatusResp, 965
unpack_uim_SLQSUIMGetSlotsStatus_t, 1134
pUname
_SLQSAVMSSettingsAPNInfo, 86
pUpdateCompleteStatus
_SLQSOMADMSessionInfo, 91
pUser
pack_wds_SLQSStartDataSession_t, 695
pUserAcknowledgementReq
cdmaMsgDecodingParams, 167
pUserConfigEvents
imsCfgIndRegisterInfo, 317
pUserData
slqssendasyncsmsparams_s, 845
sms_sendAsyncsmsParams, 865
pUserId
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pUserIdSize
LibpackProfile3GPP2, 340
LibPackprofile_3GPP2, 351
Profile3GPP2, 731
pUsername
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
pack_wds_SetDefaultProfile_t, 682
Profile3GPP, 726
qmiWdsRunTimeSettings, 768
ssdatasession_params, 889
pUsernameSize
LibpackProfile3GPP, 335
LibPackprofile_3GPP, 346
Profile3GPP, 726
pUtServiceRat
IMSAServiceStatus, 314
pUtServiceStatus
IMSAServiceStatus, 314
pV4sessionId
WdsByteTotals, 1308
WdsConnectionRate, 1309
WdsPktStatisticsResp, 1320
pV6sessionId
WdsByteTotals, 1308
WdsConnectionRate, 1309
WdsPktStatisticsResp, 1320
pVOIPSvcRAT
imsaSvcStatusInfo, 316
pVOIPSvcStatus
imsaSvcStatusInfo, 316
pVTSvcRAT
imsaSvcStatusInfo, 316
pVTSvcStatus
imsaSvcStatusInfo, 316
pVerboseFailReasonType
unpack_wds_SLQSStartDataSession_t, 1185
pVerboseFailureReason
unpack_wds_SLQSStartDataSession_t, 1185
pVersionString
_SLQSSwiGetOSInfoParams, 97
_SLQSSwiSetOSInfoParams, 99
pVertConfidence
LocInjectPositionReq, 378
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 755
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pVertReliability
LocInjectPositionReq, 378
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 756
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pVertUnc
LocInjectPositionReq, 378
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocPositionReportInd, 756
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_PositionRpt_Ind_t, 1041
pVerticalConfidence
PDSPositionData, 703
pVerticalUnc
PDSPositionData, 703
pVoIPConfigEvents
imsCfgIndRegisterInfo, 318
pVoiceDomainPref
pack_voice_SLQSVoiceGetConfig_t, 669

voiceGetConfigReq, 1257
pVoiceDomainPrefStatus
unpack_voice_SLQSVoiceSetConfig_t, 1160
voiceSetConfigResp, 1273
pVoicePrivacy
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1142
unpack_voice_SLQSVoiceGetCallInfo_t, 1147
voiceCallInfoResp, 1236
voiceGetAllCallInfo, 1244
pVoipServiceRat
IMSAServiceStatus, 314
pVoipServiceStatus
IMSAServiceStatus, 314
pVolume
SetM2MAudioProfileReq, 820
pVsServiceRat
IMSAServiceStatus, 314
pVsServiceStatus
IMSAServiceStatus, 314
pVtServiceRat
IMSAServiceStatus, 314
pVtServiceStatus
IMSAServiceStatus, 314
pWCDMABER
GetErrRateResp, 266
unpack_nas_SLQSGetErrorRate_t, 1057
pWCDMACallBarringSysInfo
nasGetSysInfoResp, 530
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1066
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pWCDMACipherDomainSysInfo
nasGetSysInfoResp, 530
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1066
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pWCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pWCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pWCDMAECIOThreshList
nas_WCDMAECIOThresh, 516
WCDMAECIOThresh, 1283
pWCDMAInfoLTENeighborCell
nasCellLocationInfoResp, 523
unpack_nas_SLQSNasGetCellLocationInfo_t, 1071
pWCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pWCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 616
setSignalStrengthInfo, 829
pWCDMARSSIThreshList
nas_WCDMARSSIThresh, 517
WCDMARSSIThresh, 1289
pWCDMASSInfo
nasGetSigInfoResp, 527
pWCDMASigInfo
nasSigInfo, 552
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1075
pWCDMASrvStatusInfo
nasGetSysInfoResp, 530
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1066
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pWCDMASysInfo
nasGetSysInfoResp, 530
nasSysInfo, 558
unpack_nas_SLQSGetSysInfo_t, 1066
unpack_nas_SLQSSysInfoCallback_ind_t, 1084
pWcdmaUARFCN
nasSwiGetChannelLockResp, 553
nasSwiSetChannelLockReq, 555
pack_nas_SLQSNASSwiSetChannelLock_t, 622
unpack_nas_SLQSNASSwiGetChannelLock_t, 1076
pWifiState
PDSPosMethodStateReq, 704
pXid
QmiCbkLocBestAvailPosInd, 745
unpack_loc_BestAvailPos_Ind_t, 1032
pXtraDataState
PDSPosMethodStateReq, 704
pXtraTimeState
PDSPosMethodStateReq, 704
pack_dms_ActivateAutomatic
dms.h, 1337
pack_dms_ActivateAutomatic_t, 577
actCode, 577
pack_dms_GetActivationState
dms.h, 1338
pack_dms_GetBandCapability
dms.h, 1338
pack_dms_GetCrashAction
dms.h, 1338
pack_dms_GetCustFeature
dms.h, 1339
pack_dms_GetCustFeaturesV2
dms.h, 1339
pack_dms_GetCustFeaturesV2_t, 578
cust_id, 578
list_type, 578
Tlvresult, 578
pack_dms_GetDeviceCap
dms.h, 1339
pack_dms_GetDeviceCapabilities
dms.h, 1339
pack_dms_GetDeviceHardwareRev
dms.h, 1340
pack_dms_GetDeviceMfr
dms.h, 1340
pack_dms_GetDeviceSerialNumbers

dms.h, 1340
pack_dms_GetFSN
dms.h, 1342
pack_dms_GetFirmwareInfo
dms.h, 1341
pack_dms_GetFirmwareRevision
dms.h, 1341
pack_dms_GetFirmwareRevisions
dms.h, 1342
pack_dms_GetHardwareRevision
dms.h, 1342
pack_dms_GetIMSI
dms.h, 1343
pack_dms_GetManufacturer
dms.h, 1343
pack_dms_GetModelID
dms.h, 1343
pack_dms_GetNetworkTime
dms.h, 1344
pack_dms_GetOfflineReason
dms.h, 1344
pack_dms_GetPRLVersion
dms.h, 1345
pack_dms_GetPower
dms.h, 1344
pack_dms_GetSerialNumbers
dms.h, 1345
pack_dms_GetUSBComp
dms.h, 1345
pack_dms_GetVoiceNumber
dms.h, 1346
pack_dms_ResetToFactoryDefaults
dms.h, 1346
pack_dms_ResetToFactoryDefaults_t, 578
spc, 578
pack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1349
pack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1350
pack_dms_SLQSDmsSwiIndicationRegister_t, 582
resetInfoInd, 582
pack_dms_SLQSGetBandCapability
dms.h, 1350
pack_dms_SLQSGetERIFile
dms.h, 1351
pack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1351
pack_dms_SLQSSwiGetCrashInfo
dms.h, 1351
pack_dms_SLQSSwiGetCrashInfo_t, 582
clear, 582
pack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1352
pack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1352
pack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1352
pack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1353
pack_dms_SLQSSwiGetHostDevInfo
dms.h, 1353
pack_dms_SLQSSwiGetOSInfo
dms.h, 1353
pack_dms_SLQSSwiGetSerialNoExt
dms.h, 1354
pack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1354
pack_dms_SLQSSwiSetDyingGaspCfg_t, 582
pDestSMSContent, 583
pDestSMSNum, 583
pack_dms_SLQSSwiSetHostDevInfo
dms.h, 1354
pack_dms_SLQSSwiSetHostDevInfo_t, 583
manString, 583
modelString, 583
plasmaIDString, 583
swVerString, 584
pack_dms_SLQSSwiSetOSInfo
dms.h, 1355
pack_dms_SLQSSwiSetOSInfo_t, 584
nameString, 584
versionString, 584
pack_dms_SLQSUIMGetState
dms.h, 1355
pack_dms_SetActivationStatusCallback
dms.h, 1347
pack_dms_SetActivationStatusCallback_t, 579
activationState, 579
pack_dms_SetCrashAction
dms.h, 1347
pack_dms_SetCrashAction_t, 579
crashAction, 579
pack_dms_SetCustFeature
dms.h, 1347
pack_dms_SetCustFeature_t, 579
DHCPRelayEnabled, 580
DisableIMSI, 580
GPSLPM, 580
GPSSel, 580
GpsEnable, 580
IPFamSupport, 580
IsVoiceEnabled, 580
RMAutoConnect, 580
SMSSupport, 580
pack_dms_SetCustFeaturesV2
dms.h, 1348
pack_dms_SetCustFeaturesV2_t, 580
cust_id, 581
cust_value, 581
Tlvresult, 581
value_length, 581
pack_dms_SetEventReport
dms.h, 1348
pack_dms_SetEventReport_t, 581
mode, 581
pack_dms_SetFirmwarePreference

dms.h, 1348
pack_dms_SetPower
dms.h, 1349
pack_dms_SetPower_t, 581
mode, 581
Tlvresult, 581
pack_dms_SetUSBComp
dms.h, 1349
pack_dms_SetUSBComp_t, 581
Tlvresult, 581
USBComp, 582
pack_dms_UIMChangePIN
dms.h, 1355
pack_dms_UIMChangePIN_t, 584
id, 585
newValue, 585
oldValue, 585
pack_dms_UIMGetControlKeyStatus
dms.h, 1356
pack_dms_UIMGetControlKeyStatus_t, 585
facility, 585
pack_dms_UIMGetICCID
dms.h, 1356
pack_dms_UIMGetICCID_t, 585
Tlvresult, 586
pack_dms_UIMGetPINStatus
dms.h, 1357
pack_dms_UIMSetControlKeyProtection
dms.h, 1357
pack_dms_UIMSetControlKeyProtection_t, 586
facility, 586
facilityCk, 586
facilityState, 586
pack_dms_UIMSetPINProtection
dms.h, 1357
pack_dms_UIMSetPINProtection_t, 586
bEnable, 587
id, 587
value, 587
pack_dms_UIMUnblockControlKey
dms.h, 1358
pack_dms_UIMUnblockControlKey_t, 587
facility, 587
facilityCk, 588
pack_dms_UIMUnblockPIN
dms.h, 1358
pack_dms_UIMUnblockPIN_t, 588
id, 588
newPin, 588
pukValue, 588
pack_dms_UIMVerifyPIN
dms.h, 1358
pack_dms_UIMVerifyPIN_t, 588
id, 589
value, 589
pack_dms_ValidateSPC
dms.h, 1359
pack_fms_GetImagesPreference
fms.h, 1382
pack_fms_GetImagesPreference_t, 589
Tlvresult, 589
pack_fms_GetStoredImages
fms.h, 1383
pack_fms_GetStoredImages_t, 589
Tlvresult, 590
pack_fms_SetImagesPreference
fms.h, 1383
pack_fms_SetImagesPreference_t, 590
bForceDownload, 590
imageListSize, 590
modemindex, 590
pImageList, 590
Tlvresult, 590
pack_loc_Delete_Assist_Data_t, 590
pBdsSVInfo, 591
pCellDb, 591
pClkInfo, 591
pGnssData, 591
pSVInfo, 591
Tlvresult, 591
pack_loc_DeleteAssistData
loc.h, 1389
pack_loc_EventRegister
loc.h, 1389
pack_loc_EventRegister_t, 591
eventRegister, 594
Tlvresult, 594
pack_loc_SLQSLOCGetBestAvailPos
loc.h, 1390
pack_loc_SLQSLOCGetBestAvailPos_t, 595
Tlvresult, 595
xid, 595
pack_loc_SLQSLOCInjectPosition
loc.h, 1391
pack_loc_SLQSLOCInjectPosition_t, 595
altitudeSrcInfo, 599
altitudeWrtEllipsoid, 599
altitudeWrtMeanSeaLevel, 599
has_altitudeSrcInfo, 599
has_altitudeWrtEllipsoid, 599
has_horConfidence, 599
has_horReliability, 599
has_horUncCircular, 599
has_latitude, 600
has_longitude, 600
has_positionSrc, 600
has_rawHorConfidence, 600
has_rawHorUncCircular, 600
has_timestampAge, 600
has_timestampUtc, 600
has_vertConfidence, 600
has_vertRelicability, 600
has_vertUnc, 600
horConfidence, 600
horReliability, 600
horUncCircular, 600

latitude, 600
longitude, 600
positionSrc, 600
rawHorConfidence, 600
rawHorUncCircular, 600
timestampAge, 600
timestampUtc, 600
vertConfidence, 600
vertReliability, 600
vertUnc, 600
pack_loc_SLQSLOCInjectSensorData
loc.h, 1391
pack_loc_SLQSLOCInjectSensorData_t, 600
accelTemp, 602
acceleroData, 602
acceleroTimeSrc, 602
gyroData, 602
gyroTemp, 602
gyroTimeSrc, 602
has_accelTemp, 602
has_acceleroTimeSrc, 602
has_accleroData, 602
has_gyroData, 602
has_gyroTemp, 602
has_gyroTimeSrc, 602
has_opaqueId, 602
opaqueId, 602
pack_loc_SLQSLOCInjectUTCTime
loc.h, 1391
pack_loc_SLQSLOCInjectUTCTime_t, 602
timeMsec, 603
timeUncMsec, 603
pack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1392
pack_loc_SLQSLOCSetCradleMountConfig_t, 603
confidence, 603
has_confidence, 603
state, 603
pack_loc_SetExtPowerState
loc.h, 1390
pack_loc_SetExtPowerState_t, 594
extPowerState, 594
Tlvresult, 594
pack_loc_SetOperationMode
loc.h, 1390
pack_loc_SetOperationMode_t, 594
mode, 595
Tlvresult, 595
pack_loc_Start
loc.h, 1392
pack_loc_Start_t, 603
pApplicationInfo, 604
pConfigAltitudeAssumed, 605
pHorizontalAccuracyLvl, 605
pIntermediateReportState, 605
pMinIntervalTime, 605
pRecurrenceType, 605
SessionId, 605
Tlvresult, 605
pack_loc_Stop
loc.h, 1392
pack_loc_Stop_t, 605
SessionId, 605
Tlvresult, 605
pack_nas_GetACCOLC
nas.h, 1408
pack_nas_GetANAAAAuthenticationStatus
nas.h, 1408
pack_nas_GetCDMANetworkParameters
nas.h, 1408
pack_nas_GetHomeNetwork
nas.h, 1409
pack_nas_GetHomeNetwork3GPP2
nas.h, 1409
pack_nas_GetNetworkPreference
nas.h, 1409
pack_nas_GetRFInfo
nas.h, 1409
pack_nas_GetServingNetwork
nas.h, 1410
pack_nas_GetServingNetworkCapabilities
nas.h, 1410
pack_nas_GetSignalStrengths
nas.h, 1410
pack_nas_InitiateDomainAttach
nas.h, 1411
pack_nas_InitiateDomainAttach_t, 605
action, 606
pack_nas_PerformNetworkScan
nas.h, 1411
pack_nas_SLQSConfigSigInfo
nas.h, 1413
pack_nas_SLQSConfigSigInfo_t, 609
pECIOThresh, 610
pIOThresh, 610
pLTESNRThresh, 610
pLTESigRptCfg, 610
pRSRPThresh, 610
pRSRQThresh, 610
pRSSIThresh, 610
pack_nas_SLQSGetErrorRate
nas.h, 1413
pack_nas_SLQSGetNetworkTime
nas.h, 1414
pack_nas_SLQSGetOperatorNameData
nas.h, 1414
pack_nas_SLQSGetPLMNName
nas.h, 1414
pack_nas_SLQSGetPLMNName_t, 610
mcc, 611
mnc, 611
pMncPcsStatus, 611
pack_nas_SLQSGetServingSystem
nas.h, 1415
pack_nas_SLQSGetSignalStrength
nas.h, 1415

pack_nas_SLQSGetSysInfo
nas.h, 1415
pack_nas_SLQSGetSysSelectionPref
nas.h, 1416
pack_nas_SLQSInitiateNetworkRegistration
nas.h, 1416
pack_nas_SLQSInitiateNetworkRegistration_t, 611
pChangeDuration, 611
pMNRInfo, 612
pMncPcsDigitStatus, 612
regAction, 612
pack_nas_SLQSNASSwiGetChannelLock
nas.h, 1418
pack_nas_SLQSNASSwiSetChannelLock
nas.h, 1419
pack_nas_SLQSNASSwiSetChannelLock_t, 621
pack_nas_SLQSNasConfigSigInfo2
nas.h, 1416
pack_nas_SLQSNasConfigSigInfo2_t, 612
pHDRIODelta, 615
pHDRIOThresh, 615
pLTESigRptConfig, 616
pack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1416
pack_nas_SLQSNasGet3GPP2Subscription_t, 616
namID, 617
pack_nas_SLQSNasGetCellLocationInfo
nas.h, 1417
pack_nas_SLQSNasGetHDRColorCode
nas.h, 1417
pack_nas_SLQSNasGetSigInfo
nas.h, 1417
pack_nas_SLQSNasGetTxRxInfo
nas.h, 1418
pack_nas_SLQSNasGetTxRxInfo_t, 617
radio_if, 617
pack_nas_SLQSNasIndicationRegisterExt
nas.h, 1418
pack_nas_SLQSNasIndicationRegisterExt_t, 617
pDDTMInd, 620
pDualStandByPrefInd, 620
pErrorRateInd, 620
pHDRSessionCloseInd, 620
pLTECphyCa, 620
pManagedRoamingInd, 620
pNetworkRejectInd, 620
pNetworkTimeInd, 620
pServingSystemInd, 620
pSignalStrengthInd, 620
pSubscriptionInfoInd, 620
pSuppressSysInfoInd, 620
pSysInfoInd, 620
pSystemSelectionInd, 620
pack_nas_SLQSNasSwiIndicationRegister
nas.h, 1419
pack_nas_SLQSNasSwiIndicationRegister_t, 620
gsmUmtsDl, 621
gsmUmtsUl, 621
lteEmmDl, 621
lteEmmUl, 621
lteEsmDl, 621
lteEsmUl, 621
pRankIndicatorInd, 621
pTimer, 621
pack_nas_SLQSNasSwiModemStatus
nas.h, 1419
pack_nas_SLQSSetBandPreference
nas.h, 1420
pack_nas_SLQSSetBandPreference_t, 622
bandPref, 623
pack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1420
pack_nas_SLQSSetSignalStrengthsCallback_t, 624
bEnable, 624
pSigIndReq, 624
pack_nas_SLQSSetSysSelectionPref
nas.h, 1420
pack_nas_SLQSSetSysSelectionPref_t, 624
pAcqOrderPref, 628
pBandPref, 628
pCSGID, 628
pChgDuration, 628
pEmerMode, 628
pGWAcqOrderPref, 628
pLTEBandPref, 628
pModePref, 629
pNetSelPref, 629
pPRLPref, 629
pRAT, 629
pRoamPref, 629
pSrvDomainPref, 629
pSrvRegRestriction, 629
pTdscdmaBandPref, 629
pack_nas_SLQSSwiGetHDRPersonality
nas.h, 1421
pack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1421
pack_nas_SLQSSwiGetHRPDStats
nas.h, 1421
pack_nas_SLQSSwiGetLteCQI
nas.h, 1422
pack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1422
pack_nas_SLQSSwiNetworkDebug
nas.h, 1422
pack_nas_SLQSSwiPSDetach
nas.h, 1423
pack_nas_SLQSSwiPSDetach_t, 629
pDetachAction, 629
pack_nas_SetACCOLC
nas.h, 1411
pack_nas_SetACCOLC_t, 606
accolc, 606
spc, 606
pack_nas_SetCDMANetworkParameters
nas.h, 1412

pack_nas_SetCDMANetworkParameters_t, 606
pApplication, 607
pBroadcast, 607
pCustomSCP, 607
pForceRev0, 607
pProtocol, 608
pRoaming, 608
pSPC, 608
pack_nas_SetLURejectCallback
nas.h, 1412
pack_nas_SetNetworkPreference
nas.h, 1412
pack_nas_SetNetworkPreference_t, 608
Duration, 608
TechnologyPref, 608
Tlvresult, 608
pack_nas_SetRFInfoCallback
nas.h, 1412
pack_nas_SlqsGetLTECphyCAInfo
nas.h, 1413
pack_pds_GetPDSDefaults
pds.h, 1441
pack_pds_GetPDSState
pds.h, 1442
pack_pds_GetPortAutomaticTracking
pds.h, 1442
pack_pds_SetPDSDefaults
pds.h, 1442
pack_pds_SetPDSDefaults_t, 629
accuracy, 630
interval, 630
operation, 630
timeout, 630
pack_pds_SetPDSState
pds.h, 1443
pack_pds_SetPDSState_t, 630
enable, 630
pack_pds_SetPortAutomaticTracking
pds.h, 1443
pack_pds_SetPortAutomaticTracking_t, 630
bAuto, 631
pack_qmi_t, 631
msgid, 631
svc, 631
timeout, 631
xid, 631
pack_qos_SLQSQosGetNetworkStatus
qos.h, 1871
pack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 1871
pack_qos_SLQSQosSwiReadApnExtraParams_t, 631
apnId, 632
pack_qos_SLQSQosSwiReadDataStats
qos.h, 1872
pack_qos_SLQSQosSwiReadDataStats_t, 632
apnId, 632
pack_qos_SLQSSetQosEventCallback
qos.h, 1873
pack_qos_SLQSSetQosEventCallback_t, 632
enable, 633
pack_rms_GetSMSWake
rms.h, 1878
pack_rms_SetSMSWake
rms.h, 1878
pack_rms_SetSMSWake_t, 633
enabled, 633
wake_mask, 633
pack_sar_SLQSGetRfSarState
sar.h, 1879
pack_sar_SLQSSetRfSarState
sar.h, 1880
pack_sar_SLQSSetRfSarState_t, 633
RfSarState, 634
pack_sms_GetSMSCAddress
sms.h, 1886
pack_sms_SLQSDeleteSMS
sms.h, 1888
pack_sms_SLQSDeleteSMS_t, 636
pMessageIndex, 637
pMessageMode, 637
pMessageTag, 637
storageType, 637
pack_sms_SLQSGetIndicationRegister
sms.h, 1888
pack_sms_SLQSGetMessageWaiting
sms.h, 1889
pack_sms_SLQSGetSMS
sms.h, 1889
pack_sms_SLQSGetSMS_t, 637
messageIndex, 638
pMessageMode, 638
storageType, 638
pack_sms_SLQSGetSMSList
sms.h, 1890
pack_sms_SLQSGetSMSList_t, 638
pMessageMode, 639
pRequestedTag, 639
storageType, 639
pack_sms_SLQSGetSmsBroadcastConfig
sms.h, 1889
pack_sms_SLQSGetSmsBroadcastConfig_t, 638
mode, 638
pack_sms_SLQSGetTransLayerInfo
sms.h, 1890
pack_sms_SLQSGetTransNWRegInfo
sms.h, 1891
pack_sms_SLQSModifySMSStatus
sms.h, 1891
pack_sms_SLQSModifySMSStatus_t, 639
messageIndex, 640
messageTag, 640
pMessageMode, 640
storageType, 640
pack_sms_SLQSSendAsyncSMS
sms.h, 1891
pack_sms_SLQSSendAsyncSMS_t, 640

pSendSmsParams, 640
pack_sms_SLQSSetIndicationRegister
sms.h, 1892
pack_sms_SLQSSetIndicationRegister_t, 640
pSetIndicationRegReq, 641
pack_sms_SLQSSetSmsBroadcastActivation
sms.h, 1892
pack_sms_SLQSSetSmsBroadcastActivation_t, 641
broadcastActivate, 641
mode, 641
pack_sms_SLQSSetSmsBroadcastConfig
sms.h, 1892
pack_sms_SLQSSetSmsBroadcastConfig_t, 641
mode, 642
pBroadcastConfig, 642
pCDMABroadcastConfig, 642
pack_sms_SLQSSetSmsStorage
sms.h, 1893
pack_sms_SLQSSetSmsStorage_t, 642
smsStorage, 642
pack_sms_SLQSSmsGetMaxStorageSize
sms.h, 1893
pack_sms_SLQSSmsGetMaxStorageSize_t, 642
pMaxStorageSizeReq, 643
pack_sms_SLQSSmsGetMessageProtocol
sms.h, 1893
pack_sms_SLQSSmsSetRoutes
sms.h, 1894
pack_sms_SLQSSmsSetRoutes_t, 643
pSetRoutesReq, 643
pack_sms_SLQSSwiGetSMSStorage
sms.h, 1894
pack_sms_SaveSMS
sms.h, 1886
pack_sms_SaveSMS_t, 634
messageFormat, 635
messageSize, 635
pMessage, 635
storageType, 635
pack_sms_SendSMS
sms.h, 1887
pack_sms_SendSMS_t, 635
messageFormat, 635
messageSize, 635
pLinktimer, 635
pMessage, 635
pack_sms_SetNewSMSCallback
sms.h, 1887
pack_sms_SetNewSMSCallback_t, 635
status, 636
pack_sms_SetSMSCAddress
sms.h, 1888
pack_sms_SetSMSCAddress_t, 636
pSMSCAddress, 636
pSMSCType, 636
pack_swiloc_SwiLocGetAutoStart
swiloc.h, 1905
pack_swiloc_SwiLocSetAutoStart
swiloc.h, 1905
pack_swiloc_SwiLocSetAutoStart_t, 643
fix_rate, 644
fix_type, 644
function, 644
max_dist, 644
max_time, 644
set_fix_rate, 644
set_fix_type, 645
set_function, 645
set_max_dist, 645
set_max_time, 645
pack_swioma_SLQSOMADMAlertCallback
swioma.h, 1907
pack_swioma_SLQSOMADMCancelSession
swioma.h, 1908
pack_swioma_SLQSOMADMCancelSession_t, 645
sessionType, 645
pack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 1908
pack_swioma_SLQSOMADMGetSessionInfo_t, 645
SessionType, 646
pack_swioma_SLQSOMADMGetSettings
swioma.h, 1909
pack_swioma_SLQSOMADMSendSelection
swioma.h, 1909
pack_swioma_SLQSOMADMSendSelection_t, 646
pDeferTime, 646
pRejectReason, 646
selection, 646
pack_swioma_SLQSOMADMSetSettings
swioma.h, 1910
pack_swioma_SLQSOMADMSetSettings_t, 646
FOTAdownload, 647
pAutosdm, 647
pFwAutoCheck, 647
pack_swioma_SLQSOMADMStartSession
swioma.h, 1911
pack_swioma_SLQSOMADMStartSession_t, 647
sessionType, 648
pack_uim_ChangePin
uim.h, 1918
pack_uim_ChangePin_t, 648
changePIN, 649
EncryptedPIN1, 649
pIndicationToken, 649
pKeyReferenceID, 649
sessionInfo, 649
Tlvresult, 649
pack_uim_GetCardStatus
uim.h, 1919
pack_uim_ReadTransparent
uim.h, 1919
pack_uim_ReadTransparent_t, 649
fileIndex, 650
pEncryptData, 650
pIndicationToken, 650
readTransparent, 650

sessionInfo, 650
TlvresultI, 650
pack_uim_SLQSUIMAuthenticate
uim.h, 1920
pack_uim_SLQSUIMAuthenticate_t, 651
authData, 652
pIndicationToken, 652
sessionInfo, 652
pack_uim_SLQSUIMDepersonalization
uim.h, 1920
pack_uim_SLQSUIMDepersonalization_t, 652
depersonilisationInfo, 652
pack_uim_SLQSUIMEventRegister
uim.h, 1920
pack_uim_SLQSUIMEventRegister_t, 652
eventMask, 652
pack_uim_SLQSUIMGetConfiguration
uim.h, 1921
pack_uim_SLQSUIMGetConfiguration_t, 652
pConfigurationMask, 653
pack_uim_SLQSUIMGetFileAttributes
uim.h, 1921
pack_uim_SLQSUIMGetFileAttributes_t, 653
fileIndex, 654
pIndicationToken, 654
sessionInfo, 654
pack_uim_SLQSUIMGetSlotsStatus
uim.h, 1922
pack_uim_SLQSUIMPowerDown
uim.h, 1922
pack_uim_SLQSUIMPowerDown_t, 654
slot, 654
pack_uim_SLQSUIMPowerUp
uim.h, 1922
pack_uim_SLQSUIMPowerUp_t, 654
pIgnoreHotSwapSwitch, 655
slot, 655
pack_uim_SLQSUIMRefreshComplete
uim.h, 1923
pack_uim_SLQSUIMRefreshComplete_t, 655
refreshComplete, 655
sessionInfo, 655
pack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 1923
pack_uim_SLQSUIMRefreshGetLastEvent_t, 655
sessionInfo, 656
pack_uim_SLQSUIMRefreshOK
uim.h, 1923
pack_uim_SLQSUIMRefreshOK_t, 656
OKtoRefresh, 656
sessionInfo, 656
pack_uim_SLQSUIMRefreshRegister
uim.h, 1924
pack_uim_SLQSUIMRefreshRegister_t, 656
regRefresh, 657
sessionInfo, 657
pack_uim_SLQSUIMReset
uim.h, 1924
pack_uim_SLQSUIMSwitchSlot
uim.h, 1924
pack_uim_SLQSUIMSwitchSlot_t, 657
bLogicalSlot, 657
ulPhysicalSlot, 657
pack_uim_SetPinProtection
uim.h, 1919
pack_uim_SetPinProtection_t, 650
EncryptedPIN1, 651
pIndicationToken, 651
pKeyReferenceID, 651
pinProtection, 651
sessionInfo, 651
TlvresultI, 651
pack_uim_UnblockPin
uim.h, 1925
pack_uim_UnblockPin_t, 658
EncryptedPIN1, 658
pIndicationToken, 658
pKeyReferenceID, 658
pinProtection, 658
sessionInfo, 658
TlvresultI, 658
pack_uim_VerifyPin
uim.h, 1925
pack_uim_VerifyPin_t, 659
pEncryptedPIN1, 659
pIndicationToken, 659
pKeyReferenceID, 659
sessionInfo, 660
TlvresultI, 660
verifyPIN, 660
pack_voice_AnswerUSSD
voice.h, 1938
pack_voice_AnswerUSSD_t, 660
pInfo, 660
pack_voice_CancelUSSD
voice.h, 1938
pack_voice_OriginateUSSD
voice.h, 1938
pack_voice_OriginateUSSD_t, 660
pInfo, 660
pack_voice_SLQSOriginateUSSD
voice.h, 1939
pack_voice_SLQSOriginateUSSD_t, 660
ussDCS, 661
ussData, 661
ussLen, 661
pack_voice_SLQSVoiceALSSelectLine
voice.h, 1939
pack_voice_SLQSVoiceALSSelectLine_t, 661
lineValue, 661
pack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 1939
pack_voice_SLQSVoiceALSSetLineSwitching_t, 661
switchOption, 662
pack_voice_SLQSVoiceAnswerCall
voice.h, 1940

pack_voice_SLQSVoiceAnswerCall_t, 662
pCallId, 662
pack_voice_SLQSVoiceBindSubscription
voice.h, 1940
pack_voice_SLQSVoiceBindSubscription_t, 662
subsType, 663
pack_voice_SLQSVoiceBurstDTMF
voice.h, 1940
pack_voice_SLQSVoiceBurstDTMF_t, 663
pack_voice_SLQSVoiceDialCall
voice.h, 1941
pack_voice_SLQSVoiceDialCall_t, 663
callNumber, 665
pCLIRType, 665
pCUGInfo, 665
pCallPartySubAdd, 665
pCallType, 665
pEmergencyCategory, 665
pSvcType, 665
pUUSInFo, 665
pack_voice_SLQSVoiceEndCall
voice.h, 1941
pack_voice_SLQSVoiceEndCall_t, 665
pCallId, 665
pack_voice_SLQSVoiceGetAllCallInfo
voice.h, 1942
pack_voice_SLQSVoiceGetCLIP
voice.h, 1943
pack_voice_SLQSVoiceGetCLIR
voice.h, 1944
pack_voice_SLQSVoiceGetCNAP
voice.h, 1944
pack_voice_SLQSVoiceGetCOLP
voice.h, 1944
pack_voice_SLQSVoiceGetCOLR
voice.h, 1945
pack_voice_SLQSVoiceGetCallBarring
voice.h, 1942
pack_voice_SLQSVoiceGetCallBarring_t, 665
pSvcClass, 666
reason, 666
pack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 1942
pack_voice_SLQSVoiceGetCallForwardingStatus_t, 666
pSvcClass, 667
Reason, 667
pack_voice_SLQSVoiceGetCallInfo
voice.h, 1943
pack_voice_SLQSVoiceGetCallInfo_t, 667
callID, 667
pack_voice_SLQSVoiceGetCallWaiting
voice.h, 1943
pack_voice_SLQSVoiceGetCallWaiting_t, 667
pSvcClass, 668
pack_voice_SLQSVoiceGetConfig
voice.h, 1945
pack_voice_SLQSVoiceGetConfig_t, 668
pAMRStatus, 669
pAirTimer, 669
pAutoAnswer, 669
pNamID, 669
pPrefVoicePrivacy, 669
pPrefVoiceSO, 669
pRoamTimer, 669
pTTYMode, 669
pVoiceDomainPref, 669
pack_voice_SLQSVoiceIndicationRegister
voice.h, 1945
pack_voice_SLQSVoiceIndicationRegister_t, 670
pRegDTMFEvents, 670
pRegVoicePrivacyEvents, 670
pSuppsNotifEvents, 670
pack_voice_SLQSVoiceManageCalls
voice.h, 1946
pack_voice_SLQSVoiceManageCalls_t, 670
pCallID, 671
SUPSType, 671
pack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 1946
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 671
pack_voice_SLQSVoiceSendFlash
voice.h, 1947
pack_voice_SLQSVoiceSendFlash_t, 672
pCallID, 672
pFlashPayLd, 672
pFlashType, 672
pack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 1947
pack_voice_SLQSVoiceSetCallBarringPassword_t, 672
newPasswd, 673
newPasswdAgain, 673
oldPasswd, 673
Reason, 673
pack_voice_SLQSVoiceSetConfig
voice.h, 1947
pack_voice_SLQSVoiceSetConfig_t, 673
pAirTimerConfig, 675
pAutoAnswer, 675
pPrefVoiceDomain, 675
pPrefVoiceSO, 675
pRoamTimerConfig, 675
pTTYMode, 675
pack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 1948
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 675
privacyPref, 675
pack_voice_SLQSVoiceSetSUPSService
voice.h, 1948
pack_voice_SLQSVoiceSetSUPSService_t, 675
pCallBarringPasswd, 677
pCallForwardingNumber, 677
pServiceClass, 677
pTimerVal, 677
reason, 677
voiceSvc, 677
pack_voice_SLQSVoiceStartContDTMF

voice.h, 1948
pack_voice_SLQSVoiceStartContDTMF_t, 677
DTMFdigit, 678
pCallID, 678
pack_voice_SLQSVoiceStopContDTMF
voice.h, 1949
pack_voice_SLQSVoiceStopContDTMF_t, 678
callID, 678
pack_wds_DHCPv4ClientLeaseChange
wds.h, 1966
pack_wds_DHCPv4ClientLeaseChange_t, 678
pEnableNotification, 679
pack_wds_GetAutoconnect
wds.h, 1966
pack_wds_GetByteTotals
wds.h, 1967
pack_wds_GetConnectionRate
wds.h, 1967
pack_wds_GetDataBearerTechnology
wds.h, 1967
pack_wds_GetDefaultProfile
wds.h, 1968
pack_wds_GetDefaultProfile_t, 679
profiletype, 679
pack_wds_GetDefaultProfileNum
wds.h, 1968
pack_wds_GetDefaultProfileNum_t, 679
family, 679
type, 679
pack_wds_GetDormancyState
wds.h, 1969
pack_wds_GetDormancyState_t, 679
pack_wds_GetLastMobileIPError
wds.h, 1969
pack_wds_GetLastMobileIPError_t, 679
pack_wds_GetMobileIP
wds.h, 1969
pack_wds_GetMobileIP_t, 680
pack_wds_GetMobileIPProfile
wds.h, 1970
pack_wds_GetMobileIPProfile_t, 680
index, 680
pack_wds_GetPacketStatistics
wds.h, 1970
pack_wds_GetPacketStatistics_t, 680
pStatMask, 680
pack_wds_GetPacketStatus
wds.h, 1971
pack_wds_GetPacketStatus_t, 680
statmask, 681
pack_wds_GetSessionDuration
wds.h, 1971
pack_wds_GetSessionDuration_t, 681
pack_wds_GetSessionState
wds.h, 1971
pack_wds_RMSetTransferStatistics
wds.h, 1972
pack_wds_RMSetTransferStatistics_t, 681
RmTrasnferStaticsReq, 681
pack_wds_SLQSCreateProfile
wds.h, 1975
pack_wds_SLQSCreateProfile_t, 686
pCurProfile, 686
pProfileId, 686
pProfileType, 686
pack_wds_SLQSDeleteProfile
wds.h, 1975
pack_wds_SLQSDeleteProfile_t, 686
profileIndex, 687
profileType, 687
pack_wds_SLQSGet3GPPConfigItem
wds.h, 1976
pack_wds_SLQSGetCurrDataSystemStat
wds.h, 1976
pack_wds_SLQSGetCurrDataSystemStat_t, 687
pack_wds_SLQSGetCurrentChannelRate
wds.h, 1976
pack_wds_SLQSGetDUNCallInfo
wds.h, 1977
pack_wds_SLQSGetDUNCallInfo_t, 687
Mask, 687
pReportChannelRate, 687
pReportConnStatus, 687
pReportDataBearerTech, 687
pReportDormStatus, 687
pTransferStatInd, 688
pack_wds_SLQSGetDataBearerTechnology
wds.h, 1977
pack_wds_SLQSGetDataBearerTechnology_t, 687
pack_wds_SLQSGetProfileSettings
wds.h, 1978
pack_wds_SLQSGetProfileSettings_t, 688
ProfileId, 688
ProfileType, 688
pack_wds_SLQSGetRuntimeSettings
wds.h, 1978
pack_wds_SLQSGetRuntimeSettings_t, 688
pReqSettings, 689
pack_wds_SLQSModifyProfile
wds.h, 1979
pack_wds_SLQSModifyProfile_t, 689
curProfile, 690
pProfileId, 690
pProfileType, 690
pack_wds_SLQSResetPacketStatics
wds.h, 1979
pack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 1981
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 693
pProfileId, 693
pack_wds_SLQSSGetLoopback
wds.h, 1981
pack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 1981
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 693
pHwConfig, 694

pProfileId, 694
pack_wds_SLQSSetLoopback
wds.h, 1982
pack_wds_SLQSSetLoopback_t, 694
loopbackMode, 694
loopbackMultiplier, 694
pack_wds_SLQSSet3GPPConfigItem
wds.h, 1979
pack_wds_SLQSSet3GPPConfigItem_t, 690
p3gppRelease, 691
pProfileList, 691
pack_wds_SLQSSetIPFamilyPreference
wds.h, 1980
pack_wds_SLQSSetIPFamilyPreference_t, 692
IPFamilyPreference, 692
pack_wds_SLQSSetWdsEventCallback
wds.h, 1980
pack_wds_SLQSSetWdsEventCallback_t, 692
currentDataBearer, 692
dataBearer, 692
dataBearerTechExt, 693
dataSystemStatus, 693
dormancyStatus, 693
interval, 693
mobileIP, 693
transferStats, 693
pack_wds_SLQSStartDataSession
wds.h, 1982
pack_wds_SLQSStartDataSession_t, 694
pAuth, 695
pPass, 695
pTech, 695
pUser, 695
pprofileid3gpp, 695
pprofileid3gpp2, 695
pack_wds_SLQSStopDataSession
wds.h, 1983
pack_wds_SLQSStopDataSession_t, 695
psid, 696
pack_wds_SLQSWdsGoActive
wds.h, 1983
pack_wds_SLQSWdsGoDormant
wds.h, 1983
pack_wds_SLQSWdsSetEventReport
wds.h, 1984
pack_wds_SLQSWdsSetEventReport_t, 696
pCurrChannelRateInd, 697
pCurrDataBearerTechInd, 697
pCurrPrefDataSysInd, 697
pDataBearerTechInd, 697
pDataCallStatusChangeInd, 697
pDataSystemStatusChangeInd, 697
pDormancyStatusInd, 697
pMIPStatusInd, 697
pTransferStatInd, 697
pack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 1984
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 698
contextId, 698
contextType, 698
pack_wds_SetAutoconnect
wds.h, 1972
pack_wds_SetAutoconnect_t, 681
acroamsetting, 681
acsetting, 681
pack_wds_SetDefaultProfile
wds.h, 1973
pack_wds_SetDefaultProfile_t, 681
authentication, 682
ipAddress, 682
pApnname, 682
pName, 682
pPassword, 682
pUsername, 682
pdpType, 682
primaryDNS, 682
profileType, 682
secondaryDNS, 682
pack_wds_SetDefaultProfileNum
wds.h, 1973
pack_wds_SetDefaultProfileNum_t, 682
family, 683
index, 683
type, 683
pack_wds_SetMobileIP
wds.h, 1973
pack_wds_SetMobileIP_t, 683
mode, 683
pack_wds_SetMobileIPParameters
wds.h, 1974
pack_wds_SetMobileIPParameters_t, 683
pHA2002bis, 684
pHAAuthenticator, 684
pMode, 684
pReRegPeriod, 684
pReRegTraffic, 684
pRetryInterval, 684
pRetryLimit, 684
pSPC, 684
pack_wds_SetMobileIPProfile
wds.h, 1974
pack_wds_SetMobileIPProfile_t, 685
index, 685
pAAASPI, 685
pAddress, 685
pEnabled, 685
pHASPI, 685
pMNAAA, 685
pMNHA, 685
pNAI, 685
pPrimaryHA, 685
pRevTunneling, 685
pSecondaryHA, 686
spc, 686
PackCreateProfileOut, 698
ExtErrorCode, 698

ProfileIndex, 698
ProfileType, 698
package_name
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
unpack_omaDmFotaTlv_t, 1087
PackageID
_SwiAvmsEventInd, 100
packageSize
omaDmFotaTlvExt, 575
packageid_str
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
packetSrvStatus
qaGobiApiCbk.h, 1466
packetZone
CDMASysInfo, 174
nas_CDMASysInfo, 418
packetZoneValid
CDMASysInfo, 174
nas_CDMASysInfo, 418
packgetDyingGaspCfg, 698
pDestSMSContent, 699
pDestSMSNum, 699
packgetDyingGaspStatistics, 699
pSMSAttemptedFlag, 699
pTimeStamp, 699
path
fileInfo, 242
uim_fileInfo, 947
pathLen
fileInfo, 242
uim_fileInfo, 947
pbAuto
unpack_pds_GetPortAutomaticTracking_t, 1089
pbIMSRegistered
imsaRegStatusInfo, 312
pbPlatform
unpack_dms_GetOfflineReason_t, 1001
pci
cellParams, 176
ltePCI, 392
nas_cellParams, 420
nas_ltePCI, 456
nas_PhyCaAggPcellInfo, 476
nas_PhyCaAggScellIndType, 477
nas_PhyCaAggScellInfo, 480
nas_umtsLTENbrCell, 513
NASPhyCaAggPcellInfo, 542
NASPhyCaAggScellIndType, 544
NASPhyCaAggScellInfo, 545
PhyCaAggPcellInfo, 707
PhyCaAggScellIndType, 709
PhyCaAggScellInfo, 712
umtsLTENbrCell, 983
pcsDigit
PlmnID, 714
pcsfFQDNAddress
PCSCFFQDNAddressList, 701
wds_PCSCFFQDNAddressList, 1302
pdpType
pack_wds_SetDefaultProfile_t, 682
pdptype
unpack_wds_GetDefaultProfile_t, 1166
pds.h, 1441
pack_pds_GetPDSDefaults, 1441
pack_pds_GetPDSState, 1442
pack_pds_GetPortAutomaticTracking, 1442
pack_pds_SetPDSDefaults, 1442
pack_pds_SetPDSState, 1443
pack_pds_SetPortAutomaticTracking, 1443
unpack_pds_GetPDSDefaults, 1443
unpack_pds_GetPDSState, 1444
unpack_pds_GetPortAutomaticTracking, 1444
unpack_pds_SetPDSDefaults, 1444
unpack_pds_SetPDSState, 1445
unpack_pds_SetPortAutomaticTracking, 1445
peakRate
tokenBucket, 926
unpack_qos_tokenBucket_t, 1108
peakThroughputClass
GPRSQoS, 283
GPRSRequestedQoS, 284
LibPackGPRSRequestedQoS, 329
wds_GPRSQoS, 1300
peerNumberInfo, 704
callID, 705
numLen, 706
numPI, 706
numPlan, 706
numSI, 706
numType, 706
number, 706
PerformNetworkScan
qaGobiApiNas.h, 1647
PersistentTechPref
unpack_nas_GetNetworkPreference_t, 1049
persoFeature
appStats, 124
appStatus, 127
uim_appStatus, 936
persoRetries
appStats, 124
appStatus, 127
uim_appStatus, 936
persoState
appStats, 124
appStatus, 127
uim_appStatus, 936
persoUnblockRetries
appStats, 124
appStatus, 127
uim_appStatus, 936
personalizationStatus, 706
feature, 707
numFeatures, 707

unblockLeft, 707
verifyLeft, 707
pfailureCause
unpack_voice_SLQSOriginateUSSD_t, 1137
USSResp, 1192
phase
nas_rxInfo, 493
rxInfo, 788
PhyCaAggPcellInfo, 707
dl_bw_value, 707
freq, 707
iLTEbandValue, 707
NasGetLTECphyCaInfo, 525
pci, 707
TlvPresent, 708
PhyCaAggScellDlBw, 708
dl_bw_value, 708
NasGetLTECphyCaInfo, 525
TlvPresent, 708
PhyCaAggScellIndType, 709
freq, 709
NasGetLTECphyCaInfo, 525
pci, 709
scell_state, 709
TlvPresent, 709
PhyCaAggScellIndex, 708
NasGetLTECphyCaInfo, 525
scell_idx, 708
TlvPresent, 709
PhyCaAggScellInfo, 709
dl_bw_value, 712
freq, 712
iLTEbandValue, 712
NasGetLTECphyCaInfo, 526
pci, 712
scell_state, 712
TlvPresent, 712
PhysicalLayer
nas_protocolSubtypeElement, 485
protocolSubtypeElement, 733
PilotEnergy
nas_NetworkStatEVDO, 472
NetworkStatEVDO, 565
PilotPN
nas_PilotSetParams, 481
PilotSetParams, 713
PilotSetData, 712
NumPilots, 712
pPilotSetInfo, 712
PilotSetParams, 713
PilotPN, 713
PilotStrength, 713
PilotType, 713
PilotStrength
nas_PilotSetParams, 481
PilotSetParams, 713
PilotType
nas_PilotSetParams, 481
PilotSetParams, 713
pin1Len
encryptedPIN1, 235
uim_encryptedPIN1, 943
pin1Retries
appStats, 124
appStatus, 127
uim_appStatus, 936
pin1State
appStats, 124
appStatus, 127
uim_appStatus, 936
pin1Val
encryptedPIN1, 235
uim_encryptedPIN1, 943
pin2Retries
appStats, 124
appStatus, 127
uim_appStatus, 936
pin2State
appStats, 124
appStatus, 127
uim_appStatus, 936
pinID
changeUIMPIN, 177
setPINProtection, 823
uim_changeUIMPIN, 941
uim_setPINProtection, 954
uim_unblockUIMPIN, 957
uim_verifyUIMPIN, 958
unblockUIMPIN, 990
verifyUIMPIN, 1194
pinLen
changeUIMPIN, 177
uim_changeUIMPIN, 941
uim_verifyUIMPIN, 958
verifyUIMPIN, 1194
pinLength
setPINProtection, 823
uim_setPINProtection, 954
pinOperation
setPINProtection, 823
uim_setPINProtection, 954
pinProtection
pack_uim_SetPinProtection_t, 651
pack_uim_UnblockPin_t, 658
UIMSetPinProtectionReq, 975
pinVal
changeUIMPIN, 177
uim_changeUIMPIN, 941
uim_verifyUIMPIN, 958
verifyUIMPIN, 1194
pinValue
setPINProtection, 823
uim_setPINProtection, 954
PkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 1861
PkQmiNasPerformNetworkScan

qaNasPerformNetworkScan.h, 1862
PkgDescLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
PkgDescription
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
PkgName
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
PkgNameLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
pkgver
CurrentImgList, 197
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1015
PktErrRate
unpack_qos_swiQosFlow_t, 1108
pktErrRate, 713
exponent, 713
multiplier, 713
PktStatElmntsV4
WdsPktStatisticsResp, 1320
PktStatElmntsV6
WdsPktStatisticsResp, 1320
plasmaIDString
pack_dms_SLQSSwiSetHostDevInfo_t, 583
unpack_dms_SLQSSwiGetHostDevInfo_t, 1017
plmn
GERANInfo, 250
LTEInfoIntrafreq, 389
nas_GERANInfo, 434
nas_LTEInfoIntrafreq, 454
nas_UMTSInfo, 511
UMTSInfo, 981
PlmnID, 714
mcc, 714
mnc, 714
pcsDigit, 714
TlvPresent, 714
polarityIncluded
lineCtrlInfo, 358
Port, 717
port, 717
range, 717
port
Port, 717
unpack_qos_Port_t, 1093
Position Determination Service (PDS), 43
positionSrc
pack_loc_SLQSLOCInjectPosition_t, 600
pprofileid3gpp
pack_wds_SLQSStartDataSession_t, 695
pprofileid3gpp2
pack_wds_SLQSStartDataSession_t, 695
prDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
prDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
prPCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
prPCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
Precedence
unpack_qos_swiQosFilter_t, 1104
precedenceClass
GPRSQoS, 283
GPRSRequestedQoS, 284
LibPackGPRSRequestedQoS, 329
wds_GPRSQoS, 1300
precisionDilution
qaGobiApiCbk.h, 1467
precisionDilution_s, 717
HDOP, 718
PDOP, 718
VDOP, 718
PrefImageList, 718
listEntries, 718
listSize, 718
prefNetwork
unpack_wds_SLQSGetCurrDataSystemStat_t, 1175
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
prefVoiceSO, 718
evrcCapability, 720
homeOrigVoiceSO, 720
homePageVoiceSO, 720
namID, 720
roamOrigVoiceSO, 720
Preferred
nas_QmiNas3GppNetworkInfo, 486
SlqsNas3GppNetworkInfo, 842
prefixLen
IPv6Addr, 326
unpack_qos_IPv6Addr_t, 1091
presentationInd
ECTNum, 234
remotePartyNum, 779
voice_remotePartyNum, 1228
priChA
CDMAChannel, 162
nas_CDMAChannel, 413
priChB
CDMAChannel, 162
nas_CDMAChannel, 413
priSize
unpack_dms_GetFirmwareRevisions_t, 998
pridns
unpack_wds_GetDefaultProfile_t, 1166
pridnsv6
unpack_wds_GetDefaultProfile_t, 1166

primaryDNS
pack_wds_SetDefaultProfile_t, 682
PrimaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1179
PrimaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1179
primaryHA
unpack_wds_GetMobileIPProfile_t, 1168
privacyPref
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 675
voiceSetPrefPrivacy, 1274
priver
CurrentImgList, 197
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1015
priversion_str
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
Profile
GetAudioPathConfigReq, 253
GetAudioProfileResp, 256
GetAudioVolTLBConfigReq, 257
GetM2MAudioProfileResp, 273
GetM2MAudioVolumeReq, 274
GetM2MAVMuteReq, 274
GetM2MSpkrGainReq, 276
SetAudioPathConfigReq, 809
SetAudioProfileReq, 810
SetAudioVolTLBConfigReq, 811
SetM2MAudioAVCFGReq, 818
SetM2MAudioProfileReq, 820
SetM2MAudioVolumeReq, 821
SetM2MAVMuteReq, 822
SetM2MSpkrGainReq, 822
Profile3GPP, 720
pAPNClass, 725
pAPNDisabledFlag, 725
pAPNName, 725
pAPNnameSize, 725
pAddrAllocPref, 725
pAuthenticationPref, 725
pGPRSMinimumQoS, 725
pGPRSRequestedQos, 725
pIPv4AddrPref, 725
pIPv6AddPref, 725
pImCnFlag, 725
pPDNInactivTimeout, 725
pPDPtype, 725
pPassword, 725
pPasswordSize, 725
pPcscfAddrUsingDhcp, 725
pPcscfAddrUsingPCO, 725
pPdpAccessConFlag, 725
pPdpContext, 725
pPdpDataCompType, 725
pPdpHdrCompType, 725
pPriDNSIPv4AddPref, 725
pPriDNSIPv6addpref, 725
pPrimaryID, 725
pProfilename, 726
pProfilenameSize, 726
pQosClassID, 726
pSecDNSIPv4AddPref, 726
pSecDNSIPv6addpref, 726
pSecondaryFlag, 726
pTFTID1Params, 726
pTFTID2Params, 726
pUMTSMinQoS, 726
pUMTSMinQosSigInd, 726
pUMTSReqQoS, 726
pUMTSReqQoSSigInd, 726
pUsername, 726
pUsernameSize, 726
Profile3GPP2, 726
pAPNClass3GPP2, 730
pAPNEnabled3GPP2, 730
pAllowLinger, 730
pApnString, 730
pApnStringSize, 730
pAppPriority, 730
pAppType, 730
pAuthPassword, 730
pAuthPasswordSize, 730
pAuthProtocol, 730
pAuthRetryCount, 731
pAuthTimeout, 731
pDataMode, 731
pDataRate, 731
pIpcpAckTimeout, 731
pIpcpCreqRetryCount, 731
pIsPcscfAddressNedded, 731
pLcpAckTimeout, 731
pLcpCreqRetryCount, 731
pNegoDnsSrvrPref, 731
pPDNInactivTimeout3GPP2, 731
pPdnType, 731
pPppSessCloseTimer1x, 731
pPppSessCloseTimerDO, 731
pPriV6DnsAddress, 731
pPrimaryV4DnsAddress, 731
pRATType, 731
pSecV6DnsAddress, 731
pSecondaryV4DnsAddress, 731
pUserId, 731
pUserIdSize, 731
ProfileID
_GetProfileSettingIn, 64
unpack_wds_SLQSGetRuntimeSettings_t, 1179
ProfileId
pack_wds_SLQSGetProfileSettings_t, 688
profileId
wds_DHCPProfileIdTlv, 1296
wds_DHCPv4ProfileId, 1298
WdsDHCPv4ProfileId, 1315
wdsDhcpv4ProfileId, 1316
ProfileId3GPP2

unpack_qos_swiQosFlow_t, 1108
ProfileIdTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1163
ProfileIdentifier, 731
profileIndex, 732
profileType, 732
ProfileIndex
PackCreateProfileOut, 698
profileIndex
pack_wds_SLQSDeleteProfile_t, 687
ProfileIdentifier, 732
SLQSDeleteProfileParams, 840
wds_ProfileIdentifier, 1303
profileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1174
ProfileName
unpack_wds_SLQSGetRuntimeSettings_t, 1179
ProfileType
_GetProfileSettingIn, 64
pack_wds_SLQSGetProfileSettings_t, 688
PackCreateProfileOut, 698
unpack_wds_SLQSGetProfileSettings_t, 1178
profileType
pack_wds_SetDefaultProfile_t, 682
pack_wds_SLQSDeleteProfile_t, 687
ProfileIdentifier, 732
SLQSDeleteProfileParams, 840
wds_DHCPProfileIdTlv, 1296
wds_DHCPv4ProfileId, 1298
wds_ProfileIdentifier, 1303
WdsDHCPv4ProfileId, 1315
wdsDhcpv4ProfileId, 1316
profiletype
pack_wds_GetDefaultProfile_t, 679
Protocol
unpack_nas_GetCDMANetworkParameters_t, 1046
protocolSubtypeElement, 732
AccessMac, 733
AuthProt, 733
ControlMac, 733
EncryptProt, 733
ForwardMac, 733
IdleState, 733
KeyExchange, 733
MultDisc, 733
PhysicalLayer, 733
ReverseMac, 733
SecProt, 733
VirtStream, 733
ProvisionStatus
CLIPResp, 179
CLIRResp, 179
CNAPResp, 181
COLPResp, 182
COLRResp, 183
voice_CLIPResp, 1213
voice_CLIRResp, 1214
voice_CNAPResp, 1214
voice_COLPResp, 1215
voice_COLRResp, 1216
psAttachState
nas_servSystem, 498
NASServingSystemInfo, 551
ServingSystemInfo, 803
servSystem, 805
psBarStatus
CallBarringSysInfo, 143
callBarStatus, 144
nas_CallBarringSysInfo, 411
nas_callBarStatus, 412
psState
CommInfo, 184
nas_CommInfo, 422
psc
nas_UMTSInfo, 511
nas_wcdmaCellInfo, 515
nas_WCDMASysInfo, 520
UMTSInfo, 982
wcdmaCellInfo, 1282
WCDMASysInfo, 1292
pscValid
nas_WCDMASysInfo, 520
WCDMASysInfo, 1292
pscsfIPv4Addr
PCSCFIPv4ServerAddressList, 701
wds_PCSCFIPv4ServerAddressList, 1303
psetting
unpack_wds_GetAutoconnect_t, 1163
psid
pack_wds_SLQSStopDataSession_t, 696
unpack_wds_SLQSStartDataSession_t, 1185
puk1Retries
appStats, 124
appStatus, 127
uim_appStatus, 936
puk2Retries
appStats, 124
appStatus, 127
uim_appStatus, 936
pukLen
uim_unblockUIMPIN, 957
unblockUIMPIN, 990
pukVal
uim_unblockUIMPIN, 958
unblockUIMPIN, 990
pukValue
pack_dms_UIMUnblockPIN_t, 588
pv4sessionId
WdsIpAddressInfoReq, 1317
pv6sessionId
WdsIpAddressInfoReq, 1317
pwrDenialTime
lineCtrlInfo, 358

QMI_SAR_RF_STATE_1
qaGobiApiSar.h, 1693

QMI_SAR_RF_STATE_2
qaGobiApiSar.h, 1693
QMI_SAR_RF_STATE_3
qaGobiApiSar.h, 1693
QMI_SAR_RF_STATE_4
qaGobiApiSar.h, 1693
QMI_SAR_RF_STATE_5
qaGobiApiSar.h, 1693
QMI_SAR_RF_STATE_6
qaGobiApiSar.h, 1693
QMI_SAR_RF_STATE_7
qaGobiApiSar.h, 1693
QMI_SAR_RF_STATE_8
qaGobiApiSar.h, 1693
QMI_SAR_RF_STATE_DEFAULT
qaGobiApiSar.h, 1693
QMI_WDS_CURRENT_CALL_DB_MASK
qaGobiApiWds.h, 1825
QMI_WDS_LAST_CALL_DB_MASK
qaGobiApiWds.h, 1825
QCI
LibPackQosClassID, 352
QosClassID, 769
QCWWAN2kConnect
qaGobiApiDcs.h, 1539
QCWWAN2kEnumerateDevices
qaGobiApiDcs.h, 1540
QCWWAN2kGetConnectedDeviceID
qaGobiApiDcs.h, 1540
QCWWANConnect
qaGobiApiDcs.h, 1541
QCWWANDisconnect
qaGobiApiDcs.h, 1541
QCWWANEnumerateDevices
qaGobiApiDcs.h, 1541
QFlowState
unpack_qos_QosFlowInfo_t, 1094
QLIC
DeviceConfigDetail, 220
nas_DeviceConfigDetail, 429
qaCbkCatEventReportInd.h
eTLV_CBK_ALPHA_IDENTIFIER, 1446
eTLV_CBK_DISPLAY_TEXT, 1446
eTLV_CBK_END_PROACTIVE_SESSION, 1447
eTLV_CBK_GET_IN_KEY, 1446
eTLV_CBK_GET_INPUT, 1446
eTLV_CBK_LANGUAGE_NOTIFICATION, 1447
eTLV_CBK_REFRESH, 1447
eTLV_CBK_SELECT_ITEM, 1446
eTLV_CBK_SETUP_EVENT_LIST, 1447
eTLV_CBK_SETUP_IDLE_MODE_TEXT, 1447
eTLV_CBK_SETUP_MENU, 1446
eTLV_END_PROACTIVE_SESSION_LENGTH, 1447
eTLV_REFRESH_LENGTH, 1447
eTLV_SETUP_EVENT_LIST_LENGTH, 1447
qaCbkSwiOmaDmEventReportInd.h
eTLV_IND_OMA_DM_CONFIG, 1448
eTLV_IND_OMA_DM_FOTA, 1448
eTLV_IND_OMA_DM_NOT, 1448
qaGobiApiCbk.h
DEVICE_STATE_BOOT, 1501
DEVICE_STATE_DISCONNECTED, 1501
DEVICE_STATE_READY, 1501
DEVICE_STATE_SERIAL_ONLY, 1501
eQA_QMI_SVC_NA, 1501
eQA_QMI_SVC_NAS, 1501
eQA_QMI_SVC_WDS, 1501
SMS_EVENT_ETWS, 1501
SMS_EVENT_ETWS_PLMN, 1501
SMS_EVENT_MESSAGE_MODE, 1501
SMS_EVENT_MT_MESSAGE, 1501
SMS_EVENT_SMS_ON_IMS, 1501
SMS_EVENT_SMSC_ADDRESS, 1501
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE, 1501
SYS_SRV_DOMAIN_CAMPED, 1501
SYS_SRV_DOMAIN_CS_ONLY, 1501
SYS_SRV_DOMAIN_CS_PS, 1501
SYS_SRV_DOMAIN_NO_SRV, 1501
SYS_SRV_DOMAIN_PS_ONLY, 1501
qaGobiApiFms.h
eGOBI_DEV_SERIES_9X07_9X50, 1589
eGOBI_DEV_SERIES_9X15, 1589
eGOBI_DEV_SERIES_9X30, 1589
eGOBI_DEV_SERIES_G3K, 1589
eGOBI_DEV_SERIES_NON_GOBI, 1589
eGOBI_DEV_SERIES_SIERRA_GOBI, 1589
eGOBI_DEV_SERIES_UNKNOWN, 1589
eGOBI_IMG_CAR_3, 1590
eGOBI_IMG_CAR_AERIS, 1590
eGOBI_IMG_CAR_ALLTEL, 1589
eGOBI_IMG_CAR_AMX_TELCEL, 1590
eGOBI_IMG_CAR_ATT, 1589
eGOBI_IMG_CAR_BELL, 1589
eGOBI_IMG_CAR_BHARTI, 1590
eGOBI_IMG_CAR_BRASIL_VIVO, 1590
eGOBI_IMG_CAR_CHINA_MOBILE, 1590
eGOBI_IMG_CAR_CHINA_TELECOM, 1589
eGOBI_IMG_CAR_CHINA_UNICOM, 1589
eGOBI_IMG_CAR_EMOBILE, 1590
eGOBI_IMG_CAR_FACTORY, 1589
eGOBI_IMG_CAR_GENERIC, 1589
eGOBI_IMG_CAR_GENERIC_CDMA, 1589
eGOBI_IMG_CAR_IUSACELL, 1589
eGOBI_IMG_CAR_KDDI, 1589
eGOBI_IMG_CAR_KT_FREETEL, 1590
eGOBI_IMG_CAR_LEAP, 1589
eGOBI_IMG_CAR_METROPCS, 1589
eGOBI_IMG_CAR_NETCOM, 1590
eGOBI_IMG_CAR_NORF, 1589
eGOBI_IMG_CAR_NTT_DOCOMO, 1590
eGOBI_IMG_CAR_O2, 1590
eGOBI_IMG_CAR_OMH, 1589
eGOBI_IMG_CAR_ORANGE, 1590
eGOBI_IMG_CAR_RELIANCE1, 1589

eGOBI_IMG_CAR_RELIANCE2, 1590
eGOBI_IMG_CAR_ROGERS, 1590
eGOBI_IMG_CAR_SFR, 1590
eGOBI_IMG_CAR_SINGTEL_OPTUS, 1590
eGOBI_IMG_CAR_SK_TELCOM1, 1589
eGOBI_IMG_CAR_SK_TELCOM2, 1590
eGOBI_IMG_CAR_SOFTBANK, 1590
eGOBI_IMG_CAR_SPRINT, 1589
eGOBI_IMG_CAR_SWISSCOM, 1590
eGOBI_IMG_CAR_TATA, 1589
eGOBI_IMG_CAR_TELCOM_ITALIA, 1590
eGOBI_IMG_CAR_TELCOM_NZ, 1589
eGOBI_IMG_CAR_TELEFONICA, 1590
eGOBI_IMG_CAR_TELENOR, 1590
eGOBI_IMG_CAR_TELIASONERA, 1590
eGOBI_IMG_CAR_TELSTRA1, 1589
eGOBI_IMG_CAR_TELSTRA2, 1590
eGOBI_IMG_CAR_TELUS, 1589
eGOBI_IMG_CAR_TMOBILE, 1589
eGOBI_IMG_CAR_US, 1589
eGOBI_IMG_CAR_VERIZON, 1589
eGOBI_IMG_CAR_VODAFONE, 1589
eGOBI_IMG_GPS_ASSISTED, 1590
eGOBI_IMG_GPS_NO_XTRA, 1590
eGOBI_IMG_GPS_NONE, 1590
eGOBI_IMG_GPS_STAND_ALONE, 1590
eGOBI_IMG_REG_ASIA, 1590
eGOBI_IMG_REG_AUS, 1590
eGOBI_IMG_REG_EU, 1590
eGOBI_IMG_REG_GLOBAL, 1590
eGOBI_IMG_REG_LA, 1590
eGOBI_IMG_REG_NA, 1590
eGOBI_IMG_TECH_CDMA, 1591
eGOBI_IMG_TECH_UMTS, 1591
eGobi_DEV_SERIES_MC83, 1589
qaGobiApiNas.h
eNAS_LTE_CPHY_CA_BW_NRB_100, 1637
eNAS_LTE_CPHY_CA_BW_NRB_15, 1637
eNAS_LTE_CPHY_CA_BW_NRB_25, 1637
eNAS_LTE_CPHY_CA_BW_NRB_50, 1637
eNAS_LTE_CPHY_CA_BW_NRB_6, 1637
eNAS_LTE_CPHY_CA_BW_NRB_75, 1637
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1637
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1637
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1637
eNAS_RADIO_IF_GSM, 1636
eNAS_RADIO_IF_LTE, 1636
eNAS_RADIO_IF_TDSCDMA, 1636
eNAS_RADIO_IF_UMTS, 1636
eSYS_SRV_DOMAIN_CAMPED, 1636
eSYS_SRV_DOMAIN_CS_ONLY, 1636
eSYS_SRV_DOMAIN_CS_PS, 1636
eSYS_SRV_DOMAIN_NO_SRV, 1636
eSYS_SRV_DOMAIN_PS_ONLY, 1636
eSYS_SRV_DOMAIN_UNKNOWN, 1636
qaGobiApiPds.h
eSetServiceAutomaticTrackingDisable, 1670
eSetServiceAutomaticTrackingEnable, 1670
qaGobiApiSar.h
QMI_SAR_RF_STATE_1, 1693
QMI_SAR_RF_STATE_2, 1693
QMI_SAR_RF_STATE_3, 1693
QMI_SAR_RF_STATE_4, 1693
QMI_SAR_RF_STATE_5, 1693
QMI_SAR_RF_STATE_6, 1693
QMI_SAR_RF_STATE_7, 1693
QMI_SAR_RF_STATE_8, 1693
QMI_SAR_RF_STATE_DEFAULT, 1693
qaGobiApiVoice.h
VOICE_SUPS_SRV_CLASS_DATA, 1801
VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC, 1802
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC, 1802
VOICE_SUPS_SRV_CLASS_FAX, 1801
VOICE_SUPS_SRV_CLASS_NONE, 1801
VOICE_SUPS_SRV_CLASS_PACKETACCESS, 1802
VOICE_SUPS_SRV_CLASS_PADACCESS, 1802
VOICE_SUPS_SRV_CLASS_SMS, 1802
VOICE_SUPS_SRV_CLASS_VOICE, 1801
qaGobiApiWds.h
QMI_WDS_CURRENT_CALL_DB_MASK, 1825
QMI_WDS_LAST_CALL_DB_MASK, 1825
qaNasGetRFBandInfo.h
eTLV_RF_BAND_INFO, 1861
qaNasPerformNetworkScan.h
eTLV_3GPP_NETWORK_INFO, 1862
qaCbkCatEventReportInd.h, 1445
UpkQmiCbkCatEventReportInd, 1447
qaCbkSwiOmaDmEventReportInd.h, 1447
UpkQmiCbkSwiOmaDmEventReportInd, 1448
UpkQmiCbkSwiOmaDmEventReportIndExt, 1448
qaGobiApiAudio.h, 1448
SLQSGetAudioPathConfig, 1449
SLQSGetAudioProfile, 1449
SLQSGetAudioVolTLBConfig, 1450
SLQSSetAudioPathConfig, 1450
SLQSSetAudioProfile, 1451
SLQSSetAudioVolTLBConfig, 1451
qaGobiApiCat.h, 1452
CATSendEnvelopeCommand, 1452
CATSendTerminalResponse, 1452
qaGobiApiCbk.h, 1453
accelAcceptReady, 1463
accelTempAcceptReady, 1463
CBK_ENABLE_EVENT, 1461
CBK_NOCHANGE, 1461
DEREGISTER_EVENT, 1461
DEREGISTER_SRV, 1461
device_state_enum, 1501
eDevState, 1464
eQaQMIService, 1501

eSMSEventType, 1464
EVENT_MASK_CARD, 1461
FIRST_INSTANCE, 1461
gpsTime, 1464
gyroAcceptReady, 1464
gyroTempAcceptReady, 1465
INVALID_INSTACNE, 1461
IPV4, 1462
IPV4V6, 1462
IPV6, 1462
iSLQSSetDUNCallInfoCallback, 1502
iSLQSSetSignalStrengthsCallback, 1502
iSLQSSetWdsFirstInstEventCallback, 1502
iSLQSSetWdsSecondInstEventCallback, 1502
iSLQSSetWdsThirdInstEventCallback, 1502
iSLQSSetWdsXferStatsFirstInstCallback, 1502
iSLQSSetWdsXferStatsSecondInstCallback, 1502
iSetCATEventCallback, 1502
iSetSignalStrengthCallback, 1502
LteNasReleaseInfo, 1465
MAX_NO_OF_CALLS, 1462
MAX_NO_OF_FILES, 1462
MAX_NO_OF_SLOTS, 1462
MAX_PATH_LENGTH, 1462
MAXUSSDLENGTH, 1462
modemTempNotification, 1466
NAS_SRV, 1462
NUM_OF_SET, 1462
PDS_SRV, 1462
packetSrvStatus, 1466
precisionDilution, 1467
REGISTER_EVENT, 1462
REGISTER_SRV, 1462
ResetInfoNotification, 1468
SECOND_INSTANCE, 1462
SLQSNasNetworkRejectCallback, 1516
SLQSNasNetworkTimeCallBack, 1516
SLQSNasSigInfo2CallBack, 1517
SLQSNasSigInfoCallBack, 1517
SLQSNasSwiOTAMessageCallback, 1518
SLQSNasSysInfoCallBack, 1518
SLQSNasTimerCallback, 1519
SLQSSetBandPreferenceCbk, 1519
SLQSSetDHCPv4ClientLeaseStatusCallback, 1520
SLQSSetDUNCallInfoCallback, 1520
SLQSSetDataSystemStatusCallback, 1519
SLQSSetIMSAPdpStatusCallback, 1521
SLQSSetIMSARatStatusCallback, 1521
SLQSSetIMSARegStatusCallback, 1521
SLQSSetIMSASvcStatusCallback, 1522
SLQSSetIMSSMSConfigCallback, 1522
SLQSSetIMSUserConfigCallback, 1523
SLQSSetIMSVoIPConfigCallback, 1523
SLQSSetLocInjectPositionCallback, 1523
SLQSSetLocInjectUTCTimeCallback, 1524
SLQSSetModemTempCallback, 1524
SLQSSetPacketSrvStatusCallback, 1524
SLQSSetQosEventCallback, 1525
SLQSSetQosNWStatusCallback, 1525
SLQSSetQosPriEventCallback, 1526
SLQSSetQosStatusCallback, 1526
SLQSSetRegMgrConfigCallback, 1526
SLQSSetSDKTerminatedCallback, 1527
SLQSSetSIPConfigCallback, 1529
SLQSSetSMSEventCallback, 1529
SLQSSetServingSystemCallback, 1527
SLQSSetSessionStateCallback, 1528
SLQSSetSignalStrengthsCallback, 1528
SLQSSetSwiGetResetInfoCallback, 1529
SLQSSetSwiHDRPersCallback, 1530
SLQSSetSysSelectionPrefCallBack, 1530
SLQSSetTransLayerInfoCallback, 1530
SLQSSetTransNWRegInfoCallback, 1531
SLQSSetWdsEventCallback, 1531
SLQSSetWdsTransferStatisticCallback, 1532
SLQSSwiAVMSEventReportCallback, 1533
SLQSTmdMitigationLvlRptCallback, 1533
SLQSUIMSetRefreshCallBack, 1533
SLQSUIMSetStatusChangeCallBack, 1533
SLQSVoiceInfoRecCallback, 1534
SLQSVoiceSetAllCallStatusCallBack, 1534
SLQSVoiceSetDTMFEventCallBack, 1535
SLQSVoiceSetOTASPStatusCallBack, 1535
SLQSVoiceSetPrivacyChangeCallBack, 1535
SLQSVoiceSetSUPSCallBack, 1536
SLQSVoiceSetSUPSNotificationCallback, 1536
SLQSWmsAsyncRawSendCallBack, 1537
SLQSWmsMemoryFullCallBack, 1537
SLQSWmsMessageWaitingCallBack, 1538
SMSAsyncRawSend, 1469
SMSCAddressInfo, 1470
SMSEtwsMessageInfo, 1470
SMSEtwsPlmnInfo, 1471
SMSEventInfo, 1471
SMSEventType, 1501
SMSMTMessageInfo, 1472
SMSMessageModeInfo, 1472
SMSOnIMSInfo, 1472
SMSTransferRouteMTMessageInfo, 1472
sensorDataUsage, 1468
sessionInformation, 1469
sessionInformationExt, 1469
SetActivationStatusCallback, 1502
SetCATEventCallback, 1502
SetDataCapabilitiesCallback, 1503
SetDeviceStateChangeCbk, 1504
SetFwDldCompletionCbk, 1504
SetGPSCallback, 1505
SetLURejectCallback, 1508
SetLocBestAvailPosCallback, 1505
SetLocCradleMountCallback, 1505
SetLocDeleteAssistDataCallback, 1506
SetLocEngineStateCallback, 1506
SetLocEventPositionCallback, 1506
SetLocEventTimeSyncCallback, 1506

SetLocGnssSvInfoCallback, 1507
SetLocInjectSensorDataCallback, 1507
SetLocInjectTimeCallback, 1507
SetLocOpModeCallback, 1508
SetLocSensorStreamingCallback, 1508
SetLocSetExtPowerConfigCallback, 1508
SetMobileIPStatusCallback, 1509
SetNMEACallback, 1511
SetNasLTECphyCaIndCallback, 1509
SetNetChangeCbk, 1510
SetNewSMSCallback, 1510
SetOMADMStateCallback, 1511
SetPDSStateCallback, 1511
SetPowerCallback, 1512
SetRFInfoCallback, 1512
SetRMTransferStatisticsCallback, 1512
SetRankIndicatorCallback, 1512
SetRoamingIndicatorCallback, 1513
SetSLQSOMADMAlertCallback, 1514
SetSLQSOMADMAlertCallbackExt, 1514
SetSignalStrengthCallback, 1513
SetUSSDNoWaitIndicationCallback, 1515
SetUSSDNotificationCallback, 1515
SetUSSDReleaseCallback, 1516
SetUimSlotStatusChangeCallback, 1515
svUsedforFix, 1473
SwiAvmsEventInd, 1473
SwiOTAMsg, 1473
tFNASwiLTECphyCaInfo, 1474
tFNASwiOTAMsg, 1475
tFNActivationStatus, 1474
tFNAllCallStatus, 1474
tFNAsyncRawSend, 1475
tFNBandPreference, 1475
tFNBestAvailPos, 1476
tFNCATEvent, 1477
tFNCbkUimSlotStatusChangeInd, 1477
tFNDHCPv4ClientLeaseStatus, 1479
tFNDTMFEvent, 1479
tFNDUNCallInfo, 1479
tFNDataCapabilities, 1477
tFNDataSysStatus, 1478
tFNDelAssistData, 1478
tFNDeviceStateChange, 1478
tFNEventPosition, 1479
tFNFwDldCompletion, 1479
tFNGnssSvInfo, 1480
tFNHDRPersonaity, 1480
tFNImsRegMgrConfig, 1481
tFNImsSIPConfig, 1481
tFNImsSMSConfig, 1482
tFNImsUserConfig, 1482
tFNImsVoIPConfig, 1482
tFNImsaPdpStatus, 1480
tFNImsaRatStatus, 1481
tFNImsaRegStatus, 1481
tFNImsaSvcStatus, 1481
tFNInfoRec, 1482
tFNInjectPosition, 1482
tFNInjectSensorData, 1483
tFNInjectTimeStatus, 1483
tFNInjectUTCTime, 1483
tFNLUReject, 1483
tFNMemoryFull, 1485
tFNMessageWaiting, 1485
tFNMitiLvlRpt, 1485
tFNMobileIPStatus, 1485
tFNModemTempInfo, 1485
tFNNasTimer, 1486
tFNNet, 1486
tFNNetworkReject, 1486
tFNNetworkTime, 1486
tFNNewGPS, 1487
tFNNewNMEA, 1487
tFNNewRMTransferStatistics, 1487
tFNNewSMS, 1488
tFNOMADMState, 1488
tFNOTASPStatus, 1490
tFNOpMode, 1489
tFNPDSState, 1490
tFNPacketSrvState, 1490
tFNPower, 1491
tFNPrivacyChange, 1491
tFNQosNWStatus, 1491
tFNQosPriEvent, 1492
tFNQosStatus, 1492
tFNRFInfo, 1493
tFNRankIndicator, 1493
tFNResetInfo, 1493
tFNRoamingIndicator, 1494
tFNSDKTerminated, 1494
tFNSLQSOMADMAlert, 1496
tFNSLQSQOSEvent, 1496
tFNSLQSSessionState, 1496
tFNSLQSSignalStrengths, 1497
tFNSLQSWDSEvent, 1497
tFNSMSEvents, 1497
tFNSUPSInfo, 1497
tFNSUPSNotification, 1497
tFNSensorStreaming, 1494
tFNServingSystem, 1494
tFNSetCradleMount, 1495
tFNSetEngineState, 1495
tFNSetEventTimeSync, 1495
tFNSetExtPowerConfig, 1495
tFNSigInfo, 1495
tFNSignalStrength, 1496
tFNSwiAvmEventReport, 1498
tFNSysInfo, 1498
tFNSysSelectionPref, 1498
tFNUIMRefresh, 1499
tFNUIMStatusChangeInfo, 1499
tFNUSSDNoWaitIndication, 1499
tFNUSSDNotification, 1499
tFNUSSDRelease, 1500
tFNtransLayerInfo, 1498

tFNtransNWRegInfo, 1498
THIRD_INSTANCE, 1463
transLayerNotification, 1500
transNWRegInfoNotification, 1500
USSD_DCS_8BIT, 1463
USSD_DCS_ASCII, 1463
USSD_DCS_UCS2, 1463
VOICE_SRV, 1463
WDS_SRV, 1463
qaGobiApiDcs.h, 1538
LEN, 1539
PORTNAM_LEN, 1539
QCWWAN2kConnect, 1539
QCWWAN2kEnumerateDevices, 1540
QCWWAN2kGetConnectedDeviceID, 1540
QCWWANConnect, 1541
QCWWANDisconnect, 1541
QCWWANEnumerateDevices, 1541
SLQSGetDeviceMode, 1542
SLQSGetNetStatistic, 1543
SLQSGetUsbPortNames, 1543
SLQSKillSDKProcess, 1543
SLQSSetLoggingMask, 1544
SLQSStart, 1544
SLQSStart_AVAgent, 1545
SLQSStartSrv, 1545
SetSDKImagePath, 1542
qaGobiApiDms.h, 1546
ActivateAutomatic, 1555
custFeaturesInfo, 1548
custFeaturesSetting, 1550
dmsCurrentPRLInfo, 1551
ERIFileparams, 1552
GetActivationState, 1556
GetDeviceCapabilities, 1556
GetFirmwareRevision, 1557
GetFirmwareRevisions, 1558
GetHardwareRevision, 1558
GetIMSI, 1559
GetManufacturer, 1559
GetModelID, 1560
GetNetworkTime, 1560
GetOfflineReason, 1561
GetPRLVersion, 1562
GetPower, 1562
GetSerialNumbers, 1562
GetVoiceNumber, 1563
MAX_CUST_ID_LEN, 1548
MAX_FSN_LENGTH, 1548
ResetToFactoryDefaults, 1564
SLQSDmsSwiGetResetInfo, 1565
SLQSDmsSwiIndicationRegister, 1565
SLQSGetBandCapabilities, 1565
SLQSGetBandCapability, 1565
SLQSGetCurrentPRLInfo, 1567
SLQSGetCustFeatures, 1567
SLQSGetCustFeaturesV2, 1568
SLQSGetERIFile, 1568
SLQSGetSerialNumbers, 1568
SLQSSetCustFeatures, 1569
SLQSSetCustFeaturesV2, 1569
SLQSSwiClearDyingGaspStatistics, 1570
SLQSSwiGetCrashAction, 1570
SLQSSwiGetCrashInfo, 1570
SLQSSwiGetDyingGaspCfg, 1571
SLQSSwiGetDyingGaspStatistics, 1571
SLQSSwiGetFSN, 1571
SLQSSwiGetFwUpdateStatus, 1572
SLQSSwiGetHostDevInfo, 1572
SLQSSwiGetHostDevInfoParams, 1553
SLQSSwiGetOSInfo, 1573
SLQSSwiGetOSInfoParams, 1553
SLQSSwiGetSerialNoExt, 1573
SLQSSwiGetSerialNoExtParams, 1554
SLQSSwiGetUSBComp, 1573
SLQSSwiSetCrashAction, 1574
SLQSSwiSetDyingGaspCfg, 1574
SLQSSwiSetHostDevInfo, 1575
SLQSSwiSetHostDevInfoParams, 1554
SLQSSwiSetOSInfo, 1575
SLQSSwiSetOSInfoParams, 1555
SLQSSwiSetUSBComp, 1575
SLQSUIMGetState, 1576
serialNumbersInfo, 1552
SetPower, 1564
UIMChangePIN, 1576
UIMGetControlKeyStatus, 1577
UIMGetICCID, 1578
UIMGetPINStatus, 1579
UIMSetControlKeyProtection, 1580
UIMSetPINProtection, 1580
UIMUnblockControlKey, 1581
UIMUnblockPIN, 1582
UIMVerifyPIN, 1583
ValidateSPC, 1583
qaGobiApiFms.h, 1584
BUILD_ID_LEN, 1588
DEVICE_OFFLINE, 1588
DEVICE_RESET, 1588
DEVICE_SHUTDOWN, 1588
DeleteStoredImage, 1591
DownloadToSlot, 1591
eGetDeviceSeries, 1592
eGobiDeviceSeries, 1589
eGobiImageCarrier, 1589
eGobiImageGPS, 1590
eGobiImageRegion, 1590
eGobiImageTech, 1590
GetImageStore, 1593
GetImagesPreference, 1592
GetStoredImages, 1593
IMG_ID_LEN, 1588
IMGDETAILS_LEN, 1588
PRI_UPDATE_FAIL, 1588
SLQSDownloadFirmwareToSlot, 1594
SLQSGetBootVersionNumber, 1595

SLQSGetFirmwareInfo, 1596
SLQSGetImageInfo, 1596
SLQSGetImageInfo_9x15, 1597
SLQSGetImageInfoMC77xx, 1597
SLQSGetImageInfoMC83xx, 1598
SLQSGetValidFwPriCombinations, 1598
SLQSIsSpkgFormatRequired, 1599
SLQSSetCrashStateCheckIgnore, 1599
SLQSSetSIMBasedImageSwitching, 1600
SLQSSetSpkgFormatRequired, 1600
SLQSSwiGetAllCarrierImages, 1600
SLQSSwiGetFirmwareCurr, 1601
SLQSUpgradeFirmware9x15, 1601
SetImagesPreference, 1593
UNIQUE_ID_LEN, 1588
upgrade_mc77xx_fw, 1602
UpgradeFirmware2k, 1602
qaGobiApiIms.h, 1603
SLQSGetIMSSMSConfig, 1604
SLQSGetIMSUserConfig, 1604
SLQSGetIMSVoIPConfig, 1605
SLQSGetRegMgrConfig, 1605
SLQSGetSIPConfig, 1606
SLQSImsConfigIndicationRegister, 1606
SLQSSetIMSSMSConfig, 1607
SLQSSetIMSUserConfig, 1607
SLQSSetIMSVoIPConfig, 1608
SLQSSetRegMgrConfig, 1608
SLQSSetSIPConfig, 1609
qaGobiApiImsa.h, 1609
SLQSGetIMSARegStatus, 1610
SLQSGetIMSAServiceStatus, 1610
SLQSGetIMSASupportedFields, 1611
SLQSGetIMSASupportedMsg, 1611
SLQSRegisterIMSAIndication, 1612
qaGobiApiLoc.h, 1612
SLQSLOCDelAssData, 1613
SLQSLOCEventRegister, 1614
SLQSLOCGetBestAvailPos, 1614
SLQSLOCInjectPosition, 1615
SLQSLOCInjectSensorData, 1615
SLQSLOCInjectUTCTime, 1616
SLQSLOCSetCradleMountConfig, 1616
SLQSLOCSetExtPowerState, 1616
SLQSLOCSetOpMode, 1617
SLQSLOCStart, 1617
SLQSLOCStop, 1618
SwiLocGetAutoStart, 1618
SwiLocSetAutoStart, 1619
qaGobiApiNas.h, 1619
eSYS_SRV_DOMAIN, 1636
GetACCOLC, 1637
GetANAAAAuthenticationStatus, 1637
GetCDMANetworkParameters, 1638
GetHomeNetwork, 1639
GetHomeNetwork3GPP2, 1640
GetNetworkPreference, 1642
GetRFInfo, 1643
GetServingNetwork, 1643
GetServingNetworkCapabilities, 1644
GetSignalStrengths, 1645
IMSI_M_S1_LENGTH, 1624
IMSI_M_S2_LENGTH, 1624
InitiateDomainAttach, 1646
InitiateNetworkRegistration, 1646
MAX_PILOT_SETS, 1624
NAM_NAME_LENGTH, 1624
PLMN_LENGTH, 1625
PerformNetworkScan, 1647
SLQSConfigSigInfo, 1650
SLQSGetErrorRate, 1651
SLQSGetNetworkTime, 1651
SLQSGetOperatorNameData, 1651
SLQSGetPLMNName, 1652
SLQSGetServingSystem, 1652
SLQSGetSignalStrength, 1653
SLQSGetSysSelectionPref, 1653
SLQSInitiateNetworkRegistration, 1654
SLQSNASGetLTECPHYCaInfo, 1656
SLQSNASSwiGetChannelLock, 1659
SLQSNASSwiSetChannelLock, 1661
SLQSNasConfigSigInfo2, 1654
SLQSNasGet3GPP2Subscription, 1655
SLQSNasGetCellLocationInfo, 1655
SLQSNasGetHDRColorCode, 1656
SLQSNasGetSigInfo, 1656
SLQSNasGetSysInfo, 1657
SLQSNasGetTxRxInfo, 1657
SLQSNasIndicationRegister, 1658
SLQSNasIndicationRegisterExt, 1659
SLQSNasIndicationRegisterLTECphyCa, 1659
SLQSNasSwiIndicationRegister, 1660
SLQSNasSwiModemStatus, 1660
SLQSPerformNetworkScan, 1661
SLQSSetBandPreference, 1661
SLQSSetSysSelectionPref, 1663
SLQSSwiGetHDRPersonality, 1663
SLQSSwiGetHDRProtSubtype, 1664
SLQSSwiGetHRPDStats, 1664
SLQSSwiGetLteCQI, 1664
SLQSSwiGetLteSccRxInfo, 1665
SLQSSwiNetworkDebug, 1665
SLQSSwiPSDetach, 1666
SetACCOLC, 1647
SetCDMANetworkParameters, 1648
SetNetworkPreference, 1649
SlqsNas3GppNetworkRAT, 1625
slqsNetworkScanInfo, 1625
sysSelectPrefInfo, 1626
sysSelectPrefParams, 1630
UATISIZE, 1625
qaGobiApiOmadm.h, 1666
OMADMCancelSession, 1666
OMADMGetPendingNIA, 1667
OMADMGetSessionInfo, 1667
OMADMStartSession, 1669

qaGobiApiPds.h, 1669
DEFAULTBYTEVALUE, 1670
DEFAULTLONGVALUE, 1670
DEFAULTWORDVALUE, 1670
ForceXTRADownload, 1671
GetPDSDefaults, 1671
GetPDSState, 1672
GetPortAutomaticTracking, 1672
GetServiceAutomaticTracking, 1673
GetXTRAAutomaticDownload, 1673
GetXTRANetwork, 1674
GetXTRAValidity, 1674
PDSInjectTimeReference, 1675
ResetPDSData, 1675
SLQSGetAGPSConfig, 1679
SLQSGetGPSStateInfo, 1680
SLQSPDSDeterminePosition, 1680
SLQSPDSInjectAbsoluteTimeReference, 1681
SLQSPDSInjectPositionData, 1681
SLQSSetAGPSConfig, 1682
SLQSSetPositionMethodState, 1682
SetPDSDefaults, 1676
SetPDSState, 1677
SetPortAutomaticTracking, 1677
SetServiceAutomaticTracking, 1678
SetXTRAAutomaticDownload, 1678
SetXTRANetwork, 1679
StartPDSTrackingSessionExt, 1683
StopPDSTrackingSession, 1684
qaGobiApiQos.h, 1684
SLQSQosGetFlowStatus, 1685
SLQSQosGetGranted, 1685
SLQSQosGetNWProf, 1686
SLQSQosGetNetworkStatus, 1686
SLQSQosModify, 1687
SLQSQosRel, 1687
SLQSQosReq, 1688
SLQSQosReset, 1688
SLQSQosResume, 1689
SLQSQosSuspend, 1689
SLQSQosSwiReadApnExtraParams, 1689
SLQSQosSwiReadDataStats, 1690
qaGobiApiRms.h, 1690
GetSMSWake, 1691
SetSMSWake, 1691
qaGobiApiSar.h, 1692
eQMISARRFState, 1692
SLQSGetRfSarState, 1693
SLQSSetRfSarState, 1694
qaGobiApiSms.h, 1694
ABSOLUTE_VALIDITY, 1696
CONFIG_LEN, 1696
getIndicationRegResp, 1696
GetSMSCAddress, 1700
getTransLayerInfoResp, 1697
getTransNWRegInfoResp, 1697
MAX_SMS_ROUTES, 1696
NUM_OF_SET, 1696
qaQmi3GPP2BroadcastCfgInfo, 1698
qaQmi3GPPBroadcastCfgInfo, 1698
SLQSCDMADecodeMTTextMsg, 1702
SLQSCDMAEncodeMOTextMsg, 1703
SLQSDeleteSMS, 1703
SLQSGetIndicationRegister, 1704
SLQSGetMessageWaiting, 1705
SLQSGetSMS, 1705
SLQSGetSMSList, 1707
SLQSGetSmsBroadcastConfig, 1706
SLQSGetTransLayerInfo, 1708
SLQSGetTransNWRegInfo, 1708
SLQSModifySMSStatus, 1708
SLQSSendAsyncSMS, 1709
SLQSSendLongSMS, 1709
SLQSSendSMS, 1710
SLQSSetIndicationRegister, 1711
SLQSSetSmsBroadcastActivation, 1711
SLQSSetSmsBroadcastConfig, 1712
SLQSSetSmsStorage, 1712
SLQSSmsGetMaxStorageSize, 1713
SLQSSmsGetMessageProtocol, 1713
SLQSSmsSetRoutes, 1714
SLQSSwiGetSMSStorage, 1714
SLQSWCDMADecodeLongTextMsg, 1715
SLQSWCDMADecodeMTTextMsg, 1715
SLQSWCDMAEncodeMOTextMsg, 1715
SaveSMS, 1700
SendSMS, 1701
setIndicationRegReq, 1699
SetSMSCAddress, 1702
TIME_DATE_BUF, 1696
TIME_STAMP_BUF, 1696
transLayerInfo, 1699
qaGobiApiSwi.h, 1716
SLQSGetPidof, 1716
SLQSGetSdkVersion, 1717
SLQSSendRawQMI, 1717
qaGobiApiSwiAudio.h, 1717
SLQSGetM2MAVMute, 1719
SLQSGetM2MAudioProfile, 1718
SLQSGetM2MAudioVolume, 1719
SLQSGetM2MSpkrGain, 1719
SLQSSetM2MAVMute, 1722
SLQSSetM2MAudioAVCFG, 1720
SLQSSetM2MAudioLPBK, 1720
SLQSSetM2MAudioNVDef, 1721
SLQSSetM2MAudioProfile, 1721
SLQSSetM2MAudioVolume, 1721
SLQSSetM2MSpkrGain, 1722
qaGobiApiSwiAvms.h, 1723
QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 1725
QmiSwiAvmsEventReportConfigResp, 1726
QmiSwiAvmsEventReportConnectionRequest-Resp, 1726
QmiSwiAvmsEventReportDataSessionStatus-Resp, 1727

QmiSwiAvmsEventReportHTTPStatusResp, 1727
QmiSwiAvmsEventReportNotificationResp, 1728
QmiSwiAvmsEventReportPackageIDResp, 1729
QmiSwiAvmsEventReportRegStatusResp, 1729
QmiSwiAvmsEventReportSessionTypeResp, 1729
QmiSwiAvmsEventReportWAMSParamChange-Resp, 1730
QmiSwiAvmsSetSettingsAPNInfo, 1730
QmiSwiAvmsSetSettingsConnectionRetryTimers, 1730
QmiSwiAvmsSetSettingsPeriodInfo, 1730
SLQSAVMSGetSessionInfo, 1732
SLQSAVMSGetSettings, 1732
SLQSAVMSSelection, 1732
SLQSAVMSSetEventReport, 1733
SLQSAVMSSetSettings, 1733
SLQSAVMSSetSettingsIgnoreAutoReboot, 1734
SLQSAVMSSetSettingsReq, 1730
SLQSAVMSSettings, 1731
SLQSAVMSSettingsAPNInfo, 1731
SLQSAVMSSettingsConnectionRetryTimers, 1731
SLQSAVMSSettingsPeriodsInfo, 1731
SLQSAVMSStartSession, 1734
SLQSAVMSStopSession, 1735
qaGobiApiSwiOmadms.h, 1735
SLQSOMADMCancelSession, 1741
SLQSOMADMGetSessionInfo, 1741
SLQSOMADMGetSettings, 1742
SLQSOMADMGetSettings2, 1742
SLQSOMADMSendSelection, 1743
SLQSOMADMSendSelection2, 1743
SLQSOMADMSessionInfo, 1736
SLQSOMADMSetSettings, 1744
SLQSOMADMSetSettings2, 1745
SLQSOMADMSetSettings3, 1745
SLQSOMADMSettings, 1738
SLQSOMADMSettingsReqParams, 1739
SLQSOMADMSettingsReqParams3, 1740
SLQSOMADMStartSession, 1745
SLQSOMADMStartSession2, 1746
qaGobiApiTableBandClasses.h, 1746
qaGobiApiTableCallControlReturnReasons.h, 1749
qaGobiApiTableCallEndReasons.h, 1750
qaGobiApiTableCarrierCodes.h, 1765
qaGobiApiTableCodingScheme.h, 1767
qaGobiApiTableGpsCapabilityCodes.h, 1770
qaGobiApiTablePowerModes.h, 1770
qaGobiApiTableRadioInterfaces.h, 1771
qaGobiApiTableRegionCodes.h, 1772
qaGobiApiTableServiceOptions.h, 1772
qaGobiApiTableSupServiceInfoClasses.h, 1774
qaGobiApiTableSwiAudio.h, 1775
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 1775
qaGobiApiTableVoiceCallEndReasons.h, 1777
qaGobiApiTmd.h, 1783
SLQSTmdDeRegNotMitigationLvl, 1784
SLQSTmdGetMitigationDevList, 1784
SLQSTmdGetMitigationLvl, 1785
SLQSTmdRegNotMitigationLvl, 1785
qaGobiApiUim.h, 1785
MAX_ICCID_LENGTH, 1788
MAX_NO_OF_SLOTS, 1788
MAX_PATH_LENGTH, 1788
MAX_PUK_LENGTH, 1788
MAX_SLOTS_STATUS, 1788
SLQSUIMAuthenticate, 1788
SLQSUIMChangePin, 1788
SLQSUIMDepersonalization, 1789
SLQSUIMEventRegister, 1789
SLQSUIMGetCardStatus, 1790
SLQSUIMGetConfiguration, 1790
SLQSUIMGetFileAttributes, 1791
SLQSUIMGetSlotsStatus, 1792
SLQSUIMPowerDown, 1792
SLQSUIMPowerUp, 1792
SLQSUIMReadTransparent, 1793
SLQSUIMRefreshComplete, 1793
SLQSUIMRefreshGetLastEvent, 1794
SLQSUIMRefreshOK, 1794
SLQSUIMRefreshRegister, 1795
SLQSUIMReset, 1795
SLQSUIMSetPinProtection, 1796
SLQSUIMSwitchSlot, 1796
SLQSUIMUnblockPin, 1797
SLQSUIMVerifyPin, 1797
qaGobiApiVoice.h, 1798
AnswerUSSD, 1802
CancelUSSD, 1802
MAX_CALL_NO_LEN, 1801
MAX_NO_OF_CALLS, 1801
MAXUSSDLENGTH, 1801
OriginateUSSD, 1802
PASSWORD_LENGTH, 1801
SLQSOriginateUSSD, 1803
SLQSVoiceALSSelectLine, 1803
SLQSVoiceALSSetLineSwitching, 1804
SLQSVoiceAnswerCall, 1804
SLQSVoiceBindSubscription, 1805
SLQSVoiceBurstDTMF, 1805
SLQSVoiceDialCall, 1806
SLQSVoiceEndCall, 1806
SLQSVoiceGetAllCallInfo, 1807
SLQSVoiceGetCLIP, 1809
SLQSVoiceGetCLIR, 1810
SLQSVoiceGetCNAP, 1810
SLQSVoiceGetCOLP, 1811
SLQSVoiceGetCOLR, 1811
SLQSVoiceGetCallBarring, 1807
SLQSVoiceGetCallForwardingStatus, 1808
SLQSVoiceGetCallInfo, 1808
SLQSVoiceGetCallWaiting, 1809
SLQSVoiceGetConfig, 1812
SLQSVoiceIndicationRegister, 1813
SLQSVoiceManageCalls, 1813
SLQSVoiceOrigUSSDNoWait, 1814

SLQSVoiceSendFlash, 1814
SLQSVoiceSetCallBarringPassword, 1815
SLQSVoiceSetConfig, 1815
SLQSVoiceSetPreferredPrivacy, 1816
SLQSVoiceSetSUPSService, 1816
SLQSVoiceStartContDTMF, 1817
SLQSVoiceStopContDTMF, 1817
serviceClassInformation, 1801
qaGobiApiWds.h, 1818
GetAutoconnect, 1825
GetByteTotals, 1825
GetConnectionRate, 1826
GetDataBearerTechnology, 1827
GetDefaultProfile, 1827
GetDefaultProfileLTE, 1829
GetDefaultProfileNum, 1831
GetDormancyState, 1831
GetIPAddressLTE, 1832
GetLastMobileIPError, 1832
GetMobileIP, 1833
GetMobileIPProfile, 1833
GetPacketStatistics, 1834
GetPacketStatus, 1835
GetProfileSettingIn, 1822
GetProfileSettingOut, 1822
GetSessionDuration, 1836
GetSessionState, 1836
iGetByteTotals, 1837
iGetConnectionRate, 1837
iGetPacketStatistics, 1837
iSLQSMISetIPFamilyPreference, 1837
qmiDataBearerMasks, 1825
QmiProfileInfo, 1822
QmiWDSDataBearerTechnology, 1823
QmiWDSDataBearers, 1822
RMSetTransferStatistics, 1837
SLQSAutoConnect, 1845
SLQSCreateProfile, 1846
SLQSDeleteProfile, 1846
SLQSGet3GPPConfigItem, 1847
SLQSGetByteTotals, 1847
SLQSGetConnectionRate, 1848
SLQSGetCurrDataSystemStat, 1848
SLQSGetCurrentChannelRate, 1849
SLQSGetDUNCallInfo, 1850
SLQSGetDataBearerTechnology, 1849
SLQSGetDataBearerTechnologyExt, 1850
SLQSGetPacketStatistics, 1850
SLQSGetProfile, 1851
SLQSGetProfileSettings, 1852
SLQSGetRuntimeSettings, 1853
SLQSGetSessionState, 1853
SLQSModifyProfile, 1854
SLQSResetPacketStatics, 1855
SLQSSGetDHCPv4ClientConfig, 1857
SLQSSGetLoopback, 1857
SLQSSSetDHCPv4ClientConfig, 1857
SLQSSSetLoopback, 1858
SLQSSet3GPPConfigItem, 1855
SLQSSetProfile, 1855
SLQSStartStopDataSession, 1858
SLQSWdsGoActive, 1858
SLQSWdsGoDormant, 1859
SLQSWdsSetEventReport, 1859
SLQSWdsSwiPDPRuntimeSettings, 1860
SetActiveMobileIPProfile, 1837
SetAutoconnect, 1838
SetDefaultProfile, 1838
SetDefaultProfileLTE, 1839
SetDefaultProfileLTEV2, 1841
SetDefaultProfileNum, 1842
SetMobileIP, 1843
SetMobileIPParameters, 1843
SetMobileIPProfile, 1844
slqs3GPPConfigItem, 1823
WDS_IsGobiDevice, 1860
qaNasGetRFBandInfo.h, 1860
PkQmiNasGetRFBandInfo, 1861
UpkQmiNasGetRFBandInfo, 1861
qaNasPerformNetworkScan.h, 1861
FORBIDDEN_INDEX, 1862
INDEX_ZERO, 1862
PREFERRED_INDEX, 1862
PkQmiNasPerformNetworkScan, 1862
ROAMING_INDEX, 1862
UpkQmiNasPerformNetworkScan, 1862
qaQmi3GPP2BroadcastCfgInfo
qaGobiApiSms.h, 1698
qaQmi3GPPBroadcastCfgInfo
qaGobiApiSms.h, 1698
qaQmi3Gpp2TimeZone, 734
daylightSavings, 734
leapSeconds, 734
localTimeOffset, 734
qaQmiInterfaceInfo, 735
qaQmiinstanceid, 735
qaQmisvctype, 735
v4sessionId, 735
v6sessionId, 735
qaQmiServingSystemParam, 735
BasestationID, 738
BasestationLatitude, 738
BasestationLongitude, 738
CDMA_P_Rev, 738
CDMASystemInfoExt, 738
CallBarStatus, 738
CellID, 738
concSvcInfo, 738
CurrentPLMN, 738
DTMInd, 739
DataSrvCapabilities, 739
defaultRoamInd, 739
DetailedSvcInfo, 739
Gpp2TimeZone, 739
GppNetworkDSTAdjustment, 739
GppTimeZone, 739

hdrPersonality, 739
Lac, 739
NetworkID, 739
PRLInd, 739
roamIndicatorVal, 739
RoamingIndicatorList, 739
ServingSystem, 739
SystemID, 739
trackAreaCode, 739
qaQmiinstanceid
qaQmiInterfaceInfo, 735
qaQmisvctype
qaQmiInterfaceInfo, 735
qm_wds_ds_profile_extended_err_codes
qmerrno.h, 1868
qmerrno.h
eQCWWAN_ERR_API_MUTEX_TIMEOUT, 1865
eQCWWAN_ERR_BUFFER_SZ, 1864
eQCWWAN_ERR_CANCEL_OP, 1865
eQCWWAN_ERR_DRIVER, 1865
eQCWWAN_ERR_ENUM_BEGIN, 1864
eQCWWAN_ERR_ENUM_END, 1865
eQCWWAN_ERR_FILE_COPY, 1864
eQCWWAN_ERR_FILE_OPEN, 1864
eQCWWAN_ERR_GENERAL, 1864
eQCWWAN_ERR_INTERNAL, 1864
eQCWWAN_ERR_INVALID_ARG, 1864
eQCWWAN_ERR_INVALID_DEVID, 1864
eQCWWAN_ERR_INVALID_FILE, 1864
eQCWWAN_ERR_INVALID_QMI_RSP, 1864
eQCWWAN_ERR_INVALID_XID, 1865
eQCWWAN_ERR_MALFORMED_QMI_RSP, 1864
eQCWWAN_ERR_MEMORY, 1864
eQCWWAN_ERR_MULTIPLE_DEVICES, 1865
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED, 1865
eQCWWAN_ERR_NO_CANCELABLE_OP, 1865
eQCWWAN_ERR_NO_CONNECTION, 1864
eQCWWAN_ERR_NO_DEVICE, 1864
eQCWWAN_ERR_NO_SIGNAL, 1865
eQCWWAN_ERR_NONE, 1864
eQCWWAN_ERR_NULL_TLV, 1868
eQCWWAN_ERR_OFFLINE, 1865
eQCWWAN_ERR_PDU_GENERATION, 1865
eQCWWAN_ERR_QMI_ABORTED, 1865
eQCWWAN_ERR_QMI_ACCESS_DENIED, 1867
eQCWWAN_ERR_QMI_ACK_NOT_SENT, 1867
eQCWWAN_ERR_QMI_ARG_TOO_LONG, 1865
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, 1866
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, 1866
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, 1867
eQCWWAN_ERR_QMI_CALL_FAILED, 1865
eQCWWAN_ERR_QMI_CARD_BUSY_RSP, 1868
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, 1867
eQCWWAN_ERR_QMI_CAT_END, 1868
eQCWWAN_ERR_QMI_CAT_START, 1868
eQCWWAN_ERR_QMI_CAUSE_CODE, 1866
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, 1865
eQCWWAN_ERR_QMI_CONNECT, 1864
eQCWWAN_ERR_QMI_DEVICE_IN_USE, 1865
eQCWWAN_ERR_QMI_DEVICE_NOT_READY, 1866
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, 1866
eQCWWAN_ERR_QMI_DISABLED, 1866
eQCWWAN_ERR_QMI_ENCODING, 1866
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, 1868
eQCWWAN_ERR_QMI_EVENT_REG_FAILED, 1868
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, 1867
eQCWWAN_ERR_QMI_FDN_RESTRICT, 1867
eQCWWAN_ERR_QMI_FLOW_SUSPENDED, 1866
eQCWWAN_ERR_QMI_GENERAL, 1866
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, 1867
eQCWWAN_ERR_QMI_IFACE, 1864
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, 1867
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, 1866
eQCWWAN_ERR_QMI_INCORRECT_PIN, 1865
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, 1867
eQCWWAN_ERR_QMI_INJECT_TIMEOUT, 1867
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, 1866
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, 1866
eQCWWAN_ERR_QMI_INTERNAL, 1865
eQCWWAN_ERR_QMI_INVALID_ARG, 1866
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, 1865
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, 1866
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, 1868
eQCWWAN_ERR_QMI_INVALID_HANDLE, 1865
eQCWWAN_ERR_QMI_INVALID_ID, 1866
eQCWWAN_ERR_QMI_INVALID_INDEX, 1866
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, 1866
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, 1866
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, 1866

eQCWWAN_ERR_QMI_INVALID_OPERATION, 1867
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, 1865
eQCWWAN_ERR_QMI_INVALID_PINID, 1865
eQCWWAN_ERR_QMI_INVALID_PROFILE, 1865
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, 1865
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, 1866
eQCWWAN_ERR_QMI_INVALID_QMI_CMD, 1867
eQCWWAN_ERR_QMI_INVALID_QOS_ID, 1866
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, 1866
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, 1865
eQCWWAN_ERR_QMI_INVALID_TECH_PREF, 1865
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, 1868
eQCWWAN_ERR_QMI_INVALID_TRANSITION, 1866
eQCWWAN_ERR_QMI_INVALID_TX_ID, 1865
eQCWWAN_ERR_QMI_MALFORMED_MSG, 1865
eQCWWAN_ERR_QMI_MAX, 1867
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, 1866
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, 1866
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, 1866
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, 1866
eQCWWAN_ERR_QMI_MISSING_ARG, 1865
eQCWWAN_ERR_QMI_MSG_BLOCKED, 1867
eQCWWAN_ERR_QMI_NETWORK_ABORTED, 1867
eQCWWAN_ERR_QMI_NETWORK_NOT_READY, 1866
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, 1866
eQCWWAN_ERR_QMI_NO_EFFECT, 1865
eQCWWAN_ERR_QMI_NO_ENTRY, 1866
eQCWWAN_ERR_QMI_NO_FREE_PROFILE, 1865
eQCWWAN_ERR_QMI_NO_MEMORY, 1865
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, 1865
eQCWWAN_ERR_QMI_NO_RADIO, 1867
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, 1867
eQCWWAN_ERR_QMI_NO_THRESHOLDS, 1865
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, 1866
eQCWWAN_ERR_QMI_NOT_PROVISIONED, 1865
eQCWWAN_ERR_QMI_NOT_SUPPORTED, 1867
eQCWWAN_ERR_QMI_OFFSET, 1865
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, 1865
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, 1865
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, 1867
eQCWWAN_ERR_QMI_OUT_OF_CALL, 1865
eQCWWAN_ERR_QMI_PIN_BLOCKED, 1866
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, 1866
eQCWWAN_ERR_QMI_POLICY_MISMATCH, 1867
eQCWWAN_ERR_QMI_REQ, 1864
eQCWWAN_ERR_QMI_REQ_SCH, 1864
eQCWWAN_ERR_QMI_REQ_TO, 1864
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, 1866
eQCWWAN_ERR_QMI_RSP, 1864
eQCWWAN_ERR_QMI_RSP_TO, 1864
eQCWWAN_ERR_QMI_SEGMENT_ORDER, 1867
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, 1867
eQCWWAN_ERR_QMI_SESSION_INACTIVE, 1866
eQCWWAN_ERR_QMI_SESSION_INVALID, 1866
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, 1866
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, 1867
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, 1866
eQCWWAN_ERR_QMI_SMSC_ADDR, 1867
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, 1867
eQCWWAN_ERR_QMI_TPDU_TYPE, 1867
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, 1865
eQCWWAN_ERR_QMI_UNKNOWN, 1866
eQCWWAN_ERR_QMI_WIDTH, 1868
eQCWWAN_ERR_RESET, 1865
eQCWWAN_ERR_SWICM_AM_VERS_ERROR, 1867
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, 1867
eQCWWAN_ERR_SWICM_END, 1868
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, 1867
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, 1867
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, 1867

eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, 1867
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, 1867
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, 1867
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, 1867
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, 1867
eQCWWAN_ERR_SWICM_SOCKET_IN_USE, 1867
eQCWWAN_ERR_SWICM_START, 1867
eQCWWAN_ERR_SWICM_TIMEOUT, 1867
eQCWWAN_ERR_SWICM_V4DWN_V6DWN, 1867
eQCWWAN_ERR_SWICM_V4DWN_V6UP, 1867
eQCWWAN_ERR_SWICM_V4UP_V6DWN, 1867
eQCWWAN_ERR_SWICM_V4UP_V6UP, 1867
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, 1868
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, 1868
eQCWWAN_ERR_SWIDCS_END, 1868
eQCWWAN_ERR_SWIDCS_FILEIO_ERR, 1868
eQCWWAN_ERR_SWIDCS_IOCTL_ERR, 1868
eQCWWAN_ERR_SWIDCS_START, 1868
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, 1868
eQCWWAN_ERR_SWIIM_END, 1868
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, 1868
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, 1868
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, 1868
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, 1868
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX, 1868
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, 1868
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE, 1868
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, 1868
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, 1868
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, 1868
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE, 1868
eQCWWAN_ERR_SWIIM_INVALID_PATH, 1868
eQCWWAN_ERR_SWIIM_OPENING_DIR, 1868
eQCWWAN_ERR_SWIIM_OPENING_FILE, 1868
eQCWWAN_ERR_SWIIM_START, 1868
eQCWWAN_ERR_SWISM_END, 1868
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, 1868
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, 1868
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, 1868
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, 1868
eQCWWAN_ERR_SWISMS_START, 1868
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, 1869
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, 1869
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, 1869
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES, 1869
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY, 1869
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET, 1869
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET, 1869
eWDS_ERR_PROFILE_REG_END, 1869
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID, 1869
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED, 1869
eWDS_ERR_PROFILE_REG_RESULT_FAIL, 1869
eWDS_ERR_PROFILE_REG_RESULT_LIST_END, 1869
qmerrno.h, 1862
eQCWWANError, 1864
qm_wds_ds_profile_extended_err_codes, 1868
QmiCbkCatEventStatusReportInd, 739
CCETlv, 739
event_Index, 739
QmiCbkLocBestAvailPosInd, 739
pAltitudeWrtEllipsoid, 744
pAltitudeWrtMeanSeaLevel, 744
pGpsTime, 744

pHeading, 744
pHeadingUnc, 744
pHorCirConf, 744
pHorEllpConf, 744
pHorReliability, 744
pHorUncCircular, 744
pHorUncEllipseOrientAzimuth, 744
pHorUncEllipseSemiMajor, 744
pHorUncEllipseSemiMinor, 744
pLatitude, 745
pLongitude, 745
pMagneticDeviation, 745
pPrecisionDilution, 745
pSensorDataUsage, 745
pSpeedHorizontal, 745
pSpeedUnc, 745
pSpeedVertical, 745
pSpeedVerticalUnc, 745
pSvUsedforFix, 745
pTechnologyMask, 745
pTimeSrc, 745
pTimeUnc, 745
pTimestampUtc, 745
pVertConfidence, 745
pVertReliability, 745
pVertUnc, 745
pXid, 745
status, 745
QmiCbkLocCradleMountInd, 745
cradleMountConfigStatus, 746
QmiCbkLocEngineStateInd, 746
engineState, 746
QmiCbkLocEventTimeSyncInd, 746
timeSyncRefCounter, 747
QmiCbkLocInjectPositionInd, 747
status, 747
QmiCbkLocInjectSensorDataInd, 748
injectSensorDataStatus, 749
pAccelSamplesAccepted, 749
pAccelTempSamplesAccepted, 749
pGyroSamplesAccepted, 749
pGyroTempSamplesAccepted, 749
pOpaqueIdentifier, 749
QmiCbkLocInjectTimeInd, 749
injectTimeSyncStatus, 749
QmiCbkLocInjectUTCTimeInd, 749
status, 750
QmiCbkLocPositionReportInd, 750
pAltitudeAssumed, 755
pAltitudeWrtEllipsoid, 755
pAltitudeWrtMeanSeaLevel, 755
pFixId, 755
pGpsTime, 755
pHeading, 755
pHeadingUnc, 755
pHorConfidence, 755
pHorReliability, 755
pHorUncCircular, 755
pHorUncEllipseOrientAzimuth, 755
pHorUncEllipseSemiMajor, 755
pHorUncEllipseSemiMinor, 755
pLatitude, 755
pLeapSeconds, 755
pLongitude, 755
pMagneticDeviation, 755
pPrecisionDilution, 755
pSensorDataUsage, 755
pSpeedHorizontal, 755
pSpeedUnc, 755
pSpeedVertical, 755
pSvUsedforFix, 755
pTechnologyMask, 755
pTimeSrc, 755
pTimeUnc, 755
pTimestampUtc, 755
pVertConfidence, 755
pVertReliability, 756
pVertUnc, 756
sessionId, 756
sessionStatus, 756
QmiCbkLocSensorStreamingInd, 756
pAccelAcceptReady, 756
pAccelTempAcceptReady, 756
pGyroAcceptReady, 756
pGyroTempAcceptReady, 756
QmiCbkLocSetExtPowerConfigInd, 756
status, 757
QmiCbkNasLTECphyCaInfo, 757
sPhyCaAggPcellInfo, 758
sPhyCaAggScellDlBw, 758
sPhyCaAggScellIndType, 758
sPhyCaAggScellIndex, 758
sPhyCaAggScellInfo, 758
QmiCbkSwiOmaDmEventStatusReportInd, 758
SITlv, 758
QmiCbkSwiOmaDmEventStatusReportIndExt, 758
SITlv, 758
QmiCbkTmdMitiLvlRptInd, 758
currentMitigationLvl, 759
MitigationDevInfo, 759
QmiCbkWdsStatisticsIndState, 759
RxDropConutTlv, 759
RxOkByteCountTlv, 760
RxOkConutTlv, 760
TxDropConutTlv, 760
TxOkByteCountTlv, 760
TxOkConutTlv, 760
qmiDataBearerMasks
qaGobiApiWds.h, 1825
QmiNas3GppNetworkInfo, 761
pDesription, 762
pForbidden, 762
pInUse, 762
pMCC, 762
pMNC, 762
pPreferred, 762

pRoaming, 762
QmiNasGetRFBandInfoResp, 762
pInstancesSize, 762
pRFBandInfoElements, 762
results, 762
QmiNasPerformNetworkScanResp, 762
pInstanceSize, 763
pInstances, 763
results, 763
QmiProfileInfo
qaGobiApiWds.h, 1822
qmiSmsMessageList, 763
messageIndex, 763
messageTag, 763
QmiSwiAvmsEventReportBinaryUpdateSessionInfo-Resp
qaGobiApiSwiAvms.h, 1725
QmiSwiAvmsEventReportConfigResp
qaGobiApiSwiAvms.h, 1726
QmiSwiAvmsEventReportConnectionRequestResp
qaGobiApiSwiAvms.h, 1726
QmiSwiAvmsEventReportDataSessionStatusResp
qaGobiApiSwiAvms.h, 1727
QmiSwiAvmsEventReportHTTPStatusResp
qaGobiApiSwiAvms.h, 1727
QmiSwiAvmsEventReportNotificationResp
qaGobiApiSwiAvms.h, 1728
QmiSwiAvmsEventReportPackageIDResp
qaGobiApiSwiAvms.h, 1729
QmiSwiAvmsEventReportRegStatusResp
qaGobiApiSwiAvms.h, 1729
QmiSwiAvmsEventReportSessionTypeResp
qaGobiApiSwiAvms.h, 1729
QmiSwiAvmsEventReportWAMSParamChangeResp
qaGobiApiSwiAvms.h, 1730
QmiSwiAvmsSetSettingsAPNInfo
qaGobiApiSwiAvms.h, 1730
QmiSwiAvmsSetSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 1730
QmiSwiAvmsSetSettingsPeriodInfo
qaGobiApiSwiAvms.h, 1730
QmiWDSDataBearerTechnology
qaGobiApiWds.h, 1823
qmiWDSDataBearerTechnology, 763
currentNetwork, 763
ratMask, 763
soMask, 764
QmiWDSDataBearers
qaGobiApiWds.h, 1822
QmiWdsIpAddressInfo, 764
pIPAddressV4, 764
pIPAddressV6, 764
pIPv6prefixlen, 764
qmiWdsRunTimeSettings, 764
pAPNName, 767
pAuthentication, 767
pDomainList, 767
pGPRSGrantedQoS, 767
pGWAddressV4, 767
pIMCNflag, 767
pIPAddressV4, 767
pIPFamilyPreference, 767
pIPV6AddrInfo, 767
pIPV6GWAddrInfo, 767
pMtu, 767
pPCSCFAddrPCO, 767
pPCSCFFQDNAddrList, 767
pPDPType, 767
pPrimaryDNSV4, 768
pPrimaryDNSV6, 768
pProfileID, 768
pProfileName, 768
pSecondaryDNSV4, 768
pSecondaryDNSV6, 768
pServerAddrList, 768
pSubnetMaskV4, 768
pTechnology, 768
pUMTSGrantedQoS, 768
pUsername, 768
qmifwinfo_s, 760
dev, 760
g, 761
s, 761
qos.h, 1869
pack_qos_SLQSQosGetNetworkStatus, 1871
pack_qos_SLQSQosSwiReadApnExtraParams, 1871
pack_qos_SLQSQosSwiReadDataStats, 1872
pack_qos_SLQSSetQosEventCallback, 1873
unpack_qos_SLQSQosGetNetworkStatus, 1873
unpack_qos_SLQSQosSwiReadApnExtraParams, 1874
unpack_qos_SLQSQosSwiReadDataStats, 1874
unpack_qos_SLQSSetQosEventCallback, 1875
unpack_qos_SLQSSetQosEventCallback_ind, 1875
unpack_qos_SLQSSetQosNWStatusCallback_ind, 1875
unpack_qos_SLQSSetQosPriEventCallback_ind, 1876
unpack_qos_SLQSSetQosStatusCallback_ind, 1877
qos_id
QosMap, 773
QosClassID, 768
gDlBitRate, 769
gUlBitRate, 769
maxDlBitRate, 769
maxUlBitRate, 769
QCI, 769
qosDeliveryOrder
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
QosEventInfo, 769

pDataBearer, 770
pPacketsCountRX, 770
pPacketsCountTX, 770
pTotalBytesRX, 770
pTotalBytesTX, 770
qosFlow
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
QosFlowInfo, 771
pBearerID, 771
pQFlowState, 771
pRxQFilter, 771
pRxQFlowGranted, 771
pTxQFilter, 771
pTxQFlowGranted, 771
unpack_qos_SLQSSetQosEventCallback_ind_t, 1098
QosFlowInfoState, 772
id, 772
isNewFlow, 772
state, 772
QosMap, 772
dscp, 773
qos_id, 773
state, 773
Quality of Service (QOS), 54

RAN
unpack_nas_GetServingNetwork_t, 1051
RAT
SlqsNas3GppNetworkRAT, 87
nas_QmiNas3GppNetworkRAT, 487
RATMask
CurrNetworkInfo, 201
currNetworkInfo, 201
wds_currNetworkInfo, 1294
REGISTER_EVENT
qaGobiApiCbk.h, 1462
REGISTER_SRV
qaGobiApiCbk.h, 1462
RFBandInfoElements, 780
activeBandClass, 781
activeChannel, 781
radioInterface, 781
unpack_nas_GetRFInfo_t, 1050
RFTlv
unpack_nas_SetEventReportInd_t, 1053
RMAutoConnect
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
RMSetTransferStatistics
qaGobiApiWds.h, 1837
ROAMING_INDEX
qaNasPerformNetworkScan.h, 1862
RPCause
SMSAsyncRawSend_s, 869
RPTlv
NASQmiCbkNasSystemSelPrefInd, 549
RRTlv
unpack_nas_SetEventReportInd_t, 1053
RSRPThresListLen
nas_RSRPThresh, 490
RSRPThresh, 784
RSRPThresh, 783
pRSRPThresList, 784
RSRPThresListLen, 784
RSRQThresListLen
nas_RSRQThresh, 491
RSRQThresh, 785
RSRQThresh, 784
pRSRQThresList, 785
RSRQThresListLen, 785
RSSIThresListLen
nas_RSSIThresh, 492
RSSIThresh, 786
RSSIThresh, 785
pRSSIThresList, 786
RSSIThresListLen, 786
RX_EC_IO
nas_NetworkStat1x, 470
NetworkStat1x, 563
RX_PWR
nas_NetworkStat1x, 470
nas_NetworkStatEVDO, 472
NetworkStat1x, 563
NetworkStatEVDO, 565
RXAGCList, 786
pRXAIG, 787
pRXComprSlope, 787
pRXComprThres, 787
pRXExpSlope, 787
pRXExpThres, 787
pRXStaticGain, 787
RXAVCList, 787
pAVRXAVCHeadroom, 787
pAVRXAVCSens, 787
RXChan
LTEInfo, 387
nas_LTEInfo, 451
rXDroppedCount
unpack_wds_GetPacketStatus_t, 1171
RXOKBytesCount
DUNCallInfoInd, 231
rXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1171
rXOkBytesCount
unpack_wds_GetPacketStatus_t, 1171
RXPCMIIRFltr, 789
pFlag, 790
pStage0Val, 790
pStage1Val, 790
pStage2Val, 790
pStage3Val, 790
pStage4Val, 790
pStageCnt, 790
rXPacketErrors
unpack_wds_GetPacketStatus_t, 1171

rXPacketOverflows
unpack_wds_GetPacketStatus_t, 1171
rXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1171
radio
unpack_nas_GetSignalStrengths_t, 1052
radio_if
nasGetTxRxInfoReq, 530
pack_nas_SLQSNasGetTxRxInfo_t, 617
radioIf
ecioListElement, 233
errorRateListElement, 237
nas_ecioListElement, 430
nas_errorRateListElement, 432
nas_rsrqInformation, 490
nas_rxSignalStrengthListElement, 495
nasNetworkReject, 539
rsrqInformation, 784
rxSignalStrengthListElement, 792
RadioIfaces
unpack_dms_GetDeviceCap_t, 994
unpack_dms_GetDeviceCapabilities_t, 995
unpack_nas_GetServingNetwork_t, 1051
RadioIfacesSize
unpack_dms_GetDeviceCap_t, 994
unpack_nas_GetServingNetwork_t, 1051
radioIfacesSize
unpack_dms_GetDeviceCapabilities_t, 995
radioInterface
nas_RFInfoTlv, 488
nas_roamIndList, 489
nas_servSystem, 498
nas_SignalStrengthTlv, 499
nas_timeInfo, 508
RFBandInfoElements, 781
roamIndList, 782
servSystem, 805
timeInfo, 922
radioInterfaceList
NASServingSystemInfo, 551
ServingSystemInfo, 803
radioInterfaceNo
NASServingSystemInfo, 551
ServingSystemInfo, 804
radioInterfaceSize
nas_RFInfoTlv, 488
range
Port, 717
unpack_qos_Port_t, 1093
RankIndicatorInd, 773
Count1, 773
Count2, 773
rat
CSGID, 193
MNRInfo, 406
nas_CSGID, 423
nas_MNRInfo, 467
ratMask
dataBearerTechnology, 211
qmiWDSDataBearerTechnology, 763
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
ratValue
DataBearerTech, 209
rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 600
rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 600
rawLen
fileAttributes, 241
uim_fileAttributes, 946
rawValue
fileAttributes, 241
uim_fileAttributes, 946
rcv4
ssdatasession_params, 889
rcv6
ssdatasession_params, 889
readResult, 773
content, 774
contentLen, 774
readTransparent
pack_uim_ReadTransparent_t, 650
UIMReadTransparentReq, 968
readTransparentInfo, 774
length, 774
offset, 774
Reason
pack_voice_SLQSVoiceGetCallForwardingStatus_t, 667
pack_voice_SLQSVoiceSetCallBarringPassword_t, 673
voiceGetCallFWReq, 1246
voiceSetCallBarringPwdInfo, 1269
reason
ccSUPSType, 161
pack_voice_SLQSVoiceGetCallBarring_t, 666
pack_voice_SLQSVoiceSetSUPSService_t, 677
redirNumInfo, 776
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1100
voice_ccSUPSType, 1212
voiceGetCallBarringReq, 1244
voiceSetSUPSServiceReq, 1276
receiptAction
sms_routeEntry, 863
smsRouteEntry, 881
receivedBytes
omaDmFotaTlvExt, 575
reconfigReqd
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1181
reconfiguration_required
slqsSessionStateInfo, 847
recordCount

fileAttributes, 241
uim_fileAttributes, 946
recordSize
fileAttributes, 241
uim_fileAttributes, 946
redirNumInfo, 774
numLen, 776
numPlan, 776
numType, 776
number, 776
PI, 776
reason, 776
SI, 776
RedirPartyNum
arrRedirPartyNum, 133
voice_arrRedirPartyNum, 1203
refData
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1016
refString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1016
ReferenceID
CatAlPhaIdentifierTlv, 158
CatEventIDDataTlv, 159
refpn
CDMAInfo, 164
nas_CDMAInfo, 414
refreshComplete
pack_uim_SLQSUIMRefreshComplete_t, 655
UIMRefreshCompleteReq, 970
RefreshMode
CatRefreshTlv, 160
RefreshStage
CatRefreshTlv, 160
regAction
nasInitNetworkReg, 535
pack_nas_SLQSInitiateNetworkRegistration_t, 612
RegForeignNID
unpack_nas_GetCDMANetworkParameters_t, 1046
RegForeignSID
unpack_nas_GetCDMANetworkParameters_t, 1046
RegHomeSID
unpack_nas_GetCDMANetworkParameters_t, 1046
regInd
_transLayerInfoNotification, 112
regPrd
AddCDMASysInfo, 116
nas_AddCDMASysInfo, 410
regRefresh
pack_uim_SLQSUIMRefreshRegister_t, 657
UIMRefreshRegisterReq, 973
regRejectInfoValid
GSMSysInfo, 294
LTESysInfo, 401
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_WCDMASysInfo, 520
WCDMASysInfo, 1292
regState
nas_servSystem, 498
servSystem, 805
RegStatus
_SwiAvmsEventInd, 100
Region
fwinfo_s, 248
registerFlag
registerRefresh, 777
uim_registerRefresh, 952
registerRefresh, 776
arrfileInfo, 777
numFiles, 777
registerFlag, 777
voteForInit, 777
RegistrationState
unpack_nas_GetServingNetwork_t, 1051
registrationState
NASServingSystemInfo, 551
ServingSystemInfo, 804
rejCause
GSMSysInfo, 294
LTESysInfo, 401
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_WCDMASysInfo, 521
WCDMASysInfo, 1292
rejectCause
nas_RejectReasonTlv, 488
nasNetworkReject, 539
rejectSrvDomain
GSMSysInfo, 294
LTESysInfo, 401
nas_GSMSysInfo, 440
nas_LTESysInfo, 464
nas_WCDMASysInfo, 521
WCDMASysInfo, 1292
reliabilityClass
GPRSQoS, 283
GPRSRequestedQoS, 284
LibPackGPRSRequestedQoS, 329
wds_GPRSQoS, 1300
remPartyNumber
remotePartyNum, 779
voice_remotePartyNum, 1228
remainingRetries, 777
unblockLeft, 777
verifyLeft, 777
Remote Management Service (RMS), 45
RemotePartyName
getAllCallRmtPtyName, 252
voice_getAllCallRmtPtyName, 1221
remotePartyName, 777
callerName, 778

codingScheme, 778
nameLen, 778
namePI, 778
RemotePartyNum
getAllCallRmtPtyNum, 252
voice_getAllCallRmtPtyNum, 1222
remotePartyNum, 778
numLen, 779
presentationInd, 779
remPartyNumber, 779
ReqFieldsList, 779
requestFields, 779
requestFieldsLen, 779
requestFields
ReqFieldsList, 779
requestFieldsLen
ReqFieldsList, 779
resBerRatio
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
ResCode
FirmwareUpdatStat, 244
GetAudioVolTLBConfigResp, 258
SetAudioVolTLBConfigResp, 811
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1016
reserved
omaDmFotaTlvExt, 575
resetInfoInd
pack_dms_SLQSDmsSwiIndicationRegister_t, 582
ResetInfoNotification
qaGobiApiCbk.h, 1468
ResetPDSData
qaGobiApiPds.h, 1675
ResetToFactoryDefaults
qaGobiApiDms.h, 1564
RespFieldsList, 780
responseFields, 780
responseFieldsLen, 780
responseFields
RespFieldsList, 780
responseFieldsLen
RespFieldsList, 780
results
QmiNasGetRFBandInfoResp, 762
QmiNasPerformNetworkScanResp, 763
RetryCount
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
revPolarity
lineCtrlInfo, 358
revTunneling
unpack_wds_GetMobileIPProfile_t, 1169
ReverseMac
nas_protocolSubtypeElement, 485
protocolSubtypeElement, 733
RfSarState
pack_sar_SLQSSetRfSarState_t, 634
RmTrasnferStaticsReq
pack_wds_RMSetTransferStatistics_t, 681
rmTrasnferStaticsReq, 781
bResetStatistics, 781
ulMask, 781
rms.h, 1877
pack_rms_GetSMSWake, 1878
pack_rms_SetSMSWake, 1878
unpack_rms_GetSMSWake, 1878
unpack_rms_SetSMSWake, 1879
RmtPtyNum
arrRemotePartyNum, 134
voice_arrRemotePartyNum, 1204
roamIndList, 781
numInstances, 782
radioInterface, 782
roamIndicator, 782
roamIndicator
nas_roamIndList, 489
roamIndList, 782
RoamIndicatorVal
unpack_nas_SLQSGetServingSystem_t, 1062
roamIndicatorVal
qaQmiServingSystemParam, 739
roamOrigVoiceSO
prefVoiceSO, 720
voice_prefVoiceSO, 1227
RoamPref
NASRoamPreferenceTlv, 550
roamStatus
nas_sysInfoCommon, 504
sysInfoCommon, 912
roamStatusValid
nas_sysInfoCommon, 504
sysInfoCommon, 912
roamTimer, 782
namID, 783
roamTimerValue, 783
roamTimerValue
roamTimer, 783
voice_roamTimer, 1229
Roaming
nas_QmiNas3GppNetworkInfo, 486
SlqsNas3GppNetworkInfo, 842
unpack_nas_GetCDMANetworkParameters_t, 1046
unpack_nas_GetServingNetwork_t, 1051
roaming
unpack_nas_SetRoamingIndicatorCallback_ind_t, 1056
roaming_ind
RoamingInfo, 782
RoamingIndicatorList
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
RoamingInfo, 782

roaming_ind, 782
TlvPresent, 782
routeList
sms_setRoutesReq, 867
smsSetRoutesReq, 881
routeStorage
sms_routeEntry, 863
smsRouteEntry, 881
rptRate
LTESigRptCfg, 395
LTESigRptConfig, 396
nas_LTESigRptCfg, 459
nas_LTESigRptConfig, 460
rscp
nas_rxInfo, 493
nas_UMTSInfo, 511
rxInfo, 788
TDSCDMASigInfoExt, 915
tdscdmaSigInfoExt, 915
UMTSInfo, 982
rsrp
cellParams, 176
LTESSInfo, 399
lteSSInfo, 399
nas_cellParams, 420
nas_rxInfo, 493
nas_RxSigInfo, 494
nas_umtsLTENbrCell, 513
rxInfo, 788
RxSigInfo, 791
umtsLTENbrCell, 983
rsrplevel
lteRsrpinformation, 393
nas_lteRsrpinformation, 457
rsrq
cellParams, 176
LTESSInfo, 399
lteSSInfo, 399
nas_cellParams, 420
nas_rsrqInformation, 491
nas_SccRxInfo, 495
nas_umtsLTENbrCell, 513
rsrqInformation, 784
SccRxInfo, 796
umtsLTENbrCell, 983
rsrqDelta
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
rsrqInfo
nas_SLQSSignalStrengthsInformation, 501
slqsSignalStrengthInfo, 849
SLQSSignalStrengthsInformation, 853
unpack_nas_SLQSGetSignalStrength_t, 1063
rsrqInformation, 784
radioIf, 784
rsrq, 784
rssi
CDMASSInfo, 170
cdmaSSInfo, 170
cellParams, 176
gsmCellInfo, 289
HDRSSInfo, 301
hdrSSInfo, 302
LTESSInfo, 399
lteSSInfo, 399
nas_cellParams, 420
nas_gsmCellInfo, 436
TDSCDMASigInfoExt, 915
tdscdmaSigInfoExt, 915
unpack_nas_GetSignalStrengths_t, 1052
rts
WdsRunTimeSettings, 1321
rx_bytes
NetStats, 560
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
rx_errors
NetStats, 560
rx_overflows
NetStats, 560
rx_packets
NetStats, 560
rx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
rxChainIndex
nas_RxSigInfo, 494
RxSigInfo, 791
RxDropConutTlv
QmiCbkWdsStatisticsIndState, 759
unpack_RMTransferStatistics_ind_t, 1111
rxInfo, 787
ecio, 788
isRadioTuned, 788
phase, 788
rscp, 788
rsrp, 788
rxPower, 788
rxLev
GERANInfo, 250
nas_GERANInfo, 434
rxOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1177
RxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 760
unpack_RMTransferStatistics_ind_t, 1111
RxOkConutTlv
QmiCbkWdsStatisticsIndState, 760
unpack_RMTransferStatistics_ind_t, 1111
rxPower
nas_rxInfo, 493
nas_RxSigInfo, 494
rxInfo, 788
RxSigInfo, 791
RxQFilter
unpack_qos_QosFlowInfo_t, 1094

RxQFlowGranted
unpack_qos_QosFlowInfo_t, 1094
RxSigInfo, 790
isRadioTuned, 791
rsrp, 791
rxChainIndex, 791
rxPower, 791
rxSignalStrength
nas_rxSignalStrengthListElement, 495
rxSignalStrengthListElement, 792
rxSignalStrengthDelta
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
rxSignalStrengthInfo
nas_SLQSSignalStrengthsInformation, 501
SLQSSignalStrengthsInformation, 853
rxSignalStrengthList
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
rxSignalStrengthListElement, 791
radioIf, 792
rxSignalStrength, 792
rxSignalStrengthListLen
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063

s
qmifwinfo_s, 761
SMS_EVENT_ETWS
qaGobiApiCbk.h, 1501
SMS_EVENT_ETWS_PLMN
qaGobiApiCbk.h, 1501
SMS_EVENT_MESSAGE_MODE
qaGobiApiCbk.h, 1501
SMS_EVENT_MT_MESSAGE
qaGobiApiCbk.h, 1501
SMS_EVENT_SMS_ON_IMS
qaGobiApiCbk.h, 1501
SMS_EVENT_SMSC_ADDRESS
qaGobiApiCbk.h, 1501
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE
qaGobiApiCbk.h, 1501
SYS_SRV_DOMAIN_CAMPED
qaGobiApiCbk.h, 1501
SYS_SRV_DOMAIN_CS_ONLY
qaGobiApiCbk.h, 1501
SYS_SRV_DOMAIN_CS_PS
qaGobiApiCbk.h, 1501
SYS_SRV_DOMAIN_NO_SRV
qaGobiApiCbk.h, 1501
SYS_SRV_DOMAIN_PS_ONLY
qaGobiApiCbk.h, 1501
sApnExtraParams, 792
ambr_dl, 793
ambr_dl_ext, 793
ambr_dl_ext2, 793
ambr_ul, 793
ambr_ul_ext, 793
ambr_ul_ext2, 793
apnId, 793
SCI
unpack_nas_GetCDMANetworkParameters_t, 1046
SCM
unpack_nas_GetCDMANetworkParameters_t, 1046
SDPTlv
NASQmiCbkNasSystemSelPrefInd, 549
SDU_HDR_LEN
common.h, 1327
SECOND_INSTANCE
qaGobiApiCbk.h, 1462
sGetDeviceSeriesResult, 831
eDevice, 831
uResult, 831
SHORT
SwiDataTypes.h, 1904
SI
calledPartyInfo, 146
callFWExtInfo, 150
callingPartyInfo, 154
redirNumInfo, 776
voice_callFWExtInfo, 1209
SITlv
QmiCbkSwiOmaDmEventStatusReportInd, 758
QmiCbkSwiOmaDmEventStatusReportIndExt, 758
sIntraSearch
LTEInfoIntrafreq, 389
nas_LTEInfoIntrafreq, 454
SLQSAVMSGetSessionInfo
qaGobiApiSwiAvms.h, 1732
SLQSAVMSGetSettings
qaGobiApiSwiAvms.h, 1732
SLQSAVMSSelection
qaGobiApiSwiAvms.h, 1732
SLQSAVMSSetEventReport
qaGobiApiSwiAvms.h, 1733
SLQSAVMSSetSettings
qaGobiApiSwiAvms.h, 1733
SLQSAVMSSetSettingsIgnoreAutoReboot
qaGobiApiSwiAvms.h, 1734
SLQSAVMSSetSettingsReq
qaGobiApiSwiAvms.h, 1730
SLQSAVMSSettings
qaGobiApiSwiAvms.h, 1731
SLQSAVMSSettingsAPNInfo
qaGobiApiSwiAvms.h, 1731
SLQSAVMSSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 1731
SLQSAVMSSettingsPeriodsInfo
qaGobiApiSwiAvms.h, 1731
SLQSAVMSStartSession
qaGobiApiSwiAvms.h, 1734
SLQSAVMSStopSession
qaGobiApiSwiAvms.h, 1735
SLQSAutoConnect
qaGobiApiWds.h, 1845

SLQSCDMADecodeMTTextMsg
qaGobiApiSms.h, 1702
SLQSCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1703
SLQSConfigSigInfo
qaGobiApiNas.h, 1650
SLQSCreateProfile
qaGobiApiWds.h, 1846
SLQSDeleteProfile
qaGobiApiWds.h, 1846
SLQSDeleteProfileParams, 839
profileIndex, 840
profileType, 840
SLQSDeleteSMS
qaGobiApiSms.h, 1703
SLQSDmsSwiGetResetInfo
qaGobiApiDms.h, 1565
SLQSDmsSwiIndicationRegister
qaGobiApiDms.h, 1565
SLQSDownloadFirmwareToSlot
qaGobiApiFms.h, 1594
SLQSFWINFO_SKU_SZ
qaGobiApiFms.h, 1588
SLQSGet3GPPConfigItem
qaGobiApiWds.h, 1847
SLQSGetAGPSConfig
qaGobiApiPds.h, 1679
SLQSGetAudioPathConfig
qaGobiApiAudio.h, 1449
SLQSGetAudioProfile
qaGobiApiAudio.h, 1449
SLQSGetAudioVolTLBConfig
qaGobiApiAudio.h, 1450
SLQSGetBandCapabilities
qaGobiApiDms.h, 1565
SLQSGetBandCapability
qaGobiApiDms.h, 1565
SLQSGetBootVersionNumber
qaGobiApiFms.h, 1595
SLQSGetByteTotals
qaGobiApiWds.h, 1847
SLQSGetConnectionRate
qaGobiApiWds.h, 1848
SLQSGetCurrDataSystemStat
qaGobiApiWds.h, 1848
SLQSGetCurrentChannelRate
qaGobiApiWds.h, 1849
SLQSGetCurrentPRLInfo
qaGobiApiDms.h, 1567
SLQSGetCustFeatures
qaGobiApiDms.h, 1567
SLQSGetCustFeaturesV2
qaGobiApiDms.h, 1568
SLQSGetDUNCallInfo
qaGobiApiWds.h, 1850
SLQSGetDataBearerTechnology
qaGobiApiWds.h, 1849
SLQSGetDataBearerTechnologyExt
qaGobiApiWds.h, 1850
SLQSGetDeviceMode
qaGobiApiDcs.h, 1542
SLQSGetERIFile
qaGobiApiDms.h, 1568
SLQSGetErrorRate
qaGobiApiNas.h, 1651
SLQSGetFirmwareInfo
qaGobiApiFms.h, 1596
SLQSGetGPSStateInfo
qaGobiApiPds.h, 1680
SLQSGetIMSARegStatus
qaGobiApiImsa.h, 1610
SLQSGetIMSAServiceStatus
qaGobiApiImsa.h, 1610
SLQSGetIMSASupportedFields
qaGobiApiImsa.h, 1611
SLQSGetIMSASupportedMsg
qaGobiApiImsa.h, 1611
SLQSGetIMSSMSConfig
qaGobiApiIms.h, 1604
SLQSGetIMSUserConfig
qaGobiApiIms.h, 1604
SLQSGetIMSVoIPConfig
qaGobiApiIms.h, 1605
SLQSGetImageInfo
qaGobiApiFms.h, 1596
SLQSGetImageInfo_9x15
qaGobiApiFms.h, 1597
SLQSGetImageInfoMC77xx
qaGobiApiFms.h, 1597
SLQSGetImageInfoMC83xx
qaGobiApiFms.h, 1598
SLQSGetIndicationRegister
qaGobiApiSms.h, 1704
SLQSGetM2MAVMute
qaGobiApiSwiAudio.h, 1719
SLQSGetM2MAudioProfile
qaGobiApiSwiAudio.h, 1718
SLQSGetM2MAudioVolume
qaGobiApiSwiAudio.h, 1719
SLQSGetM2MSpkrGain
qaGobiApiSwiAudio.h, 1719
SLQSGetMessageWaiting
qaGobiApiSms.h, 1705
SLQSGetNetStatistic
qaGobiApiDcs.h, 1543
SLQSGetNetworkTime
qaGobiApiNas.h, 1651
SLQSGetOperatorNameData
qaGobiApiNas.h, 1651
SLQSGetPLMNName
qaGobiApiNas.h, 1652
SLQSGetPacketStatistics
qaGobiApiWds.h, 1850
SLQSGetPidof
qaGobiApiSwi.h, 1716
SLQSGetProfile

qaGobiApiWds.h, 1851
SLQSGetProfileSettings
qaGobiApiWds.h, 1852
SLQSGetRegMgrConfig
qaGobiApiIms.h, 1605
SLQSGetRfSarState
qaGobiApiSar.h, 1693
SLQSGetRuntimeSettings
qaGobiApiWds.h, 1853
SLQSGetSIPConfig
qaGobiApiIms.h, 1606
SLQSGetSMS
qaGobiApiSms.h, 1705
SLQSGetSMSList
qaGobiApiSms.h, 1707
SLQSGetSdkVersion
qaGobiApiSwi.h, 1717
SLQSGetSerialNumbers
qaGobiApiDms.h, 1568
SLQSGetServingSystem
qaGobiApiNas.h, 1652
SLQSGetSessionState
qaGobiApiWds.h, 1853
SLQSGetSignalStrength
qaGobiApiNas.h, 1653
SLQSGetSmsBroadcastConfig
qaGobiApiSms.h, 1706
SLQSGetSysSelectionPref
qaGobiApiNas.h, 1653
SLQSGetTransLayerInfo
qaGobiApiSms.h, 1708
SLQSGetTransNWRegInfo
qaGobiApiSms.h, 1708
SLQSGetUsbPortNames
qaGobiApiDcs.h, 1543
SLQSGetValidFwPriCombinations
qaGobiApiFms.h, 1598
SLQSImsConfigIndicationRegister
qaGobiApiIms.h, 1606
SLQSInitiateNetworkRegistration
qaGobiApiNas.h, 1654
SLQSIsSpkgFormatRequired
qaGobiApiFms.h, 1599
SLQSKillSDKProcess
qaGobiApiDcs.h, 1543
SLQSLOCDelAssData
qaGobiApiLoc.h, 1613
SLQSLOCEventRegister
qaGobiApiLoc.h, 1614
SLQSLOCGetBestAvailPos
qaGobiApiLoc.h, 1614
SLQSLOCInjectPosition
qaGobiApiLoc.h, 1615
SLQSLOCInjectSensorData
qaGobiApiLoc.h, 1615
SLQSLOCInjectUTCTime
qaGobiApiLoc.h, 1616
SLQSLOCSetCradleMountConfig
qaGobiApiLoc.h, 1616
SLQSLOCSetExtPowerState
qaGobiApiLoc.h, 1616
SLQSLOCSetOpMode
qaGobiApiLoc.h, 1617
SLQSLOCStart
qaGobiApiLoc.h, 1617
SLQSLOCStop
qaGobiApiLoc.h, 1618
SLQSModifyProfile
qaGobiApiWds.h, 1854
SLQSModifySMSStatus
qaGobiApiSms.h, 1708
SLQSNASGetLTECPHYCaInfo
qaGobiApiNas.h, 1656
SLQSNASSwiGetChannelLock
qaGobiApiNas.h, 1659
SLQSNASSwiSetChannelLock
qaGobiApiNas.h, 1661
SLQSNasConfigSigInfo2
qaGobiApiNas.h, 1654
SLQSNasGet3GPP2Subscription
qaGobiApiNas.h, 1655
SLQSNasGetCellLocationInfo
qaGobiApiNas.h, 1655
SLQSNasGetHDRColorCode
qaGobiApiNas.h, 1656
SLQSNasGetSigInfo
qaGobiApiNas.h, 1656
SLQSNasGetSysInfo
qaGobiApiNas.h, 1657
SLQSNasGetTxRxInfo
qaGobiApiNas.h, 1657
SLQSNasIndicationRegister
qaGobiApiNas.h, 1658
SLQSNasIndicationRegisterExt
qaGobiApiNas.h, 1659
SLQSNasIndicationRegisterLTECphyCa
qaGobiApiNas.h, 1659
SLQSNasNetworkRejectCallback
qaGobiApiCbk.h, 1516
SLQSNasNetworkTimeCallBack
qaGobiApiCbk.h, 1516
SLQSNasSigInfo2CallBack
qaGobiApiCbk.h, 1517
SLQSNasSigInfoCallBack
qaGobiApiCbk.h, 1517
SLQSNasSwiIndicationRegister
qaGobiApiNas.h, 1660
SLQSNasSwiModemStatus
qaGobiApiNas.h, 1660
SLQSNasSwiOTAMessageCallback
qaGobiApiCbk.h, 1518
SLQSNasSysInfoCallBack
qaGobiApiCbk.h, 1518
SLQSNasTimerCallback
qaGobiApiCbk.h, 1519
SLQSOMADMCancelSession

qaGobiApiSwiOmadms.h, 1741
SLQSOMADMGetSessionInfo
qaGobiApiSwiOmadms.h, 1741
SLQSOMADMGetSettings
qaGobiApiSwiOmadms.h, 1742
SLQSOMADMGetSettings2
qaGobiApiSwiOmadms.h, 1742
SLQSOMADMSendSelection
qaGobiApiSwiOmadms.h, 1743
SLQSOMADMSendSelection2
qaGobiApiSwiOmadms.h, 1743
SLQSOMADMSessionInfo
qaGobiApiSwiOmadms.h, 1736
SLQSOMADMSetSettings
qaGobiApiSwiOmadms.h, 1744
SLQSOMADMSetSettings2
qaGobiApiSwiOmadms.h, 1745
SLQSOMADMSetSettings3
qaGobiApiSwiOmadms.h, 1745
SLQSOMADMSettings
qaGobiApiSwiOmadms.h, 1738
SLQSOMADMSettingsReqParams
qaGobiApiSwiOmadms.h, 1739
SLQSOMADMSettingsReqParams3
qaGobiApiSwiOmadms.h, 1740
SLQSOMADMStartSession
qaGobiApiSwiOmadms.h, 1745
SLQSOMADMStartSession2
qaGobiApiSwiOmadms.h, 1746
SLQSOriginateUSSD
qaGobiApiVoice.h, 1803
SLQSPDSDeterminePosition
qaGobiApiPds.h, 1680
SLQSPDSInjectAbsoluteTimeReference
qaGobiApiPds.h, 1681
SLQSPDSInjectPositionData
qaGobiApiPds.h, 1681
SLQSPerformNetworkScan
qaGobiApiNas.h, 1661
SLQSQosGetFlowStatus
qaGobiApiQos.h, 1685
SLQSQosGetGranted
qaGobiApiQos.h, 1685
SLQSQosGetNWProf
qaGobiApiQos.h, 1686
SLQSQosGetNetworkStatus
qaGobiApiQos.h, 1686
SLQSQosModify
qaGobiApiQos.h, 1687
SLQSQosRel
qaGobiApiQos.h, 1687
SLQSQosReq
qaGobiApiQos.h, 1688
SLQSQosReset
qaGobiApiQos.h, 1688
SLQSQosResume
qaGobiApiQos.h, 1689
SLQSQosSuspend
qaGobiApiQos.h, 1689
SLQSQosSwiReadApnExtraParams
qaGobiApiQos.h, 1689
SLQSQosSwiReadDataStats
qaGobiApiQos.h, 1690
SLQSRegisterIMSAIndication
qaGobiApiImsa.h, 1612
SLQSResetPacketStatics
qaGobiApiWds.h, 1855
SLQSSGetDHCPv4ClientConfig
qaGobiApiWds.h, 1857
SLQSSGetLoopback
qaGobiApiWds.h, 1857
SLQSSSTlv
unpack_nas_SetEventReportInd_t, 1053
SLQSSSetDHCPv4ClientConfig
qaGobiApiWds.h, 1857
SLQSSSetLoopback
qaGobiApiWds.h, 1858
SLQSSendAsyncSMS
qaGobiApiSms.h, 1709
SLQSSendLongSMS
qaGobiApiSms.h, 1709
SLQSSendRawQMI
qaGobiApiSwi.h, 1717
SLQSSendSMS
qaGobiApiSms.h, 1710
SLQSSet3GPPConfigItem
qaGobiApiWds.h, 1855
SLQSSetAGPSConfig
qaGobiApiPds.h, 1682
SLQSSetAudioPathConfig
qaGobiApiAudio.h, 1450
SLQSSetAudioProfile
qaGobiApiAudio.h, 1451
SLQSSetAudioVolTLBConfig
qaGobiApiAudio.h, 1451
SLQSSetBandPreference
qaGobiApiNas.h, 1661
SLQSSetBandPreferenceCbk
qaGobiApiCbk.h, 1519
SLQSSetCrashStateCheckIgnore
qaGobiApiFms.h, 1599
SLQSSetCustFeatures
qaGobiApiDms.h, 1569
SLQSSetCustFeaturesV2
qaGobiApiDms.h, 1569
SLQSSetDHCPv4ClientLeaseStatusCallback
qaGobiApiCbk.h, 1520
SLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1520
SLQSSetDataSystemStatusCallback
qaGobiApiCbk.h, 1519
SLQSSetIMSAPdpStatusCallback
qaGobiApiCbk.h, 1521
SLQSSetIMSARatStatusCallback
qaGobiApiCbk.h, 1521
SLQSSetIMSARegStatusCallback

qaGobiApiCbk.h, 1521
SLQSSetIMSASvcStatusCallback
qaGobiApiCbk.h, 1522
SLQSSetIMSSMSConfig
qaGobiApiIms.h, 1607
SLQSSetIMSSMSConfigCallback
qaGobiApiCbk.h, 1522
SLQSSetIMSUserConfig
qaGobiApiIms.h, 1607
SLQSSetIMSUserConfigCallback
qaGobiApiCbk.h, 1523
SLQSSetIMSVoIPConfig
qaGobiApiIms.h, 1608
SLQSSetIMSVoIPConfigCallback
qaGobiApiCbk.h, 1523
SLQSSetIndicationRegister
qaGobiApiSms.h, 1711
SLQSSetLocInjectPositionCallback
qaGobiApiCbk.h, 1523
SLQSSetLocInjectUTCTimeCallback
qaGobiApiCbk.h, 1524
SLQSSetLoggingMask
qaGobiApiDcs.h, 1544
SLQSSetM2MAVMute
qaGobiApiSwiAudio.h, 1722
SLQSSetM2MAudioAVCFG
qaGobiApiSwiAudio.h, 1720
SLQSSetM2MAudioLPBK
qaGobiApiSwiAudio.h, 1720
SLQSSetM2MAudioNVDef
qaGobiApiSwiAudio.h, 1721
SLQSSetM2MAudioProfile
qaGobiApiSwiAudio.h, 1721
SLQSSetM2MAudioVolume
qaGobiApiSwiAudio.h, 1721
SLQSSetM2MSpkrGain
qaGobiApiSwiAudio.h, 1722
SLQSSetModemTempCallback
qaGobiApiCbk.h, 1524
SLQSSetPacketSrvStatusCallback
qaGobiApiCbk.h, 1524
SLQSSetPositionMethodState
qaGobiApiPds.h, 1682
SLQSSetProfile
qaGobiApiWds.h, 1855
SLQSSetQosEventCallback
qaGobiApiCbk.h, 1525
SLQSSetQosNWStatusCallback
qaGobiApiCbk.h, 1525
SLQSSetQosPriEventCallback
qaGobiApiCbk.h, 1526
SLQSSetQosStatusCallback
qaGobiApiCbk.h, 1526
SLQSSetRegMgrConfig
qaGobiApiIms.h, 1608
SLQSSetRegMgrConfigCallback
qaGobiApiCbk.h, 1526
SLQSSetRfSarState
qaGobiApiSar.h, 1694
SLQSSetSDKTerminatedCallback
qaGobiApiCbk.h, 1527
SLQSSetSIMBasedImageSwitching
qaGobiApiFms.h, 1600
SLQSSetSIPConfig
qaGobiApiIms.h, 1609
SLQSSetSIPConfigCallback
qaGobiApiCbk.h, 1529
SLQSSetSMSEventCallback
qaGobiApiCbk.h, 1529
SLQSSetServingSystemCallback
qaGobiApiCbk.h, 1527
SLQSSetSessionStateCallback
qaGobiApiCbk.h, 1528
SLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1528
SLQSSetSmsBroadcastActivation
qaGobiApiSms.h, 1711
SLQSSetSmsBroadcastConfig
qaGobiApiSms.h, 1712
SLQSSetSmsStorage
qaGobiApiSms.h, 1712
SLQSSetSpkgFormatRequired
qaGobiApiFms.h, 1600
SLQSSetSwiGetResetInfoCallback
qaGobiApiCbk.h, 1529
SLQSSetSwiHDRPersCallback
qaGobiApiCbk.h, 1530
SLQSSetSysSelectionPref
qaGobiApiNas.h, 1663
SLQSSetSysSelectionPrefCallBack
qaGobiApiCbk.h, 1530
SLQSSetTransLayerInfoCallback
qaGobiApiCbk.h, 1530
SLQSSetTransNWRegInfoCallback
qaGobiApiCbk.h, 1531
SLQSSetWdsEventCallback
qaGobiApiCbk.h, 1531
SLQSSetWdsTransferStatisticCallback
qaGobiApiCbk.h, 1532
SLQSSignalStrengthsIndReq, 850
ecioDelta, 851
ecioThresholdList, 851
ecioThresholdListLen, 851
ioDelta, 851
lteRsrpDelta, 851
lteSnrDelta, 851
rsrqDelta, 851
rxSignalStrengthDelta, 851
sinrDelta, 851
sinrThresholdList, 851
sinrThresholdListLen, 851
SLQSSignalStrengthsInformation, 852
ecioInfo, 852
errorRateInfo, 853
io, 853
lteRsrpinfo, 853

lteSnrinfo, 853
rsrqInfo, 853
rxSignalStrengthInfo, 853
sinr, 853
SLQSSmsGetMaxStorageSize
qaGobiApiSms.h, 1713
SLQSSmsGetMessageProtocol
qaGobiApiSms.h, 1713
SLQSSmsSetRoutes
qaGobiApiSms.h, 1714
SLQSStart
qaGobiApiDcs.h, 1544
SLQSStart_AVAgent
qaGobiApiDcs.h, 1545
SLQSStartSrv
qaGobiApiDcs.h, 1545
SLQSStartStopDataSession
qaGobiApiWds.h, 1858
SLQSSwiAVMSEventReportCallback
qaGobiApiCbk.h, 1533
SLQSSwiClearDyingGaspStatistics
qaGobiApiDms.h, 1570
SLQSSwiGetAllCarrierImages
qaGobiApiFms.h, 1600
SLQSSwiGetCrashAction
qaGobiApiDms.h, 1570
SLQSSwiGetCrashInfo
qaGobiApiDms.h, 1570
SLQSSwiGetDyingGaspCfg
qaGobiApiDms.h, 1571
SLQSSwiGetDyingGaspStatistics
qaGobiApiDms.h, 1571
SLQSSwiGetFSN
qaGobiApiDms.h, 1571
SLQSSwiGetFirmwareCurr
qaGobiApiFms.h, 1601
SLQSSwiGetFwUpdateStatus
qaGobiApiDms.h, 1572
SLQSSwiGetHDRPersonality
qaGobiApiNas.h, 1663
SLQSSwiGetHDRProtSubtype
qaGobiApiNas.h, 1664
SLQSSwiGetHRPDStats
qaGobiApiNas.h, 1664
SLQSSwiGetHostDevInfo
qaGobiApiDms.h, 1572
SLQSSwiGetHostDevInfoParams
qaGobiApiDms.h, 1553
SLQSSwiGetLteCQI
qaGobiApiNas.h, 1664
SLQSSwiGetLteSccRxInfo
qaGobiApiNas.h, 1665
SLQSSwiGetOSInfo
qaGobiApiDms.h, 1573
SLQSSwiGetOSInfoParams
qaGobiApiDms.h, 1553
SLQSSwiGetSMSStorage
qaGobiApiSms.h, 1714
SLQSSwiGetSerialNoExt
qaGobiApiDms.h, 1573
SLQSSwiGetSerialNoExtParams
qaGobiApiDms.h, 1554
SLQSSwiGetUSBComp
qaGobiApiDms.h, 1573
SLQSSwiNetworkDebug
qaGobiApiNas.h, 1665
SLQSSwiPSDetach
qaGobiApiNas.h, 1666
SLQSSwiSetCrashAction
qaGobiApiDms.h, 1574
SLQSSwiSetDyingGaspCfg
qaGobiApiDms.h, 1574
SLQSSwiSetHostDevInfo
qaGobiApiDms.h, 1575
SLQSSwiSetHostDevInfoParams
qaGobiApiDms.h, 1554
SLQSSwiSetOSInfo
qaGobiApiDms.h, 1575
SLQSSwiSetOSInfoParams
qaGobiApiDms.h, 1555
SLQSSwiSetUSBComp
qaGobiApiDms.h, 1575
SLQSTmdDeRegNotMitigationLvl
qaGobiApiTmd.h, 1784
SLQSTmdGetMitigationDevList
qaGobiApiTmd.h, 1784
SLQSTmdGetMitigationLvl
qaGobiApiTmd.h, 1785
SLQSTmdMitigationLvlRptCallback
qaGobiApiCbk.h, 1533
SLQSTmdRegNotMitigationLvl
qaGobiApiTmd.h, 1785
SLQSUIMAuthenticate
qaGobiApiUim.h, 1788
SLQSUIMChangePin
qaGobiApiUim.h, 1788
SLQSUIMDepersonalization
qaGobiApiUim.h, 1789
SLQSUIMEventRegister
qaGobiApiUim.h, 1789
SLQSUIMGetCardStatus
qaGobiApiUim.h, 1790
SLQSUIMGetConfiguration
qaGobiApiUim.h, 1790
SLQSUIMGetFileAttributes
qaGobiApiUim.h, 1791
SLQSUIMGetSlotsStatus
qaGobiApiUim.h, 1792
SLQSUIMGetState
qaGobiApiDms.h, 1576
SLQSUIMPowerDown
qaGobiApiUim.h, 1792
SLQSUIMPowerUp
qaGobiApiUim.h, 1792
SLQSUIMReadTransparent
qaGobiApiUim.h, 1793

SLQSUIMRefreshComplete
qaGobiApiUim.h, 1793
SLQSUIMRefreshGetLastEvent
qaGobiApiUim.h, 1794
SLQSUIMRefreshOK
qaGobiApiUim.h, 1794
SLQSUIMRefreshRegister
qaGobiApiUim.h, 1795
SLQSUIMReset
qaGobiApiUim.h, 1795
SLQSUIMSetPinProtection
qaGobiApiUim.h, 1796
SLQSUIMSetRefreshCallBack
qaGobiApiCbk.h, 1533
SLQSUIMSetStatusChangeCallBack
qaGobiApiCbk.h, 1533
SLQSUIMSwitchSlot
qaGobiApiUim.h, 1796
SLQSUIMUnblockPin
qaGobiApiUim.h, 1797
SLQSUIMVerifyPin
qaGobiApiUim.h, 1797
SLQSUpgradeFirmware9x15
qaGobiApiFms.h, 1601
SLQSVoiceALSSelectLine
qaGobiApiVoice.h, 1803
SLQSVoiceALSSetLineSwitching
qaGobiApiVoice.h, 1804
SLQSVoiceAnswerCall
qaGobiApiVoice.h, 1804
SLQSVoiceBindSubscription
qaGobiApiVoice.h, 1805
SLQSVoiceBurstDTMF
qaGobiApiVoice.h, 1805
SLQSVoiceDialCall
qaGobiApiVoice.h, 1806
SLQSVoiceEndCall
qaGobiApiVoice.h, 1806
SLQSVoiceGetAllCallInfo
qaGobiApiVoice.h, 1807
SLQSVoiceGetCLIP
qaGobiApiVoice.h, 1809
SLQSVoiceGetCLIR
qaGobiApiVoice.h, 1810
SLQSVoiceGetCNAP
qaGobiApiVoice.h, 1810
SLQSVoiceGetCOLP
qaGobiApiVoice.h, 1811
SLQSVoiceGetCOLR
qaGobiApiVoice.h, 1811
SLQSVoiceGetCallBarring
qaGobiApiVoice.h, 1807
SLQSVoiceGetCallForwardingStatus
qaGobiApiVoice.h, 1808
SLQSVoiceGetCallInfo
qaGobiApiVoice.h, 1808
SLQSVoiceGetCallWaiting
qaGobiApiVoice.h, 1809
SLQSVoiceGetConfig
qaGobiApiVoice.h, 1812
SLQSVoiceIndicationRegister
qaGobiApiVoice.h, 1813
SLQSVoiceInfoRecCallback
qaGobiApiCbk.h, 1534
SLQSVoiceManageCalls
qaGobiApiVoice.h, 1813
SLQSVoiceOrigUSSDNoWait
qaGobiApiVoice.h, 1814
SLQSVoiceSendFlash
qaGobiApiVoice.h, 1814
SLQSVoiceSetAllCallStatusCallBack
qaGobiApiCbk.h, 1534
SLQSVoiceSetCallBarringPassword
qaGobiApiVoice.h, 1815
SLQSVoiceSetConfig
qaGobiApiVoice.h, 1815
SLQSVoiceSetDTMFEventCallBack
qaGobiApiCbk.h, 1535
SLQSVoiceSetOTASPStatusCallBack
qaGobiApiCbk.h, 1535
SLQSVoiceSetPreferredPrivacy
qaGobiApiVoice.h, 1816
SLQSVoiceSetPrivacyChangeCallBack
qaGobiApiCbk.h, 1535
SLQSVoiceSetSUPSCallBack
qaGobiApiCbk.h, 1536
SLQSVoiceSetSUPSNotificationCallback
qaGobiApiCbk.h, 1536
SLQSVoiceSetSUPSService
qaGobiApiVoice.h, 1816
SLQSVoiceStartContDTMF
qaGobiApiVoice.h, 1817
SLQSVoiceStopContDTMF
qaGobiApiVoice.h, 1817
SLQSWCDMADecodeLongTextMsg
qaGobiApiSms.h, 1715
SLQSWCDMADecodeMTTextMsg
qaGobiApiSms.h, 1715
SLQSWCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1715
SLQSWdsGoActive
qaGobiApiWds.h, 1858
SLQSWdsGoDormant
qaGobiApiWds.h, 1859
SLQSWdsSetEventReport
qaGobiApiWds.h, 1859
SLQSWdsSwiPDPRuntimeSettings
qaGobiApiWds.h, 1860
SLQSWmsAsyncRawSendCallBack
qaGobiApiCbk.h, 1537
SLQSWmsMemoryFullCallBack
qaGobiApiCbk.h, 1537
SLQSWmsMessageWaitingCallBack
qaGobiApiCbk.h, 1538
SMS_CONFIG_LEN
sms.h, 1884

SMS_MAX_SMS_ROUTES
sms.h, 1884
SMS_NUM_OF_SET
sms.h, 1884
SMSAsyncRawSend
qaGobiApiCbk.h, 1469
SMSAsyncRawSend_s, 868
alphaIDLen, 869
causeCode, 869
errorClass, 869
messageID, 869
msgDelFailureCause, 869
msgDelFailureType, 869
pAlphaID, 869
RPCause, 869
sendStatus, 869
TPCause, 869
userData, 869
SMSC_TYPE_LEN
sms.h, 1884
SMSCAddress, 870
data, 870
length, 870
sMSCAddress, 869
data, 870
length, 870
SMSCAddressInfo
qaGobiApiCbk.h, 1470
sMSCAddressInfo
sms.h, 1885
sMSCAddressTlv, 870
SMSCInfo, 871
TlvPresent, 871
SMSCInfo
sMSCAddressTlv, 871
SMSCTlv
unpack_sms_SetNewSMSCallback_ind_t, 1114
SMSEtwsMessage, 871
data, 872
length, 872
notificationType, 872
sMSEtwsMessage, 871
data, 871
length, 871
notificationType, 871
SMSEtwsMessageInfo
qaGobiApiCbk.h, 1470
sMSEtwsMessageInfo
sms.h, 1885
sMSEtwsMessageTlv, 872
EtwsMessageInfo, 872
TlvPresent, 872
SMSEtwsPlmn, 872
mobileCountryCode, 873
mobileNetworkCode, 873
sMSEtwsPlmn, 873
mobileCountryCode, 873
mobileNetworkCode, 873
SMSEtwsPlmnInfo
qaGobiApiCbk.h, 1471
sMSEtwsPlmnInfo
sms.h, 1885
SMSEventInfo
qaGobiApiCbk.h, 1471
SMSEventInfo_s, 873
pEtwsMessageInfo, 874
pEtwsPlmnInfo, 874
pMTMessageInfo, 875
pMessageModeInfo, 874
pSMSCAddressInfo, 875
pSMSOnIMSInfo, 875
pTransferRouteMTMessageInfo, 875
smsEventType, 875
SMSEventType
qaGobiApiCbk.h, 1501
SMSMTMessage, 878
messageIndex, 878
storageType, 878
sMSMTMessage, 878
messageIndex, 878
storageType, 878
SMSMTMessageInfo
qaGobiApiCbk.h, 1472
sMSMTMessageInfo
sms.h, 1885
SMSMemoryInfo, 876
messageMode, 876
storageType, 876
SMSMessageMode, 876
messageMode, 877
sMSMessageMode, 877
messageMode, 877
SMSMessageModeInfo
qaGobiApiCbk.h, 1472
sMSMessageModeInfo
sms.h, 1885
SMSOnIMS, 879
smsOnIMS, 879
sMSOnIMS, 879
smsOnIMS, 879
SMSOnIMSInfo
qaGobiApiCbk.h, 1472
sMSOnIMSInfo
sms.h, 1885
sMSOnIMSTlv, 879
IMSInfo, 880
TlvPresent, 880
SMSSupport
pack_dms_SetCustFeature_t, 580
unpack_dms_GetCustFeature_t, 993
SMSTransferRouteMTMessage, 882
ackIndicator, 882
data, 882
format, 882
length, 882
transactionID, 882

sMSTransferRouteMTMessage, 882
ackIndicator, 883
data, 883
format, 883
length, 883
transactionID, 883
SMSTransferRouteMTMessageInfo
qaGobiApiCbk.h, 1472
sMSTransferRouteMTMessageInfo
sms.h, 1886
SNR
nas_NetworkStatEVDO, 472
NetworkStatEVDO, 565
sNonIntraSearch
LTEInfoIntrafreq, 389
nas_LTEInfoIntrafreq, 454
SO
nas_NetworkStat1x, 470
NetworkStat1x, 563
SOMask
CurrNetworkInfo, 201
currNetworkInfo, 201
wds_currNetworkInfo, 1294
SPC_SIZE
dms.h, 1337
sPhyCaAggPcellInfo
nasGetLTECphyCa, 525
QmiCbkNasLTECphyCaInfo, 758
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1054
sPhyCaAggScellDlBw
nasGetLTECphyCa, 525
QmiCbkNasLTECphyCaInfo, 758
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1054
sPhyCaAggScellIndType
nasGetLTECphyCa, 525
QmiCbkNasLTECphyCaInfo, 758
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1054
sPhyCaAggScellIndex
nasGetLTECphyCa, 525
QmiCbkNasLTECphyCaInfo, 758
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1054
sPhyCaAggScellInfo
nasGetLTECphyCa, 525
QmiCbkNasLTECphyCaInfo, 758
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1054
sQosFlowStat, 883
bearerId, 884
tx_bytes, 884
tx_bytes_drp, 884
tx_pkt, 884
tx_pkt_drp, 884
sQosStat, 884
apnId, 885
numQosFlow, 885
qosFlow, 885
total_rx_bytes, 885
total_rx_pkt, 885
total_tx_bytes, 885
total_tx_bytes_drp, 885
total_tx_pkt, 885
total_tx_pkt_drp, 885
SSInfo
unpack_nas_SetServingSystemCallback_ind_t, 1056
sSLQSSignalStrengthsInfo
nas_SLQSSignalStrengthsTlv, 501
SSTlv
unpack_nas_SetEventReportInd_t, 1053
SUPSInfo, 889
isModByCC, 890
svcType, 890
SUPSInformation
voiceSUPSInfo, 1280
SUPSType
pack_voice_SLQSVoiceManageCalls_t, 671
voiceManageCallsReq, 1263
SV, 890
id, 891
mask, 891
system, 891
SVInfo, 891
len, 891
pSV, 892
SWI Audio Service(SWIAUDIO), 56
SWI Open Mobile Alliance Service (SWIOMA), 49
SWI_API
SwiDataTypes.h, 1904
SWI_STRUCT_CarrierImage, 892
m_FwBuildId, 893
m_FwImageId, 893
m_PriBuildId, 893
m_PriImageId, 893
m_nCarrierId, 893
m_nFolderId, 893
m_nStorage, 893
SWIWWANCMAPI.h, 1915
samplesPerBatch
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 295
gyroTempAcceptReady_s, 295
sar.h, 1879
pack_sar_SLQSGetRfSarState, 1879
pack_sar_SLQSSetRfSarState, 1880
unpack_sar_SLQSGetRfSarState, 1880
unpack_sar_SLQSSetRfSarState, 1880
satelliteInfo, 793
azimuth, 795
elevation, 795
gnssSvId, 795
healthStatus, 795

snr, 795
svInfoMask, 795
svListLen, 795
svStatus, 795
system, 795
validMask, 795
SaveSMS
qaGobiApiSms.h, 1700
sbas_almanac_sv_msk
GPSStateInfo, 287
sbas_ephemeris_sv_msk
GPSStateInfo, 287
sbas_health_sv_msk
GPSStateInfo, 287
sbas_visible_sv_msk
GPSStateInfo, 288
SccRxInfo, 796
numInstances, 796
rsrq, 796
sigInfo, 796
snr, 796
TlvPresent, 796
scell_idx
nas_PhyCaAggScellIndex, 477
NASPhyCaAggScellIndex, 543
PhyCaAggScellIndex, 708
scell_state
nas_PhyCaAggScellIndType, 477
nas_PhyCaAggScellInfo, 480
NASPhyCaAggScellIndType, 544
NASPhyCaAggScellInfo, 545
PhyCaAggScellIndType, 709
PhyCaAggScellInfo, 712
screeningInd
connectNumInfo, 187
voice_connectNumInfo, 1218
sduErrorRatio
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
seDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
seDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
sePCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
sePCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1186
secActivate
fileAttributes, 241
uim_fileAttributes, 946
secActivateMask
fileAttributes, 241
uim_fileAttributes, 946
secChA
CDMAChannel, 162
nas_CDMAChannel, 413
secChB
CDMAChannel, 163
nas_CDMAChannel, 413
secDeactivate
fileAttributes, 241
uim_fileAttributes, 946
secDeactivateMask
fileAttributes, 241
uim_fileAttributes, 946
secIncrease
fileAttributes, 241
uim_fileAttributes, 946
secIncreaseMask
fileAttributes, 241
uim_fileAttributes, 946
SecProt
nas_protocolSubtypeElement, 485
protocolSubtypeElement, 733
secRead
fileAttributes, 241
uim_fileAttributes, 946
secReadMask
fileAttributes, 241
uim_fileAttributes, 946
secWrite
fileAttributes, 241
uim_fileAttributes, 946
secWriteMask
fileAttributes, 241
uim_fileAttributes, 946
secdns
unpack_wds_GetDefaultProfile_t, 1166
secdnsv6
unpack_wds_GetDefaultProfile_t, 1166
second
nas_timeInfo, 508
nas_UniversalTime, 514
timeInfo, 922
UniversalTime, 991
secondaryDNS
pack_wds_SetDefaultProfile_t, 682
SecondaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1179
SecondaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1180
secondaryHA
unpack_wds_GetMobileIPProfile_t, 1169
SectorIDLen
nas_NetworkStatEVDO, 472
NetworkStatEVDO, 565
selNetwork
nas_servSystem, 498
servSystem, 805
selected

BroadcastConfig, 142
CDMABroadcastConfig, 162
sms_BroadcastConfig, 855
sms_CDMABroadcastConfig, 856
selectedNetwork
NASServingSystemInfo, 551
ServingSystemInfo, 804
selection
pack_swioma_SLQSOMADMSendSelection_t, 646
SendSMS
qaGobiApiSms.h, 1701
sendStatus
SMSAsyncRawSend_s, 869
sensorData, 796
flags, 798
sensorDataLen, 798
timeOfFirstSample, 798
timeOffset, 798
xAxis, 798
yAxis, 798
zAxis, 798
sensorData_t, 798
flags, 799
sensorDataLen, 799
timeOfFirstSample, 799
timeOffset, 799
xAxis, 799
yAxis, 799
zAxis, 799
sensorDataLen
sensorData, 798
sensorData_t, 799
sensorDataUsage
qaGobiApiCbk.h, 1468
sensorDataUsage_s, 800
aidingIndicatorMask, 800
usageMask, 800
serialNumbersInfo, 800
esnSize, 801
imeiSize, 801
imeiSvnSize, 801
meidSize, 801
pESNString, 801
pIMEIString, 801
pImeiSvnString, 801
pMEIDString, 801
qaGobiApiDms.h, 1552
servSystem, 804
csAttachState, 805
numRadioInterfaces, 805
psAttachState, 805
radioInterface, 805
regState, 805
selNetwork, 805
ServerAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1180
serviceCategory
CDMABroadcastConfig, 162
sms_CDMABroadcastConfig, 856
serviceClassInformation
qaGobiApiVoice.h, 1801
serviceDomain
nas_RejectReasonTlv, 488
nasNetworkReject, 539
serviceProviderName, 802
displayCondition, 802
spn, 802
spnLength, 802
servingCellId
LTEInfoIntrafreq, 389
nas_LTEInfoIntrafreq, 454
ServingSystem
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
ServingSystemInfo, 802
csAttachState, 803
hdrPersonality, 803
psAttachState, 803
radioInterfaceList, 803
radioInterfaceNo, 804
registrationState, 804
selectedNetwork, 804
sessionEndReason
_packetSrvStatus, 70
slqsSessionStateInfo, 847
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1181
SessionId
LOCStartReq, 382
pack_loc_Start_t, 605
pack_loc_Stop_t, 605
sessionId
LOCStopReq, 382
QmiCbkLocPositionReportInd, 756
ssdatasession_params, 889
unpack_loc_PositionRpt_Ind_t, 1041
sessionInfo, 805
omaDmConfig, 806
omaDmFota, 806
omaDmNotifications, 806
pack_uim_ChangePin_t, 649
pack_uim_ReadTransparent_t, 650
pack_uim_SetPinProtection_t, 651
pack_uim_SLQSUIMAuthenticate_t, 652
pack_uim_SLQSUIMGetFileAttributes_t, 654
pack_uim_SLQSUIMRefreshComplete_t, 655
pack_uim_SLQSUIMRefreshGetLastEvent_t, 656
pack_uim_SLQSUIMRefreshOK_t, 656
pack_uim_SLQSUIMRefreshRegister_t, 657
pack_uim_UnblockPin_t, 658
pack_uim_VerifyPin_t, 660
sessionInfoTlv, 806
sessionInfoTlvExt, 807
UIMAuthenticateReq, 959
UIMChangePinReq, 960

UIMGetFileAttributesReq, 964
UIMReadTransparentReq, 968
UIMRefreshCompleteReq, 970
UIMRefreshGetLastEventReq, 972
UIMRefreshOKReq, 973
UIMRefreshRegisterReq, 973
UIMSetPinProtectionReq, 975
UIMUnblockPinReq, 979
UIMVerifyPinReq, 980
SessionInfoConfig
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1123
sessionInfoExt, 806
omaDmConfig, 806
omaDmFota, 806
SessionInfoFota
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1123
SessionInfoNotification
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1123
sessionInfoTlv, 806
sessionInfo, 806
sessionType, 806
TlvPresent, 806
sessionInfoTlvExt, 807
sessionInfo, 807
sessionType, 807
TlvPresent, 807
sessionInformation
qaGobiApiCbk.h, 1469
sessionInformationExt
qaGobiApiCbk.h, 1469
SessionState
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
sessionStatus
omaDmNotificationsTlv, 575
QmiCbkLocPositionReportInd, 756
unpack_loc_PositionRpt_Ind_t, 1041
unpack_omaDmNotificationsTlv_t, 1088
SessionType
_SwiAvmsEventInd, 100
pack_swioma_SLQSOMADMGetSessionInfo_t, 646
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
sessionType
omaDmFotaTlv, 573
pack_swioma_SLQSOMADMCancelSession_t, 645
pack_swioma_SLQSOMADMStartSession_t, 648
sessionInfoTlv, 806
sessionInfoTlvExt, 807
uim_refreshevent, 951
uim_sessionInformation, 953
uim_UIMSessionInformation, 957
UIMRefreshEvent, 971
UIMSessionInformation, 974
unpack_omaDmFotaTlv_t, 1087
set_fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 644
SwiLocSetAutoStartReq, 896
set_fix_type
pack_swiloc_SwiLocSetAutoStart_t, 645
SwiLocSetAutoStartReq, 897
set_function
pack_swiloc_SwiLocSetAutoStart_t, 645
SwiLocSetAutoStartReq, 897
set_max_dist
pack_swiloc_SwiLocSetAutoStart_t, 645
SwiLocSetAutoStartReq, 897
set_max_time
pack_swiloc_SwiLocSetAutoStart_t, 645
SwiLocSetAutoStartReq, 897
SetACCOLC
qaGobiApiNas.h, 1647
SetActivationStatusCallback
qaGobiApiCbk.h, 1502
SetActiveMobileIPProfile
qaGobiApiWds.h, 1837
SetAudioPathConfigReq, 807
pCodecSTGain, 808
pDTMFTXGain, 808
pECMode, 809
pNSEnable, 809
pRXAGCList, 809
pRXAVCAGCSwitch, 809
pRXAVCList, 809
pRXPCMIIRFltr, 809
pTXAGCList, 809
pTXAVCSwitch, 809
pTXGain, 809
pTXPCMIIRFltr, 809
Profile, 809
SetAudioProfileReq, 809
EarMute, 810
Generator, 810
MicMute, 810
Profile, 810
Volume, 810
SetAudioVolTLBConfigReq, 810
Generator, 811
Item, 811
Profile, 811
VolValue, 811
Volume, 811
SetAudioVolTLBConfigResp, 811
ResCode, 811
SetAutoconnect
qaGobiApiWds.h, 1838
SetCATEventCallback
qaGobiApiCbk.h, 1502
SetCDMANetworkParameters
qaGobiApiNas.h, 1648
setCustomSettingV2, 812

cust_id, 812
cust_value, 812
value_length, 812
SetDataCapabilitiesCallback
qaGobiApiCbk.h, 1503
SetDefaultProfile
qaGobiApiWds.h, 1838
SetDefaultProfileLTE
qaGobiApiWds.h, 1839
SetDefaultProfileLTEV2
qaGobiApiWds.h, 1841
SetDefaultProfileNum
qaGobiApiWds.h, 1842
SetDeviceStateChangeCbk
qaGobiApiCbk.h, 1504
setDyingGaspCfg, 812
pDestSMSContent, 813
pDestSMSNum, 813
SetFwDldCompletionCbk
qaGobiApiCbk.h, 1504
SetGPSCallback
qaGobiApiCbk.h, 1505
SetIMSSMSConfigReq, 813
pPhoneCtxtURI, 813
pPhoneCtxtURILen, 813
pSMSFormat, 813
pSMSOverIPNwInd, 813
SetIMSSMSConfigResp, 813
pSettingResp, 814
SetIMSUserConfigReq, 814
pIMSDomain, 814
pIMSDomainLen, 814
SetIMSUserConfigResp, 814
pSettingResp, 815
SetIMSVoIPConfigReq, 815
pAmrMode, 817
pAmrOctetAligned, 817
pAmrWBMode, 817
pAmrWBOctetAligned, 817
pAmrWbEnable, 817
pMinSessionExpiryTimer, 817
pRTPRTCPInactTimer, 817
pRingBackTimer, 817
pRingingTimer, 817
pScrAmrEnable, 817
pScrAmrWbEnable, 817
pSessionExpiryTimer, 817
SetIMSVoIPConfigResp, 817
pSettingResp, 818
SetImagesPreference
qaGobiApiFms.h, 1593
setIndicationRegReq
qaGobiApiSms.h, 1699
SetLURejectCallback
qaGobiApiCbk.h, 1508
SetLocBestAvailPosCallback
qaGobiApiCbk.h, 1505
SetLocCradleMountCallback
qaGobiApiCbk.h, 1505
SetLocDeleteAssistDataCallback
qaGobiApiCbk.h, 1506
SetLocEngineStateCallback
qaGobiApiCbk.h, 1506
SetLocEventPositionCallback
qaGobiApiCbk.h, 1506
SetLocEventTimeSyncCallback
qaGobiApiCbk.h, 1506
SetLocGnssSvInfoCallback
qaGobiApiCbk.h, 1507
SetLocInjectSensorDataCallback
qaGobiApiCbk.h, 1507
SetLocInjectTimeCallback
qaGobiApiCbk.h, 1507
SetLocOpModeCallback
qaGobiApiCbk.h, 1508
SetLocSensorStreamingCallback
qaGobiApiCbk.h, 1508
SetLocSetExtPowerConfigCallback
qaGobiApiCbk.h, 1508
SetM2MAVMuteReq, 821
EarMute, 821
MicMute, 821
pCwtMute, 821
Profile, 822
SetM2MAudioAVCFGReq, 818
Device, 818
PIFACEId, 818
pPCMParams, 818
Profile, 818
SetM2MAudioLPBKReq, 819
Enable, 819
SetM2MAudioProfileReq, 819
pCwtMute, 820
pEarMute, 820
pGenerator, 820
pMicMute, 820
pVolume, 820
Profile, 820
SetM2MAudioVolumeReq, 820
Generator, 821
Level, 821
Profile, 821
SetM2MSpkrGainReq, 822
Profile, 822
Value, 822
SetMobileIP
qaGobiApiWds.h, 1843
SetMobileIPParameters
qaGobiApiWds.h, 1843
SetMobileIPProfile
qaGobiApiWds.h, 1844
SetMobileIPStatusCallback
qaGobiApiCbk.h, 1509
SetNMEACallback
qaGobiApiCbk.h, 1511
SetNasLTECphyCaIndCallback

qaGobiApiCbk.h, 1509
SetNetChangeCbk
qaGobiApiCbk.h, 1510
SetNetworkPreference
qaGobiApiNas.h, 1649
SetNewSMSCallback
qaGobiApiCbk.h, 1510
SetOMADMStateCallback
qaGobiApiCbk.h, 1511
SetPDSDefaults
qaGobiApiPds.h, 1676
SetPDSState
qaGobiApiPds.h, 1677
SetPDSStateCallback
qaGobiApiCbk.h, 1511
setPINProtection, 822
pinID, 823
pinLength, 823
pinOperation, 823
pinValue, 823
SetPortAutomaticTracking
qaGobiApiPds.h, 1677
SetPower
qaGobiApiDms.h, 1564
SetPowerCallback
qaGobiApiCbk.h, 1512
SetRFInfoCallback
qaGobiApiCbk.h, 1512
SetRMTransferStatisticsCallback
qaGobiApiCbk.h, 1512
SetRankIndicatorCallback
qaGobiApiCbk.h, 1512
SetRegMgrConfigReq, 823
pCSCFPortName, 824
pCSCFPortNameLen, 824
pIMSTestMode, 824
pPriCSCFPort, 824
SetRegMgrConfigResp, 824
pSettingResp, 824
SetRoamingIndicatorCallback
qaGobiApiCbk.h, 1513
SetSDKImagePath
qaGobiApiDcs.h, 1542
SetSIPConfigReq, 829
pSIPLocalPort, 830
pSigCompEnabled, 830
pSubscribeTimer, 830
pTimerSIPReg, 830
pTimerT1, 830
pTimerT2, 830
pTimerTf, 830
SetSIPConfigResp, 830
pSettingResp, 831
SetSLQSOMADMAlertCallback
qaGobiApiCbk.h, 1514
SetSLQSOMADMAlertCallbackExt
qaGobiApiCbk.h, 1514
SetSMSCAddress
qaGobiApiSms.h, 1702
SetSMSWake
qaGobiApiRms.h, 1691
SetServiceAutomaticTracking
qaGobiApiPds.h, 1678
SetSignalStrengthCallback
qaGobiApiCbk.h, 1513
setSignalStrengthInfo, 824
pCDMAECIODelta, 828
pCDMAECIOThresh, 828
pCDMARSSIDelta, 828
pCDMARSSIThresh, 828
pGSMRSSIDelta, 828
pGSMRSSIThresh, 828
pHDRECIODelta, 828
pHDRECIOThresh, 828
pHDRIODelta, 828
pHDRIOThresh, 828
pHDRRSSIDelta, 828
pHDRRSSIThresh, 828
pHDRSINRDelta, 828
pHDRSINRThresh, 828
pLTERSRPDelta, 828
pLTERSRPThresh, 828
pLTERSRQDelta, 828
pLTERSRQThresh, 828
pLTERSSIDelta, 829
pLTERSSIThresh, 829
pLTESNRDelta, 829
pLTESNRThresh, 829
pLTESigRptConfig, 829
pTDSCDMAECIODelta, 829
pTDSCDMAECIOThresh, 829
pTDSCDMARSCPDelta, 829
pTDSCDMARSCPThresh, 829
pTDSCDMARSSIDelta, 829
pTDSCDMARSSIThresh, 829
pTDSCDMASINRDelta, 829
pTDSCDMASINRThresh, 829
pWCDMAECIODelta, 829
pWCDMAECIOThresh, 829
pWCDMARSSIDelta, 829
pWCDMARSSIThresh, 829
SetUSSDNoWaitIndicationCallback
qaGobiApiCbk.h, 1515
SetUSSDNotificationCallback
qaGobiApiCbk.h, 1515
SetUSSDReleaseCallback
qaGobiApiCbk.h, 1516
SetUimSlotStatusChangeCallback
qaGobiApiCbk.h, 1515
SetXTRAAutomaticDownload
qaGobiApiPds.h, 1678
SetXTRANetwork
qaGobiApiPds.h, 1679
SetupEventList
CatEventListTlv, 160
Severity

unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
severity
omaDmFotaTlv, 573
unpack_omaDmFotaTlv_t, 1087
Short Message Service (SMS), 42
shortName
nas_PLMNNetworkNameData, 483
nasPLMNNameResp, 548
PLMNNetworkNameData, 716
unpack_nas_SLQSGetPLMNName_t, 1059
shortNameCI
nasPLMNNameResp, 548
unpack_nas_SLQSGetPLMNName_t, 1059
shortNameEn
nasPLMNNameResp, 548
unpack_nas_SLQSGetPLMNName_t, 1060
shortNameLen
nas_PLMNNetworkNameData, 483
nasPLMNNameResp, 548
PLMNNetworkNameData, 716
unpack_nas_SLQSGetPLMNName_t, 1060
shortNameSB
nasPLMNNameResp, 548
unpack_nas_SLQSGetPLMNName_t, 1060
shortNameSpareBits
nas_PLMNNetworkNameData, 483
PLMNNetworkNameData, 717
sid
CDMAInfo, 164
nas_CDMAInfo, 414
nas_sidNid, 498
sidNid, 832
unpack_nas_GetHomeNetwork_t, 1048
SidNid
homeSIDNID, 304
nas_homeSIDNID, 445
sidNid, 831
nid, 832
sid, 832
SigInd
LibPackUMTSReqQoSSigInd, 357
UMTSReqQoSSigInd, 989
sigInfo, 832
nas_SccRxInfo, 496
pECIOThresh, 833
pHDRSINRThresh, 833
pIOThresh, 833
pLTESNRThresh, 833
pLTESigRptCfg, 833
pRSRPThresh, 833
pRSRQThresh, 833
pRSSIThresh, 833
pTDSCDMASINRCONFThresh, 833
SccRxInfo, 796
signal
signalInfo, 834
signalInfo, 833
alertPitch, 834
signal, 834
signalType, 834
signalStrength
nas_SignalStrengthTlv, 499
SignalStrengthDataType, 834
thresholds, 834
thresholdsSize, 834
signalStrengthReqMask
slqsSignalStrengthInfo, 849
unpack_nas_SLQSGetSignalStrength_t, 1063
signalType
signalInfo, 834
SimCapability
unpack_dms_GetDeviceCap_t, 994
simCapability
unpack_dms_GetDeviceCapabilities_t, 995
sinr
HDRSSInfo, 301
hdrSSInfo, 302
nas_SLQSSignalStrengthsInformation, 501
slqsSignalStrengthInfo, 850
SLQSSignalStrengthsInformation, 853
TDSCDMASigInfoExt, 916
tdscdmaSigInfoExt, 915
unpack_nas_SLQSGetSignalStrength_t, 1063
sinrDelta
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
sinrThresholdList
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
sinrThresholdListLen
nas_SLQSSignalStrengthsIndReq, 500
SLQSSignalStrengthsIndReq, 851
sku_str
slqsfwinfo_s, 841
unpack_dms_GetFirmwareInfo_t, 997
slot
pack_uim_SLQSUIMPowerDown_t, 654
pack_uim_SLQSUIMPowerUp_t, 655
UIMPowerDownReq, 967
UIMPowerUpReq, 967
slot_t, 834
bICCID, 835
bICCIDLength, 835
bLogicalSlot, 835
uPhyCardStatus, 835
uPhySlotStatus, 835
slotInf, 835
AppStatus, 836
cardState, 836
errorState, 836
numApp, 836
upinRetries, 836
upinState, 837
upukRetries, 837
SlotInfo

cardStatus, 156
uim_cardStatus, 940
slotInfo, 837
AppStatus, 838
cardState, 838
errorState, 838
numApp, 838
upinRetries, 838
upinState, 838
upukRetries, 838
slots_t, 838
uimSlotStatus, 838
slotsstatusChange
UIMSlotStatusChangeInfo, 977
unpack_uim_SetUimSlotStatusChangeCallback_-ind_t, 1131
slqs3GPPConfigItem
qaGobiApiWds.h, 1823
SlqsNas3GppNetworkInfo, 841
Description, 842
Forbidden, 842
InUse, 842
MCC, 842
MNC, 842
Preferred, 842
Roaming, 842
SlqsNas3GppNetworkRAT
qaGobiApiNas.h, 1625
SlqsNasPcsDigit, 842
includes_pcs_digit, 843
MCC, 843
MNC, 843
slqsNetworkScanInfo
qaGobiApiNas.h, 1625
SlqsProfile3GPP
unpackWdsProfileParam, 1187
wds_profileInfo, 1304
WdsProfileParam, 1320
SlqsProfile3GPP2
unpackWdsProfileParam, 1187
wds_profileInfo, 1304
WdsProfileParam, 1320
slqsSessionStateInfo, 846
pQmiInterfaceInfo, 847
reconfiguration_required, 847
sessionEndReason, 847
state, 847
slqsSignalStrengthInfo, 847
ecioList, 849
ecioListLen, 849
errorRateList, 849
errorRateListLen, 849
Io, 849
ltersrp, 849
ltesnr, 849
rsrqInfo, 849
rxSignalStrengthList, 849
rxSignalStrengthListLen, 849
signalStrengthReqMask, 849
sinr, 850
slqsWdsEventInfo, 853
pDataBearer, 854
pDormancyStatus, 854
pPacketsCountRX, 854
pPacketsCountTX, 854
pQmiInterfaceInfo, 854
pTotalBytesRX, 855
pTotalBytesTX, 855
slqsautoconnect, 838
acroamsetting, 839
acsetting, 839
action, 839
slqsfwinfo_s, 840
appversion_str, 841
bootversion_str, 841
carrier_str, 841
cur_carr_name, 841
cur_carr_rev, 841
modelid_str, 841
packageid_str, 841
priversion_str, 841
sku_str, 841
slqssendasyncsmsparams_s, 843
messageFormat, 845
messageSize, 845
pFollowOnDC, 845
pForceOnDC, 845
pLinktimer, 845
pMessage, 845
pRetryMessage, 845
pRetryMessageId, 845
pServiceOption, 845
pSmsOnIms, 845
pUserData, 845
slqssendsmsparams_s, 845
messageFailureCode, 846
messageFormat, 846
messageID, 846
messageSize, 846
pLinktimer, 846
pMessage, 846
pSmsOnIms, 846
sms.h
LITEQMI_QMI_CBK_PARAM_NOCHANGE, 1886
LITEQMI_QMI_CBK_PARAM_RESET, 1886
LITEQMI_QMI_CBK_PARAM_SET, 1886
sms.h, 1881
eqmiCbkSetStatus, 1886
MAX_MSE_TWS_MSG, 1884
MAX_SMS_LIST_SIZE, 1884
pack_sms_GetSMSCAddress, 1886
pack_sms_SLQSDeleteSMS, 1888
pack_sms_SLQSGetIndicationRegister, 1888
pack_sms_SLQSGetMessageWaiting, 1889
pack_sms_SLQSGetSMS, 1889
pack_sms_SLQSGetSMSList, 1890

pack_sms_SLQSGetSmsBroadcastConfig, 1889
pack_sms_SLQSGetTransLayerInfo, 1890
pack_sms_SLQSGetTransNWRegInfo, 1891
pack_sms_SLQSModifySMSStatus, 1891
pack_sms_SLQSSendAsyncSMS, 1891
pack_sms_SLQSSetIndicationRegister, 1892
pack_sms_SLQSSetSmsBroadcastActivation, 1892
pack_sms_SLQSSetSmsBroadcastConfig, 1892
pack_sms_SLQSSetSmsStorage, 1893
pack_sms_SLQSSmsGetMaxStorageSize, 1893
pack_sms_SLQSSmsGetMessageProtocol, 1893
pack_sms_SLQSSmsSetRoutes, 1894
pack_sms_SLQSSwiGetSMSStorage, 1894
pack_sms_SaveSMS, 1886
pack_sms_SendSMS, 1887
pack_sms_SetNewSMSCallback, 1887
pack_sms_SetSMSCAddress, 1888
SMS_CONFIG_LEN, 1884
SMS_MAX_SMS_ROUTES, 1884
SMS_NUM_OF_SET, 1884
SMSC_TYPE_LEN, 1884
sMSCAddressInfo, 1885
sMSEtwsMessageInfo, 1885
sMSEtwsPlmnInfo, 1885
sMSMTMessageInfo, 1885
sMSMessageModeInfo, 1885
sMSOnIMSInfo, 1885
sMSTransferRouteMTMessageInfo, 1886
unpack_sms_GetSMSCAddress, 1895
unpack_sms_SLQSDeleteSMS, 1897
unpack_sms_SLQSGetIndicationRegister, 1897
unpack_sms_SLQSGetMessageWaiting, 1897
unpack_sms_SLQSGetSMS, 1898
unpack_sms_SLQSGetSMSList, 1898
unpack_sms_SLQSGetSmsBroadcastConfig, 1898
unpack_sms_SLQSGetTransLayerInfo, 1899
unpack_sms_SLQSGetTransNWRegInfo, 1899
unpack_sms_SLQSModifySMSStatus, 1899
unpack_sms_SLQSSendAsyncSMS, 1900
unpack_sms_SLQSSetIndicationRegister, 1900
unpack_sms_SLQSSetSmsBroadcastActivation, 1900
unpack_sms_SLQSSetSmsBroadcastConfig, 1901
unpack_sms_SLQSSetSmsStorage, 1901
unpack_sms_SLQSSmsGetMaxStorageSize, 1901
unpack_sms_SLQSSmsGetMessageProtocol, 1902
unpack_sms_SLQSSmsSetRoutes, 1902
unpack_sms_SLQSSwiGetSMSStorage, 1902
unpack_sms_SLQSWmsMemoryFullCallBack_ind, 1903
unpack_sms_SaveSMS, 1895
unpack_sms_SendSMS, 1895
unpack_sms_SetNewSMSCallback, 1896
unpack_sms_SetNewSMSCallback_ind, 1896
unpack_sms_SetSMSCAddress, 1896
sms_BroadcastConfig, 855
  fromServiceId, 855
  selected, 855
  toServiceId, 855
sms_CDMABroadcastConfig, 855
  language, 856
  selected, 856
  serviceCategory, 856
sms_getIndicationReg, 856
  pRegCallStatInfoEvt, 857
  pRegTransLayerInfoEvt, 857
  pRegTransNWRegInfoEvt, 857
sms_getMsgWaitingInfo, 857
  msgWaitInfo, 857
  numInstances, 857
sms_getTransLayerInfo, 857
  pRegInd, 858
  pTransLayerInfo, 858
sms_getTransNWRegInfo, 858
  pRegStatus, 858
sms_maxStorageSizeReq, 859
  pMessageMode, 859
  storageType, 859
sms_maxStorageSizeResp, 859
  freeSlots, 860
  maxStorageSize, 860
sms_messageWaitingInfoContent, 860
  activeInd, 860
  msgCount, 860
  msgType, 860
sms_msgProtocolResp, 861
  msgProtocol, 861
sms_qaQmi3GPP2BroadcastCfgInfo, 861
  activated_ind, 862
  CDMABroadcastConfig, 862
  num_instances, 862
sms_qaQmi3GPPBroadcastCfgInfo, 862
  activated_ind, 862
  broadcastConfig, 862
  num_instances, 862
sms_routeEntry, 862
  messageClass, 863
  messageType, 863
  receiptAction, 863
  routeStorage, 863
sms_sendAsyncsmsParams, 863
  messageFormat, 865
  messageSize, 865
  pFollowOnDC, 865
  pForceOnDC, 865
  pLinktimer, 865
  pMessage, 865
  pRetryMessage, 865
  pRetryMessageId, 865
  pServiceOption, 865
  pSmsOnIms, 865
  pUserData, 865
sms_setIndicationReg, 866

pRegCallStatInfoEvt, 866
pRegTransLayerInfoEvt, 866
pRegTransNWRegInfoEvt, 866
sms_setRoutesReq, 866
numOfRoutes, 867
pTransferStatusReport, 867
routeList, 867
sms_transLayerInfo, 867
TransCap, 867
TransType, 868
smsEventType
SMSEventInfo_s, 875
smsMaxStorageSizeReq, 875
pMessageMode, 875
storageType, 875
smsMaxStorageSizeResp, 875
freeSlots, 876
maxStorageSize, 876
smsMsgprotocolResp, 877
msgProtocol, 878
smsOnIMS
SMSOnIMS, 879
sMSOnIMS, 879
smsRouteEntry, 880
messageClass, 881
messageType, 881
receiptAction, 881
routeStorage, 881
smsSetRoutesReq, 881
numOfRoutes, 881
pTransferStatusReport, 881
routeList, 881
smsStorage
pack_sms_SLQSSetSmsStorage_t, 642
snr
loc_satelliteInfo, 367
LTESSInfo, 399
lteSSInfo, 399
nas_SccRxInfo, 496
satelliteInfo, 795
SccRxInfo, 796
snrlevel
lteSnrinformation, 397
nas_lteSnrinformation, 460
soMask
DataBearerTech, 209
dataBearerTechnology, 211
qmiWDSDataBearerTechnology, 764
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
Source
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
source
_getResetInfoNotification, 66
altitudeSrcInfo, 120
altSrcInfo_t, 121
dmsSwiGetResetInfo, 227
unpack_dms_GetNetworkTime_t, 1000
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1007
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1008
sourceIPMask
LibPackTFTIDParams, 354
TFTIDParams, 920
SourceLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
spc
pack_dms_ResetToFactoryDefaults_t, 578
pack_nas_SetACCOLC_t, 606
pack_wds_SetMobileIPProfile_t, 686
Specific Absorption Rate (SAR), 48
spn
nas_serviceProviderName, 496
nasPLMNNameResp, 548
serviceProviderName, 802
unpack_nas_SLQSGetPLMNName_t, 1060
spnEncoding
nasPLMNNameResp, 548
unpack_nas_SLQSGetPLMNName_t, 1060
spnLength
nas_serviceProviderName, 496
nasPLMNNameResp, 548
serviceProviderName, 802
unpack_nas_SLQSGetPLMNName_t, 1060
srcPortRangeEnd
LibPackTFTIDParams, 354
TFTIDParams, 920
srcPortRangeStart
LibPackTFTIDParams, 354
TFTIDParams, 920
srvCapability
detailSvcInfo, 218
nas_detailSvcInfo, 428
nas_sysInfoCommon, 504
sysInfoCommon, 912
srvCapabilityValid
nas_sysInfoCommon, 504
sysInfoCommon, 912
srvDomain
nas_sysInfoCommon, 504
sysInfoCommon, 912
SrvDomainPref
NASServDomainPrefTlv, 550
srvDomainValid
nas_sysInfoCommon, 504
sysInfoCommon, 912
srvOption
arrSvcOption, 134
voice_arrSvcOption, 1204
srvStatus
detailSvcInfo, 219
GSMSrvStatusInfo, 291
nas_detailSvcInfo, 428
nas_GSMSrvStatusInfo, 437

nas_SrvStatusInfo, 502
SrvStatusInfo, 886
SrvStatusInfo, 885
isPrefDataPath, 886
srvStatus, 886
srxlev
cellParams, 176
gsmCellInfo, 289
nas_cellParams, 420
nas_gsmCellInfo, 436
nas_umtsLTENbrCell, 513
nas_wcdmaCellInfo, 515
umtsLTENbrCell, 983
wcdmaCellInfo, 1283
ssdatasession_params, 886
action, 888
failureReason, 888
failureReasonv4, 888
failureReasonv6, 888
instanceId, 888
ipfamily, 888
pAuthentication, 888
pPassword, 888
pProfileId3GPP, 888
pProfileId3GPP2, 888
pTechnology, 888
pUsername, 889
rcv4, 889
rcv6, 889
sessionId, 889
v4sessionId, 889
v6sessionId, 889
verbFailReason, 889
verbFailReasonType, 889
stage
uim_refreshevent, 951
UIMRefreshEvent, 971
StartPDSTrackingSessionExt
qaGobiApiPds.h, 1683
State
nas_NetworkStat1x, 470
nas_NetworkStatEVDO, 472
NetworkStat1x, 563
NetworkStatEVDO, 565
state
LocSetCradleMountReq, 380
omaDmConfigTlv, 569
omaDmConfigTlvExt, 571
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
pack_loc_SLQSLOCSetCradleMountConfig_t, 603
QosFlowInfoState, 772
QosMap, 773
slqsSessionStateInfo, 847
unpack_dms_GetActivationState_t, 992
unpack_dms_SLQSUIMGetState_t, 1020
unpack_omaDmConfigTlv_t, 1085
unpack_omaDmFotaTlv_t, 1087
unpack_qos_QosFlowInfoState_t, 1095
statmask
pack_wds_GetPacketStatus_t, 681
StatsMask
TransferStatInd, 928
transferStatInd, 928
statsMask
TrStatInd, 929
wds_TrStatInd, 1304
StatsPeriod
TransferStatInd, 928
transferStatInd, 928
statsPeriod
TrStatInd, 929
wds_TrStatInd, 1304
Status
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
status
delAssistDataStatus, 216
lteEARFCN, 384
ltePCI, 392
nas_lteEARFCN, 448
nas_ltePCI, 456
nas_wcdmaUARFCN, 521
pack_sms_SetNewSMSCallback_t, 636
QmiCbkLocBestAvailPosInd, 745
QmiCbkLocInjectPositionInd, 747
QmiCbkLocInjectUTCTimeInd, 750
QmiCbkLocSetExtPowerConfigInd, 757
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_DeleteAssistData_Ind_t, 1033
unpack_loc_SetExtPowerConfig_Ind_t, 1042
unpack_loc_SetOperationMode_Ind_t, 1043
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1099
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1100
wcdmaUARFCN, 1293
statusChange
UIMStatusChangeInfo, 977
StopPDSTrackingSession
qaGobiApiPds.h, 1684
storageIndex
FMSImageIdElement, 245
ImageIdElement, 307
storageType
pack_sms_SaveSMS_t, 635
pack_sms_SLQSDeleteSMS_t, 637
pack_sms_SLQSGetSMS_t, 638
pack_sms_SLQSGetSMSList_t, 639
pack_sms_SLQSModifySMSStatus_t, 640
sms_maxStorageSizeReq, 859
smsMaxStorageSizeReq, 875
SMSMemoryInfo, 876
SMSMTMessage, 878
sMSMTMessage, 878

unpack_sms_SLQSWmsMemoryFullCallBack_ind-_t, 1120
String
unpack_dms_GetDeviceHardwareRev_t, 995
unpack_dms_GetDeviceMfr_t, 995
unpack_dms_GetFSN_t, 998
unpack_dms_UIMGetICCID_t, 1021
stringSize
unpack_dms_GetDeviceHardwareRev_t, 995
unpack_dms_GetDeviceMfr_t, 995
unpack_dms_UIMGetICCID_t, 1022
subAddr
calledPartySubAdd, 146
voice_calledPartySubAdd, 1206
subAddrLen
calledPartySubAdd, 146
voice_calledPartySubAdd, 1206
subAddrType
calledPartySubAdd, 146
voice_calledPartySubAdd, 1206
subnetMask
IPv4Addr, 326
unpack_qos_IPv4Addr_t, 1091
SubnetMaskV4
unpack_wds_SLQSGetRuntimeSettings_t, 1180
subsType
pack_voice_SLQSVoiceBindSubscription_t, 663
voiceBindSubscriptionInfo, 1232
SupUSBComps
unpack_dms_GetUSBComp_t, 1003
SuppOA
CUGInfo, 194
voice_CUGInfo, 1218
SuppPrefCUG
CUGInfo, 194
voice_CUGInfo, 1218
supportedMsgLen
SupportedMsgList, 889
SupportedMsgList, 889
supportedMsgLen, 889
supportedMsgs, 889
supportedMsgs
SupportedMsgList, 889
svInfoMask
loc_satelliteInfo, 367
satelliteInfo, 795
svListLen
loc_satelliteInfo, 367
satelliteInfo, 795
svStatus
loc_satelliteInfo, 367
satelliteInfo, 795
svUsedforFix
qaGobiApiCbk.h, 1473
svUsedforFix_s, 892
gnssSvUsedList, 892
gnssSvUsedList_len, 892
svc
pack_qmi_t, 631
SvcClass
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1209
voice_callFWInfo, 1210
SvcStatus
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1209
voice_callFWInfo, 1210
svcType
ccSUPSType, 161
SUPSInfo, 890
voice_ccSUPSType, 1212
sw1
cardResult, 155
uim_cardResult, 939
sw2
cardResult, 155
uim_cardResult, 939
swVerString
pack_dms_SLQSSwiSetHostDevInfo_t, 584
unpack_dms_SLQSSwiGetHostDevInfo_t, 1017
SwiAvmsEventInd
qaGobiApiCbk.h, 1473
SwiDataTypes.h, 1903
BOOL, 1904
BYTE, 1904
CHAR, 1904
FLOAT, 1904
INT32, 1904
INT8, 1904
LPCSTR, 1904
SHORT, 1904
SWI_API, 1904
ULONG, 1904
ULONGLONG, 1904
UNUSEDPARAM, 1904
USHORT, 1904
WORD, 1904
SwiLocGetAutoStart
qaGobiApiLoc.h, 1618
SwiLocGetAutoStartResp, 893
fix_rate, 895
fix_rate_reported, 895
fix_type, 895
fix_type_reported, 895
function, 895
function_reported, 895
max_dist, 895
max_dist_reported, 895
max_time, 895
max_time_reported, 895
SwiLocSetAutoStart
qaGobiApiLoc.h, 1619
SwiLocSetAutoStartReq, 895
fix_rate, 896

fix_type, 896
function, 896
max_dist, 896
max_time, 896
set_fix_rate, 896
set_fix_type, 897
set_function, 897
set_max_dist, 897
set_max_time, 897
swiModemStatusResp, 897
commonInfo, 897
pLTEInfo, 897
SwiOTAMsg
qaGobiApiCbk.h, 1473
SwiOTAMsg_s, 897
data, 898
data_len, 898
pLteNasRelInfo, 898
pTime, 898
type, 898
swiPDPRuntimeSettingsReq, 898
contextId, 899
contextType, 899
swiPDPRuntimeSettingsResp, 899
pAPNName, 901
pBearerId, 901
pContextId, 901
pIPv4Address, 901
pIPv4GWAddress, 901
pIPv6Address, 901
pIPv6GWAddress, 901
pPrDNSIPv4Address, 901
pPrDNSIPv6Address, 901
pPrPCSCFIPv4Address, 901
pPrPCSCFIPv6Address, 901
pSeDNSIPv4Address, 901
pSeDNSIPv6Address, 901
pSePCSCFIPv4Address, 901
pSePCSCFIPv6Address, 901
swiQosFilter, 901
index, 903
pEspSpi, 903
pIPv4DstAddr, 903
pIPv4SrcAddr, 903
pIPv6DstAddr, 903
pIPv6Label, 903
pIPv6SrcAddr, 903
pIPv6TrafCls, 903
pId, 903
pNxtHdrProto, 903
pPrecedence, 903
pTCPDstPort, 903
pTCPSrcPort, 903
pTos, 903
pTranDstPort, 903
pTranSrcPort, 903
pUDPDstPort, 903
pUDPSrcPort, 903
version, 904
swiQosFlow, 904
index, 906
p3GPP2Pri, 906
p3GPPImCn, 906
p3GPPResResidualBER, 906
p3GPPSigInd, 906
p3GPPTraHdlPri, 906
pDataRate, 906
pJitter, 906
pLatency, 906
pLteQci, 906
pMaxAllowedPktSz, 906
pMinPolicedPktSz, 907
pPktErrRate, 907
pProfileId3GPP2, 907
pTokenBucket, 907
pTrafficClass, 907
swiQosGranted, 907
pRxFlow, 907
pTxFlow, 907
swiQosIds, 907
pIds, 907
sz, 908
swiQosModifyReq, 908
id, 908
pRxFilter, 908
pRxFlow, 908
pTxFilter, 908
pTxFlow, 908
swiQosReq, 908
index, 909
pRxFilter, 909
pRxFlow, 909
pTxFilter, 909
pTxFlow, 909
swiRMTrasnferStaticsReq, 909
bResetStatistics, 910
ulMask, 910
swiloc.h, 1904
pack_swiloc_SwiLocGetAutoStart, 1905
pack_swiloc_SwiLocSetAutoStart, 1905
unpack_swiloc_SwiLocGetAutoStart, 1905
unpack_swiloc_SwiLocSetAutoStart, 1906
swioma.h, 1906
pack_swioma_SLQSOMADMAlertCallback, 1907
pack_swioma_SLQSOMADMCancelSession, 1908
pack_swioma_SLQSOMADMGetSessionInfo, 1908
pack_swioma_SLQSOMADMGetSettings, 1909
pack_swioma_SLQSOMADMSendSelection, 1909
pack_swioma_SLQSOMADMSetSettings, 1910
pack_swioma_SLQSOMADMStartSession, 1911
unpack_swioma_SLQSOMADMAlertCallback, 1911
unpack_swioma_SLQSOMADMAlertCallback_ind, 1912

unpack_swioma_SLQSOMADMCancelSession, 1912
unpack_swioma_SLQSOMADMGetSessionInfo, 1913
unpack_swioma_SLQSOMADMGetSettings, 1913
unpack_swioma_SLQSOMADMSendSelection, 1914
unpack_swioma_SLQSOMADMSetSettings, 1914
unpack_swioma_SLQSOMADMStartSession, 1914
switchOption
pack_voice_SLQSVoiceALSSetLineSwitching_t, 662
voiceALSSetLineSwitchInfo, 1231
sysInfoCDMA
CDMASysInfo, 174
nas_CDMASysInfo, 419
sysInfoCommon, 910
isSysForbidden, 912
isSysForbiddenValid, 912
roamStatus, 912
roamStatusValid, 912
srvCapability, 912
srvCapabilityValid, 912
srvDomain, 912
srvDomainValid, 912
sysInfoGSM
GSMSysInfo, 294
nas_GSMSysInfo, 440
sysInfoHDR
HDRSysInfo, 304
nas_HDRSysInfo, 445
sysInfoLTE
LTESysInfo, 401
nas_LTESysInfo, 464
sysInfoWCDMA
nas_WCDMASysInfo, 521
WCDMASysInfo, 1292
sysSelectPrefInfo
qaGobiApiNas.h, 1626
sysSelectPrefParams
qaGobiApiNas.h, 1630
system
loc_satelliteInfo, 367
loc_SV, 369
satelliteInfo, 795
SV, 891
SystemID
qaQmiServingSystemParam, 739
unpack_nas_SLQSGetServingSystem_t, 1062
systemID
CDMASysInfo, 174
nas_CDMASysInfo, 419
systemMode
CommInfo, 184
nas_CommInfo, 422
sz
swiQosIds, 908
szAPN
_QmiSwiAvmsSetSettingsAPNInfo, 80
szAlertMsg
_QmiSwiAvmsEventReportConfigResp, 74
szDescription
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
szName
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
szPWD
_QmiSwiAvmsSetSettingsAPNInfo, 80
szUname
_QmiSwiAvmsSetSettingsAPNInfo, 80
szVersion
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73

t3396_apn
nasTimers, 559
unpack_nas_SLQSNasTimerCallback_ind_t, 1077
t3396_plmn_id
nasTimers, 559
unpack_nas_SLQSNasTimerCallback_ind_t, 1077
t3396_val
nasTimers, 559
unpack_nas_SLQSNasTimerCallback_ind_t, 1077
t_Sv, 913
entries, 913
len, 913
t_gpsTime, 912
gpsTimeOfWeekMs, 912
gpsWeek, 912
t_sensor, 912
aidingIndicatorMask, 912
usageMask, 912
TCPDstPort
unpack_qos_swiQosFilter_t, 1104
TCPSrcPort
unpack_qos_swiQosFilter_t, 1104
TDSCDMAECIOThresh, 913
TDSCDMAECIOThreshListLen
nas_TDSCDMAECIOThresh, 504
TDSCDMAECIOThresh, 913
TDSCDMARSCPThresh, 913
TDSCDMARSCPThreshListLen
nas_TDSCDMARSCPThresh, 505
TDSCDMARSCPThresh, 914
TDSCDMARSSIThresh, 914
TDSCDMARSSIThreshListLen
nas_TDSCDMARSSIThresh, 505
TDSCDMARSSIThresh, 914
TDSCDMASINRCONFThresh, 916
TDSCDMASINRThresh, 916
TDSCDMASINRThreshListLen
nas_TDSCDMASINRThresh, 506
TDSCDMASINRThresh, 917
TDSCDMASigInfoExt, 915
ecio, 915

rscp, 915
rssi, 915
sinr, 916
tFNASwiLTECphyCaInfo
qaGobiApiCbk.h, 1474
tFNASwiOTAMsg
qaGobiApiCbk.h, 1475
tFNActivationStatus
qaGobiApiCbk.h, 1474
tFNAllCallStatus
qaGobiApiCbk.h, 1474
tFNAsyncRawSend
qaGobiApiCbk.h, 1475
tFNBandPreference
qaGobiApiCbk.h, 1475
tFNBestAvailPos
qaGobiApiCbk.h, 1476
tFNCATEvent
qaGobiApiCbk.h, 1477
tFNCbkUimSlotStatusChangeInd
qaGobiApiCbk.h, 1477
tFNDHCPv4ClientLeaseStatus
qaGobiApiCbk.h, 1479
tFNDTMFEvent
qaGobiApiCbk.h, 1479
tFNDUNCallInfo
qaGobiApiCbk.h, 1479
tFNDataCapabilities
qaGobiApiCbk.h, 1477
tFNDataSysStatus
qaGobiApiCbk.h, 1478
tFNDelAssistData
qaGobiApiCbk.h, 1478
tFNDeviceStateChange
qaGobiApiCbk.h, 1478
tFNEventPosition
qaGobiApiCbk.h, 1479
tFNFwDldCompletion
qaGobiApiCbk.h, 1479
tFNGnssSvInfo
qaGobiApiCbk.h, 1480
tFNHDRPersonaity
qaGobiApiCbk.h, 1480
tFNImsRegMgrConfig
qaGobiApiCbk.h, 1481
tFNImsSIPConfig
qaGobiApiCbk.h, 1481
tFNImsSMSConfig
qaGobiApiCbk.h, 1482
tFNImsUserConfig
qaGobiApiCbk.h, 1482
tFNImsVoIPConfig
qaGobiApiCbk.h, 1482
tFNImsaPdpStatus
qaGobiApiCbk.h, 1480
tFNImsaRatStatus
qaGobiApiCbk.h, 1481
tFNImsaRegStatus
qaGobiApiCbk.h, 1481
tFNImsaSvcStatus
qaGobiApiCbk.h, 1481
tFNInfoRec
qaGobiApiCbk.h, 1482
tFNInjectPosition
qaGobiApiCbk.h, 1482
tFNInjectSensorData
qaGobiApiCbk.h, 1483
tFNInjectTimeStatus
qaGobiApiCbk.h, 1483
tFNInjectUTCTime
qaGobiApiCbk.h, 1483
tFNLUReject
qaGobiApiCbk.h, 1483
tFNMemoryFull
qaGobiApiCbk.h, 1485
tFNMessageWaiting
qaGobiApiCbk.h, 1485
tFNMitiLvlRpt
qaGobiApiCbk.h, 1485
tFNMobileIPStatus
qaGobiApiCbk.h, 1485
tFNModemTempInfo
qaGobiApiCbk.h, 1485
tFNNasTimer
qaGobiApiCbk.h, 1486
tFNNet
qaGobiApiCbk.h, 1486
tFNNetworkReject
qaGobiApiCbk.h, 1486
tFNNetworkTime
qaGobiApiCbk.h, 1486
tFNNewGPS
qaGobiApiCbk.h, 1487
tFNNewNMEA
qaGobiApiCbk.h, 1487
tFNNewRMTransferStatistics
qaGobiApiCbk.h, 1487
tFNNewSMS
qaGobiApiCbk.h, 1488
tFNOMADMState
qaGobiApiCbk.h, 1488
tFNOTASPStatus
qaGobiApiCbk.h, 1490
tFNOpMode
qaGobiApiCbk.h, 1489
tFNPDSState
qaGobiApiCbk.h, 1490
tFNPacketSrvState
qaGobiApiCbk.h, 1490
tFNPower
qaGobiApiCbk.h, 1491
tFNPrivacyChange
qaGobiApiCbk.h, 1491
tFNQosNWStatus
qaGobiApiCbk.h, 1491
tFNQosPriEvent

qaGobiApiCbk.h, 1492
tFNQosStatus
qaGobiApiCbk.h, 1492
tFNRFInfo
qaGobiApiCbk.h, 1493
tFNRankIndicator
qaGobiApiCbk.h, 1493
tFNResetInfo
qaGobiApiCbk.h, 1493
tFNRoamingIndicator
qaGobiApiCbk.h, 1494
tFNSDKTerminated
qaGobiApiCbk.h, 1494
tFNSLQSOMADMAlert
qaGobiApiCbk.h, 1496
tFNSLQSQOSEvent
qaGobiApiCbk.h, 1496
tFNSLQSSessionState
qaGobiApiCbk.h, 1496
tFNSLQSSignalStrengths
qaGobiApiCbk.h, 1497
tFNSLQSWDSEvent
qaGobiApiCbk.h, 1497
tFNSMSEvents
qaGobiApiCbk.h, 1497
tFNSUPSInfo
qaGobiApiCbk.h, 1497
tFNSUPSNotification
qaGobiApiCbk.h, 1497
tFNSensorStreaming
qaGobiApiCbk.h, 1494
tFNServingSystem
qaGobiApiCbk.h, 1494
tFNSetCradleMount
qaGobiApiCbk.h, 1495
tFNSetEngineState
qaGobiApiCbk.h, 1495
tFNSetEventTimeSync
qaGobiApiCbk.h, 1495
tFNSetExtPowerConfig
qaGobiApiCbk.h, 1495
tFNSigInfo
qaGobiApiCbk.h, 1495
tFNSignalStrength
qaGobiApiCbk.h, 1496
tFNSwiAvmEventReport
qaGobiApiCbk.h, 1498
tFNSysInfo
qaGobiApiCbk.h, 1498
tFNSysSelectionPref
qaGobiApiCbk.h, 1498
tFNUIMRefresh
qaGobiApiCbk.h, 1499
tFNUIMStatusChangeInfo
qaGobiApiCbk.h, 1499
tFNUSSDNoWaitIndication
qaGobiApiCbk.h, 1499
tFNUSSDNotification
qaGobiApiCbk.h, 1499
tFNUSSDRelease
qaGobiApiCbk.h, 1500
tFNtransLayerInfo
qaGobiApiCbk.h, 1498
tFNtransNWRegInfo
qaGobiApiCbk.h, 1498
TFTIDParams, 919
destPortRangeEnd, 920
destPortRangeStart, 920
eValid, 920
filterId, 920
flowLabel, 920
IPSECSPI, 920
ipVersion, 920
nextHeader, 920
pSourceIP, 920
sourceIPMask, 920
srcPortRangeEnd, 920
srcPortRangeStart, 920
tosMask, 920
THIRD_INSTANCE
qaGobiApiCbk.h, 1463
TIME_DATE_BUF
qaGobiApiSms.h, 1696
TIME_STAMP_BUF
qaGobiApiSms.h, 1696
TPCause
SMSAsyncRawSend_s, 869
TRMessageTlv
unpack_sms_SetNewSMSCallback_ind_t, 1115
TX_PWR
nas_NetworkStat1x, 470
NetworkStat1x, 563
TXAGCList, 930
pTXAIG, 931
pTXComprSlope, 931
pTXComprThres, 931
pTXExpSlope, 931
pTXExpThres, 931
pTXStaticGain, 931
TXChan
LTEInfo, 387
nas_LTEInfo, 451
tXDroppedCount
unpack_wds_GetPacketStatus_t, 1171
TXOKBytesCount
DUNCallInfoInd, 231
tXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1171
tXOkBytesCount
unpack_wds_GetPacketStatus_t, 1171
TXPCMIIRFltr, 932
pFlag, 933
pStage0Val, 933
pStage1Val, 933
pStage2Val, 933
pStage3Val, 933

pStage4Val, 933
pStageCnt, 933
tXPacketErrors
unpack_wds_GetPacketStatus_t, 1171
tXPacketOverflows
unpack_wds_GetPacketStatus_t, 1171
tXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1171
Tables, 61
tac
LTEInfoIntrafreq, 389
LTESysInfo, 401
nas_LTEInfoIntrafreq, 454
nas_LTESysInfo, 464
tacValid
LTESysInfo, 402
nas_LTESysInfo, 464
TdsBandCapability
unpack_dms_SLQSGetBandCapability_t, 1012
tdscdmaSigInfoExt, 914
ecio, 915
rscp, 915
rssi, 915
sinr, 915
tech
NWProfile, 569
techName
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1181
techType
DataBearerTech, 209
Technology
DeviceConfigDetail, 220
fwinfo_s, 248
nas_DeviceConfigDetail, 429
unpack_wds_SLQSGetRuntimeSettings_t, 1180
TechnologyPref
pack_nas_SetNetworkPreference_t, 608
tempData_t, 917
temperature, 917
temperatureDataLen, 918
timeOfFirstSample, 918
timeOffset, 918
timeSource, 918
temperature
CommInfo, 184
nas_CommInfo, 422
tempData_t, 917
tempratureData, 919
temperatureDataLen
tempData_t, 918
tempratureData, 919
tempratureData, 918
temperature, 919
temperatureDataLen, 919
timeOfFirstSample, 919
timeOffset, 919
timeSource, 919
textMsgLength
cdmaMsgEncodingParams, 169
Thermal Mitigation Device(TMD), 58
threshGsmHigh
lteGsmCellInfo, 385
nas_lteGsmCellInfo, 449
threshGsmLow
lteGsmCellInfo, 385
nas_lteGsmCellInfo, 449
threshServingLow
LTEInfoIntrafreq, 389
nas_LTEInfoIntrafreq, 454
threshXHigh
infoInterFreq, 325
nas_infoInterFreq, 447
threshXLow
infoInterFreq, 325
nas_infoInterFreq, 447
threshXhigh
lteWcdmaCellInfo, 402
nas_lteWcdmaCellInfo, 465
threshXlow
lteWcdmaCellInfo, 403
nas_lteWcdmaCellInfo, 465
thresholds
SignalStrengthDataType, 834
thresholdsSize
SignalStrengthDataType, 834
Time
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
wcdmaLongMsgDecodingParams, 1286
wcdmaMsgDecodingParams, 1287
time
NASTimeInfoTlv, 558
Time_uncert_ms
GPSStateInfo, 288
timeInfo, 921
day, 922
dayLtSavingAdj, 922
dayOfWeek, 922
hour, 922
minute, 922
month, 922
radioInterface, 922
second, 922
timeZone, 922
TlvPresent, 922
year, 922
TimeLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
timeMsec
pack_loc_SLQSLOCInjectUTCTime_t, 603
timeOfFirstSample
sensorData, 798
sensorData_t, 799

tempData_t, 918
tempratureData, 919
timeOffset
sensorData, 798
sensorData_t, 799
tempData_t, 918
tempratureData, 919
timeSource
tempData_t, 918
tempratureData, 919
TimeStmp_gps_week
GPSStateInfo, 288
TimeStmp_tow_ms
GPSStateInfo, 288
timeSyncRefCounter
QmiCbkLocEventTimeSyncInd, 747
timeTlv
NASQmiCbkNasSwiOTAMessageInd, 549
timeUncMsec
pack_loc_SLQSLOCInjectUTCTime_t, 603
timeZone
nas_timeInfo, 508
timeInfo, 922
timeout
pack_pds_SetPDSDefaults_t, 630
pack_qmi_t, 631
Timers
_QmiSwiAvmsSetSettingsConnectionRetryTimers, 80
_SLQSAVMSSettingsConnectionRetryTimers, 86
timestamp
unpack_dms_GetNetworkTime_t, 1000
timestampAge
pack_loc_SLQSLOCInjectPosition_t, 600
timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 600
timingAdvance
GERANInfo, 250
nas_GERANInfo, 434
TlvPresent
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
_QmiSwiAvmsEventReportConnectionRequestResp, 75
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
_QmiSwiAvmsEventReportHTTPStatusResp, 76
_QmiSwiAvmsEventReportNotificationResp, 77
_QmiSwiAvmsEventReportPackageIDResp, 78
_QmiSwiAvmsEventReportRegStatusResp, 78
_QmiSwiAvmsEventReportSessionTypeResp, 79
_QmiSwiAvmsEventReportWAMSParamChangeResp, 79
CatCommonEventTlv, 158
CsgId, 192
DataULongLongTlv, 215
DataULongTlv, 215
dms_ActivationStatusTlv, 222
dms_OperatingModeTlv, 223
eTWSPLMNInfoTlv, 237
messageModeTlv, 403
nas_PhyCaAggPcellInfo, 476
nas_PhyCaAggScellDlBw, 476
nas_PhyCaAggScellIndex, 477
nas_PhyCaAggScellIndType, 477
nas_PhyCaAggScellInfo, 480
nas_RejectReasonTlv, 488
nas_RFInfoTlv, 488
nas_SccRxInfo, 496
nas_SignalStrengthTlv, 499
nas_SLQSSignalStrengthsTlv, 501
nas_timeInfo, 508
NASBandPreferenceTlv, 521
NASEmergencyModeTlv, 523
NASGWAcqOrderPrefTlv, 531
NASLTEBandPreferenceTlv, 536
NASLteNasReleaseInfoTlv, 536
NASModePreferenceTlv, 536
NASNetSelPreferenceTlv, 536
NASOTAMessageTlv, 541
NASPhyCaAggPcellInfo, 542
NASPhyCaAggScellDlBw, 543
NASPhyCaAggScellIndex, 543
NASPhyCaAggScellIndType, 544
NASPhyCaAggScellInfo, 545
NASPRLPreferenceTlv, 548
NASRoamPreferenceTlv, 550
NASServDomainPrefTlv, 550
NASTimeInfoTlv, 558
newMTMessageTlv, 566
PhyCaAggPcellInfo, 708
PhyCaAggScellDlBw, 708
PhyCaAggScellIndex, 709
PhyCaAggScellIndType, 709
PhyCaAggScellInfo, 712
PlmnID, 714
RoamingInfo, 782
SccRxInfo, 796
sessionInfoTlv, 806
sessionInfoTlvExt, 807
sMSCAddressTlv, 871
sMSEtwsMessageTlv, 872
sMSOnIMSTlv, 880
timeInfo, 922
transferRouteMessageTlv, 927
wds_DataULongLongTlv, 1294
wds_DataULongTlv, 1295
wds_DHCPLeaseOptTlv, 1295
wds_DHCPLeaseStateTlv, 1295
wds_DHCPProfileIdTlv, 1296
wds_IPv4AdTlv, 1300
Tlvresult
pack_dms_GetCustFeaturesV2_t, 578
pack_dms_SetCustFeaturesV2_t, 581
pack_dms_SetPower_t, 581

pack_dms_SetUSBComp_t, 581
pack_dms_UIMGetICCID_t, 586
pack_fms_GetImagesPreference_t, 589
pack_fms_GetStoredImages_t, 590
pack_fms_SetImagesPreference_t, 590
pack_loc_Delete_Assist_Data_t, 591
pack_loc_EventRegister_t, 594
pack_loc_SetExtPowerState_t, 594
pack_loc_SetOperationMode_t, 595
pack_loc_SLQSLOCGetBestAvailPos_t, 595
pack_loc_Start_t, 605
pack_loc_Stop_t, 605
pack_nas_SetNetworkPreference_t, 608
pack_uim_ChangePin_t, 649
pack_uim_ReadTransparent_t, 650
pack_uim_SetPinProtection_t, 651
pack_uim_UnblockPin_t, 658
pack_uim_VerifyPin_t, 660
unpack_dms_GetBandCapability_t, 992
unpack_dms_GetCrashAction_t, 992
unpack_dms_GetCustFeature_t, 993
unpack_dms_GetCustFeaturesV2_t, 993
unpack_dms_GetDeviceCap_t, 994
unpack_dms_GetDeviceHardwareRev_t, 995
unpack_dms_GetDeviceMfr_t, 995
unpack_dms_GetDeviceSerialNumbers_t, 996
unpack_dms_GetFirmwareInfo_t, 997
unpack_dms_GetFirmwareRevision_t, 997
unpack_dms_GetFirmwareRevisions_t, 998
unpack_dms_GetFSN_t, 998
unpack_dms_GetIMSI_t, 999
unpack_dms_GetManufacturer_t, 999
unpack_dms_GetModelID_t, 1000
unpack_dms_GetNetworkTime_t, 1000
unpack_dms_GetOfflineReason_t, 1001
unpack_dms_GetPower_t, 1002
unpack_dms_GetPRLVersion_t, 1002
unpack_dms_GetUSBComp_t, 1003
unpack_dms_GetVoiceNumber_t, 1003
unpack_dms_ResetToFactoryDefaults_t, 1004
unpack_dms_SetActivationStatusCallback_t, 1004
unpack_dms_SetCustFeature_t, 1005
unpack_dms_SetCustFeaturesV2_t, 1005
unpack_dms_SetEventReport_ind_t, 1006
unpack_dms_SetEventReport_t, 1006
unpack_dms_SetFirmwarePreference_t, 1006
unpack_dms_SetPower_t, 1006
unpack_dms_SetUSBComp_t, 1007
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1007
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1008
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1009
unpack_dms_SLQSGetERIFile_t, 1012
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 1013
unpack_dms_SLQSSwiGetCrashInfo_t, 1013
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1014
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1014
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1016
unpack_dms_SLQSSwiGetHostDevInfo_t, 1017
unpack_dms_SLQSSwiGetOSInfo_t, 1018
unpack_dms_SLQSSwiGetSerialNoExt_t, 1018
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1019
unpack_dms_SLQSSwiSetHostDevInfo_t, 1019
unpack_dms_SLQSSwiSetOSInfo_t, 1020
unpack_dms_SLQSUIMGetState_t, 1020
unpack_dms_UIMGetControlKeyStatus_t, 1021
unpack_dms_UIMGetICCID_t, 1022
unpack_dms_UIMGetPINStatus_t, 1023
unpack_dms_UIMSetControlKeyProtection_t, 1024
unpack_dms_UIMSetPINProtection_t, 1024
unpack_dms_UIMUnblockControlKey_t, 1025
unpack_fms_GetImagesPreference_t, 1025
unpack_fms_GetStoredImages_t, 1026
unpack_fms_SetImagesPreference_t, 1027
unpack_loc_BestAvailPos_Ind_t, 1032
unpack_loc_Delete_Assist_Data_t, 1033
unpack_loc_DeleteAssistData_Ind_t, 1034
unpack_loc_EngineState_Ind_t, 1034
unpack_loc_EventRegister_t, 1034
unpack_loc_GnssSvInfo_Ind_t, 1035
unpack_loc_PositionRpt_Ind_t, 1041
unpack_loc_SetExtPowerConfig_Ind_t, 1042
unpack_loc_SetExtPowerState_t, 1043
unpack_loc_SetOperationMode_Ind_t, 1043
unpack_loc_SetOperationMode_t, 1044
unpack_loc_SLQSLOCGetBestAvailPos_t, 1044
unpack_loc_Start_t, 1044
unpack_loc_Stop_t, 1045
unpack_nas_GetNetworkPreference_t, 1049
unpack_nas_SetNetworkPreference_t, 1055
unpack_nas_SetServingSystemCallback_ind_t, 1056
unpack_nas_SlqsGetLTECphyCAInfo_t, 1057
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1076
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1078
unpack_uim_ChangePin_t, 1128
unpack_uim_GetCardStatus_t, 1129
unpack_uim_ReadTransparent_t, 1129
unpack_uim_SetPinProtection_t, 1130
unpack_uim_UnblockPin_t, 1136
unpack_uim_VerifyPin_t, 1136
unpack_wds_SLQSCreateProfile_t, 1173
unpack_wds_SLQSGetProfileSettings_t, 1178
unpack_wds_SLQSSetIPFamilyPreference_t, 1180
TmdDeRegNotMitigationLvlReq, 922
mitigationDevID, 923
mitigationDevIDLen, 923
TmdGetMitigationDevListResp, 923
pMitigationDevList, 923

pMitigationDevListLen, 923
TmdGetMitigationLvlReq, 924
mitigationDevID, 924
mitigationDevIDLen, 924
TmdGetMitigationLvlResp, 924
pCurrentmitigationLvl, 925
pReqMitigationLvl, 925
TmdMitigationLvlIndReq, 925
mitigationDevID, 925
mitigationDevIDLen, 925
TmdRegNotMitigationLvlReq, 925
mitigationDevID, 926
mitigationDevIDLen, 926
toServiceId
BroadcastConfig, 142
sms_BroadcastConfig, 855
toggleMode
lineCtrlInfo, 358
TokenBucket
unpack_qos_swiQosFlow_t, 1108
tokenBucket, 926
bucketSz, 926
peakRate, 926
tokenRate, 926
tokenRate
tokenBucket, 926
unpack_qos_tokenBucket_t, 1108
Tos, 926
mask, 927
val, 927
tosMask
LibPackTFTIDParams, 354
TFTIDParams, 920
total_rx_bytes
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
total_rx_pkt
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
total_tx_bytes
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
total_tx_bytes_drp
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
total_tx_pkt
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
total_tx_pkt_drp
sQosStat, 885
unpack_qos_SLQSQosSwiReadDataStats_t, 1098
TrStatInd, 928
statsMask, 929
statsPeriod, 929
TrackAreaCode
unpack_nas_SLQSGetServingSystem_t, 1062
trackAreaCode
qaQmiServingSystemParam, 739
TrafficClass
unpack_qos_swiQosFlow_t, 1108
trafficClass
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
trafficPriority
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
TranDstPort
unpack_qos_swiQosFilter_t, 1104
TranSrcPort
unpack_qos_swiQosFilter_t, 1104
TransCap
_transLayerinfo, 111
sms_transLayerInfo, 867
transLayerInfo
qaGobiApiSms.h, 1699
transLayerNotification
qaGobiApiCbk.h, 1500
transNWRegInfoNotification
qaGobiApiCbk.h, 1500
TransType
_transLayerinfo, 111
sms_transLayerInfo, 868
transactionID
SMSTransferRouteMTMessage, 882
sMSTransferRouteMTMessage, 883
transferDelay
LibPackUMTSQoS, 356
UMTSMinQoS, 986
UMTSQoS, 988
wds_UMTSMinQoS, 1307
TransferRouteMTMessageInfo
transferRouteMessageTlv, 927
transferRouteMessageTlv, 927
TlvPresent, 927
TransferRouteMTMessageInfo, 927
TransferStatInd, 927
StatsMask, 928
StatsPeriod, 928
transferStatInd, 928
StatsMask, 928
StatsPeriod, 928
transferStats
pack_wds_SLQSSetWdsEventCallback_t, 693
TransferStatsDataType, 928
interval, 928
trueIMSI, 929
imsiT1112, 930
imsiTS1, 930
imsiTS2, 930
imsiTaddrNum, 930
mccT, 930
trueSrvStatus

GSMSrvStatusInfo, 291
nas_GSMSrvStatusInfo, 437
tx_bytes
NetStats, 560
sQosFlowStat, 884
unpack_QosFlowStat_t, 1110
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
tx_bytes_drp
sQosFlowStat, 884
unpack_QosFlowStat_t, 1110
tx_errors
NetStats, 561
tx_overflows
NetStats, 561
tx_packets
NetStats, 561
tx_pkt
sQosFlowStat, 884
unpack_QosFlowStat_t, 1110
tx_pkt_drp
sQosFlowStat, 884
unpack_QosFlowStat_t, 1110
tx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
TxDropConutTlv
QmiCbkWdsStatisticsIndState, 760
unpack_RMTransferStatistics_ind_t, 1111
txInfo, 931
isInTraffic, 931
txPower, 932
txOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1177
TxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 760
unpack_RMTransferStatistics_ind_t, 1111
TxOkConutTlv
QmiCbkWdsStatisticsIndState, 760
unpack_RMTransferStatistics_ind_t, 1111
txPower
nas_txInfo, 510
txInfo, 932
TxQFilter
unpack_qos_QosFlowInfo_t, 1094
TxQFlowGranted
unpack_qos_QosFlowInfo_t, 1094
type
_getResetInfoNotification, 66
dmsSwiGetResetInfo, 228
pack_wds_GetDefaultProfileNum_t, 679
pack_wds_SetDefaultProfileNum_t, 683
SwiOTAMsg_s, 898
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1007
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1008
unpack_qmi_t, 1090
typeSize
unpack_sms_GetSMSCAddress_t, 1113

u16PRLVersion
unpack_dms_GetPRLVersion_t, 1002
u8PRLPreference
unpack_dms_GetPRLVersion_t, 1002
UATISIZE
qaGobiApiNas.h, 1625
UDPDstPort
unpack_qos_swiQosFilter_t, 1104
UDPSrcPort
unpack_qos_swiQosFilter_t, 1104
UIMAuthenticateReq, 958
authData, 959
pIndicationToken, 959
sessionInfo, 959
UIMAuthenticateResp, 959
pAuthenticateResult, 960
pCardResult, 960
pIndicationToken, 960
UIMChangePIN
qaGobiApiDms.h, 1576
UIMChangePinReq, 960
changePIN, 960
pIndicationToken, 960
pKeyReferenceID, 960
sessionInfo, 960
UIMDepersonalizationReq, 961
depersonilisationInfo, 961
UIMDepersonalizationResp, 961
pRemainingRetries, 961
UIMEventRegisterReqResp, 961
eventMask, 962
UIMGetCardStatusResp, 962
pCardStatus, 962
pHotSwapStatus, 962
UIMGetConfigurationReq, 962
pConfigurationMask, 963
UIMGetConfigurationResp, 963
pAutoSelection, 964
pHaltSubscription, 964
pPersonalizationStatus, 964
UIMGetControlKeyStatus
qaGobiApiDms.h, 1577
UIMGetFileAttributesReq, 964
fileIndex, 964
pIndicationToken, 964
sessionInfo, 964
UIMGetFileAttributesResp, 964
pCardResult, 965
pFileAttributes, 965
pIndicationToken, 965
UIMGetICCID
qaGobiApiDms.h, 1578
UIMGetPINStatus
qaGobiApiDms.h, 1579
UIMGetSlotsStatusResp, 965
pNumberOfPhySlot, 965
pUimSlotsStatus, 965

UIMPinResp, 966
pEncryptedPIN1, 966
pIndicationToken, 966
pRemainingRetries, 966
UIMPowerDownReq, 966
slot, 967
UIMPowerUpReq, 967
pIgnoreHotSwapSwitch, 967
slot, 967
UIMReadTransparentReq, 967
fileIndex, 968
pEncryptData, 968
pIndicationToken, 968
readTransparent, 968
sessionInfo, 968
UIMReadTransparentResp, 968
pCardResult, 969
pEncryptedData, 969
pIndicationToken, 969
pReadResult, 969
UIMRefreshCompleteReq, 969
refreshComplete, 970
sessionInfo, 970
UIMRefreshEvent, 970
aid, 971
aidLength, 971
arrfileInfo, 971
mode, 971
numOfFiles, 971
sessionType, 971
stage, 971
UIMRefreshGetLastEventReq, 971
sessionInfo, 972
UIMRefreshGetLastEventResp, 972
pRefreshEvent, 972
UIMRefreshOKReq, 972
OKtoRefresh, 973
sessionInfo, 973
UIMRefreshRegisterReq, 973
regRefresh, 973
sessionInfo, 973
UIMSessionInformation, 973
aid, 974
aidLength, 974
sessionType, 974
UIMSetControlKeyProtection
qaGobiApiDms.h, 1580
UIMSetPINProtection
qaGobiApiDms.h, 1580
UIMSetPinProtectionReq, 974
pIndicationToken, 975
pKeyReferenceID, 975
pinProtection, 975
sessionInfo, 975
UIMSlotStatus, 975
bICCID, 976
bICCIDLength, 976
bLogicalSlot, 976
uPhyCardStatus, 976
uPhySlotStatus, 976
UIMSlotStatusChangeInfo, 977
bNumberOfPhySlots, 977
slotsstatusChange, 977
UIMSlotsStatus, 975
uimSlotStatus, 975
UIMStatusChangeInfo, 977
statusChange, 977
UIMSwitchSlotReq, 977
bLogicalSlot, 978
ulPhysicalSlot, 978
UIMUnblockControlKey
qaGobiApiDms.h, 1581
UIMUnblockPIN
qaGobiApiDms.h, 1582
UIMUnblockPinReq, 978
pIndicationToken, 979
pKeyReferenceID, 979
sessionInfo, 979
unblockPIN, 979
UIMVerifyPIN
qaGobiApiDms.h, 1583
UIMVerifyPinReq, 979
pEncryptedPIN1, 980
pIndicationToken, 980
pKeyReferenceID, 980
sessionInfo, 980
verifyPIN, 980
ULONG
SwiDataTypes.h, 1904
ULONGLONG
SwiDataTypes.h, 1904
UMTSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1180
UMTSInfo, 980
cellID, 981
ecio, 981
geranInst, 981
GeranInstInfo, 981
lac, 981
plmn, 981
psc, 982
rscp, 982
UMTSInstInfo, 982
uarfcn, 982
umtsInst, 982
UMTSInstInfo
nas_UMTSInfo, 512
UMTSInfo, 982
UMTSLTENbrCell
nas_WCDMAInfoLTENeighborCell, 517
WCDMAInfoLTENeighborCell, 1284
UMTSMinQoS, 983
deliveryErrSDU, 985
grntDownlinkBitrate, 985
grntUplinkBitrate, 986
maxDownlinkBitrate, 986

maxSDUSize, 986
maxUplinkBitrate, 986
qosDeliveryOrder, 986
resBerRatio, 986
sduErrorRatio, 986
trafficClass, 986
trafficPriority, 986
transferDelay, 986
UMTSQoS, 986
deliveryErrSDU, 988
grntDownlinkBitrate, 988
grntUplinkBitrate, 988
maxDownlinkBitrate, 988
maxSDUSize, 988
maxUplinkBitrate, 988
qosDeliveryOrder, 988
resBerRatio, 988
sduErrorRatio, 988
trafficClass, 988
trafficPriority, 988
transferDelay, 988
UMTSReqQoS
LibPackUMTSReqQoSSigInd, 357
UMTSReqQoSSigInd, 989
UMTSReqQoSSigInd, 988
SigInd, 989
UMTSReqQoS, 989
UMTSinstInfo, 982
umtsEcio, 982
umtsPsc, 982
umtsRscp, 982
umtsUarfcn, 982
UNIQUE_ID_LEN
dms.h, 1337
qaGobiApiFms.h, 1588
UNUSEDPARAM
common.h, 1327
SwiDataTypes.h, 1904
uPhyCardStatus
slot_t, 835
UIMSlotStatus, 976
uPhySlotStatus
slot_t, 835
UIMSlotStatus, 976
uResult
sGetDeviceSeriesResult, 831
USBComp
pack_dms_SetUSBComp_t, 582
unpack_dms_GetUSBComp_t, 1003
USBCompConfig, 1187
pUSBComp, 1188
USBCompParams, 1188
pNumSupUSBComps, 1190
pSupUSBComps, 1190
pUSBComp, 1190
USHORT
SwiDataTypes.h, 1904
USSD_DCS_8BIT
qaGobiApiCbk.h, 1463
USSD_DCS_ASCII
qaGobiApiCbk.h, 1463
USSD_DCS_UCS2
qaGobiApiCbk.h, 1463
USSDNoWaitIndicationInfo, 1190
pAlphaIdentifier, 1191
pError, 1191
pFailureCause, 1191
pUSSDData, 1191
USSDRespFNetwork, 1191
pRespData, 1191
pTypeCode, 1191
USSInfo, 1191
ussDCS, 1192
ussData, 1192
ussLen, 1192
USSInformation
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 672
voiceOrigUSSDNoWaitInfo, 1264
USSResp, 1192
pAlphaIDInfo, 1192
pCCSuppsType, 1192
pCallId, 1192
pCcResultType, 1192
pUSSDInfo, 1193
pfailureCause, 1192
UUSData
UUSInfo, 1194
voice_UUSInfo, 1231
UUSDatalen
UUSInfo, 1194
voice_UUSInfo, 1231
UUSDcs
UUSInfo, 1194
voice_UUSInfo, 1231
UUSInfo, 1193
UUSData, 1194
UUSDatalen, 1194
UUSDcs, 1194
UUSType, 1194
UUSType
UUSInfo, 1194
voice_UUSInfo, 1231
uarfcn
lteWcdmaCellInfo, 403
nas_lteWcdmaCellInfo, 465
nas_UMTSInfo, 511
nas_wcdmaUARFCN, 521
UMTSInfo, 982
wcdmaUARFCN, 1293
ueInIdle
LTEInfoInterfreq, 387
LTEInfoIntrafreq, 389
LTEInfoNeighboringGSM, 390
LTEInfoNeighboringWCDMA, 391
nas_LTEInfoInterfreq, 452
nas_LTEInfoIntrafreq, 454

nas_LTEInfoNeighboringGSM, 455
nas_LTEInfoNeighboringWCDMA, 456
uim.h, 1915
MAX_ICCID_LENGTH, 1918
MAX_NO_OF_SLOTS, 1918
MAX_SLOTS_STATUS, 1918
pack_uim_ChangePin, 1918
pack_uim_GetCardStatus, 1919
pack_uim_ReadTransparent, 1919
pack_uim_SLQSUIMAuthenticate, 1920
pack_uim_SLQSUIMDepersonalization, 1920
pack_uim_SLQSUIMEventRegister, 1920
pack_uim_SLQSUIMGetConfiguration, 1921
pack_uim_SLQSUIMGetFileAttributes, 1921
pack_uim_SLQSUIMGetSlotsStatus, 1922
pack_uim_SLQSUIMPowerDown, 1922
pack_uim_SLQSUIMPowerUp, 1922
pack_uim_SLQSUIMRefreshComplete, 1923
pack_uim_SLQSUIMRefreshGetLastEvent, 1923
pack_uim_SLQSUIMRefreshOK, 1923
pack_uim_SLQSUIMRefreshRegister, 1924
pack_uim_SLQSUIMReset, 1924
pack_uim_SLQSUIMSwitchSlot, 1924
pack_uim_SetPinProtection, 1919
pack_uim_UnblockPin, 1925
pack_uim_VerifyPin, 1925
unpack_uim_ChangePin, 1926
unpack_uim_GetCardStatus, 1926
unpack_uim_ReadTransparent, 1926
unpack_uim_SLQSUIMAuthenticate, 1927
unpack_uim_SLQSUIMDepersonalization, 1928
unpack_uim_SLQSUIMEventRegister, 1928
unpack_uim_SLQSUIMGetConfiguration, 1928
unpack_uim_SLQSUIMGetFileAttributes, 1929
unpack_uim_SLQSUIMGetSlotsStatus, 1929
unpack_uim_SLQSUIMPowerDown, 1930
unpack_uim_SLQSUIMPowerUp, 1930
unpack_uim_SLQSUIMRefreshComplete, 1930
unpack_uim_SLQSUIMRefreshGetLastEvent, 1930
unpack_uim_SLQSUIMRefreshOK, 1931
unpack_uim_SLQSUIMRefreshRegister, 1931
unpack_uim_SLQSUIMReset, 1931
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind, 1932
unpack_uim_SLQSUIMSwitchSlot, 1932
unpack_uim_SetPinProtection, 1927
unpack_uim_SetUimSlotStatusChangeCallback_ind, 1927
unpack_uim_UnblockPin, 1933
unpack_uim_VerifyPin, 1933
uim_UIMSessionInformation, 956
aid, 957
aidLength, 957
sessionType, 957
uim_appStatus, 933
aidLength, 936
aidVal, 936
appState, 936
appType, 936
persoFeature, 936
persoRetries, 936
persoState, 936
persoUnblockRetries, 936
pin1Retries, 936
pin1State, 936
pin2Retries, 936
pin2State, 936
puk1Retries, 936
puk2Retries, 936
univPin, 936
uim_authenticateResult, 936
content, 937
contentLen, 937
uim_authenticationData, 937
context, 938
data, 938
dataLen, 938
uim_cardResult, 938
sw1, 939
sw2, 939
uim_cardStatus, 939
index1xPri, 940
index1xSec, 940
indexGwPri, 940
indexGwSec, 940
numSlot, 940
SlotInfo, 940
uim_changeUIMPIN, 940
oldPINLen, 941
oldPINVal, 941
pinID, 941
pinLen, 941
pinVal, 941
uim_depersonalizationInformation, 941
ckLen, 942
ckVal, 942
feature, 942
operation, 942
uim_encryptedPIN1, 942
pin1Len, 943
pin1Val, 943
uim_fileAttributes, 943
fileID, 946
fileSize, 946
fileType, 946
rawLen, 946
rawValue, 946
recordCount, 946
recordSize, 946
secActivate, 946
secActivateMask, 946
secDeactivate, 946
secDeactivateMask, 946
secIncrease, 946
secIncreaseMask, 946

secRead, 946
secReadMask, 946
secWrite, 946
secWriteMask, 946
uim_fileInfo, 946
fileID, 947
path, 947
pathLen, 947
uim_hotSwapStatus, 947
hotSwap, 947
hotSwapLength, 947
uim_personalizationStatus, 947
feature, 948
numFeatures, 948
unblockLeft, 948
verifyLeft, 948
uim_readResult, 948
content, 949
contentLen, 949
uim_readTransparentInfo, 949
length, 949
offset, 949
uim_refreshevent, 949
aid, 951
aidLength, 951
arrfileInfo, 951
mode, 951
numOfFiles, 951
sessionType, 951
stage, 951
uim_registerRefresh, 951
arrfileInfo, 952
numFiles, 952
registerFlag, 952
voteForInit, 952
uim_remainingRetries, 952
unblockLeft, 952
verifyLeft, 952
uim_sessionInformation, 953
aid, 953
aidLength, 953
sessionType, 953
uim_setPINProtection, 953
pinID, 954
pinLength, 954
pinOperation, 954
pinValue, 954
uim_slotInfo, 954
AppStatus, 956
cardState, 956
errorState, 956
numApp, 956
upinRetries, 956
upinState, 956
upukRetries, 956
uim_unblockUIMPIN, 957
newPINLen, 957
newPINVal, 957
pinID, 957
pukLen, 957
pukVal, 958
uim_verifyUIMPIN, 958
pinID, 958
pinLen, 958
pinVal, 958
uimSlotStatus
slots_t, 838
UIMSlotsStatus, 975
ulData
DataULongTlv, 215
wds_DataULongTlv, 1295
ulMask
rmTrasnferStaticsReq, 781
swiRMTrasnferStaticsReq, 910
ulMax
_QmiSwiAvmsSetSettingsPeriodInfo, 80
ulMin
_QmiSwiAvmsSetSettingsPeriodInfo, 80
ulPhysicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 657
UIMSwitchSlotReq, 978
ulPkgDownloadComplete
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
ulPkgDownloadSize
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
ullData
DataULongLongTlv, 215
wds_DataULongLongTlv, 1294
umtsEcio
nas_UMTSinstInfo, 512
UMTSinstInfo, 982
umtsInst
nas_UMTSInfo, 511
UMTSInfo, 982
umtsLTENbrCell, 982
cellIsTDD, 983
earfcn, 983
pci, 983
rsrp, 983
rsrq, 983
srxlev, 983
umtsLTENbrCellLen
nas_WCDMAInfoLTENeighborCell, 517
WCDMAInfoLTENeighborCell, 1284
umtsPsc
nas_UMTSinstInfo, 512
UMTSinstInfo, 982
umtsRscp
nas_UMTSinstInfo, 512
UMTSinstInfo, 982
umtsUarfcn
nas_UMTSinstInfo, 512
UMTSinstInfo, 982
UnPackGetProfileSettingOut, 1186

curProfile, 1187
pExtErrCode, 1187
unblockLeft
personalizationStatus, 707
remainingRetries, 777
uim_personalizationStatus, 948
uim_remainingRetries, 952
unblockPIN
UIMUnblockPinReq, 979
unblockRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1021
unpack_dms_UIMSetPINProtection_t, 1024
unpack_dms_UIMUnblockControlKey_t, 1025
unblockUIMPIN, 989
newPINLen, 990
newPINVal, 990
pinID, 990
pukLen, 990
pukVal, 990
uniqueID
CurrImageInfo, 199
image_info_t, 305
univPin
appStats, 124
appStatus, 127
uim_appStatus, 936
UniversalTime, 990
day, 991
dayOfWeek, 991
hour, 991
minute, 991
month, 991
second, 991
year, 991
universalTime
nasNetworkTime, 540
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1074
unpack_QosFlowStat_t, 1109
bearerId, 1110
tx_bytes, 1110
tx_bytes_drp, 1110
tx_pkt, 1110
tx_pkt_drp, 1110
unpack_RMTransferStatistics_ind_t, 1110
RxDropConutTlv, 1111
RxOkByteCountTlv, 1111
RxOkConutTlv, 1111
TxDropConutTlv, 1111
TxOkByteCountTlv, 1111
TxOkConutTlv, 1111
unpack_dms_ActivateAutomatic
dms.h, 1359
unpack_dms_GetActivationState
dms.h, 1359
unpack_dms_GetActivationState_t, 991
state, 992
unpack_dms_GetBandCapability
dms.h, 1360
unpack_dms_GetBandCapability_t, 992
BandCapability, 992
Tlvresult, 992
unpack_dms_GetCrashAction
dms.h, 1360
unpack_dms_GetCrashAction_t, 992
DevCrashState, 992
Tlvresult, 992
unpack_dms_GetCustFeature
dms.h, 1361
unpack_dms_GetCustFeature_t, 992
DHCPRelayEnabled, 993
DisableIMSI, 993
GPSLPM, 993
GPSSel, 993
GpsEnable, 993
IPFamSupport, 993
IsVoiceEnabled, 993
RMAutoConnect, 993
SMSSupport, 993
Tlvresult, 993
unpack_dms_GetCustFeaturesV2
dms.h, 1361
unpack_dms_GetCustFeaturesV2_t, 993
GetCustomFeatureV2, 993
Tlvresult, 993
unpack_dms_GetDeviceCap
dms.h, 1361
unpack_dms_GetDeviceCap_t, 994
DataServiceCapability, 994
MaxRXChannelRate, 994
MaxTXChannelRate, 994
RadioIfaces, 994
RadioIfacesSize, 994
SimCapability, 994
Tlvresult, 994
unpack_dms_GetDeviceCapabilities
dms.h, 1361
unpack_dms_GetDeviceCapabilities_t, 994
dataServiceCaCapability, 995
maxRxChannelRate, 995
maxTxChannelRate, 995
RadioIfaces, 995
radioIfacesSize, 995
simCapability, 995
unpack_dms_GetDeviceHardwareRev
dms.h, 1362
unpack_dms_GetDeviceHardwareRev_t, 995
String, 995
stringSize, 995
Tlvresult, 995
unpack_dms_GetDeviceMfr
dms.h, 1362
unpack_dms_GetDeviceMfr_t, 995
String, 995
stringSize, 995
Tlvresult, 995

unpack_dms_GetDeviceSerialNumbers
dms.h, 1362
unpack_dms_GetDeviceSerialNumbers_t, 995
ESNString, 996
esnSize, 996
IMEIString, 996
imeiSize, 996
imeiSvnSize, 996
ImeiSvnString, 996
MEIDString, 996
meidSize, 996
Tlvresult, 996
unpack_dms_GetFSN
dms.h, 1364
unpack_dms_GetFSN_t, 998
String, 998
Tlvresult, 998
unpack_dms_GetFirmwareInfo
dms.h, 1363
unpack_dms_GetFirmwareInfo_t, 996
appversion_str, 997
bootversion_str, 997
carrier_str, 997
cur_carr_name, 997
cur_carr_rev, 997
modelid_str, 997
packageid_str, 997
priversion_str, 997
sku_str, 997
Tlvresult, 997
unpack_dms_GetFirmwareRevision
dms.h, 1363
unpack_dms_GetFirmwareRevision_t, 997
AMSSString, 997
amssSize, 997
PRIString, 997
Tlvresult, 997
unpack_dms_GetFirmwareRevisions
dms.h, 1363
unpack_dms_GetFirmwareRevisions_t, 997
AMSSString, 998
amssSize, 998
bootSize, 998
BootString, 998
PRIString, 998
priSize, 998
Tlvresult, 998
unpack_dms_GetHardwareRevision
dms.h, 1364
unpack_dms_GetHardwareRevision_t, 998
hwVer, 999
unpack_dms_GetIMSI
dms.h, 1364
unpack_dms_GetIMSI_t, 999
imsi, 999
Tlvresult, 999
unpack_dms_GetManufacturer
dms.h, 1365
unpack_dms_GetManufacturer_t, 999
manufacturer, 999
Tlvresult, 999
unpack_dms_GetModelID
dms.h, 1365
unpack_dms_GetModelID_t, 999
modelid, 1000
Tlvresult, 1000
unpack_dms_GetNetworkTime
dms.h, 1365
unpack_dms_GetNetworkTime_t, 1000
source, 1000
timestamp, 1000
Tlvresult, 1000
unpack_dms_GetOfflineReason
dms.h, 1366
unpack_dms_GetOfflineReason_t, 1001
pReasonMask, 1001
pbPlatform, 1001
Tlvresult, 1001
unpack_dms_GetPRLVersion
dms.h, 1366
unpack_dms_GetPRLVersion_t, 1002
Tlvresult, 1002
u16PRLVersion, 1002
u8PRLPreference, 1002
unpack_dms_GetPower
dms.h, 1366
unpack_dms_GetPower_t, 1001
HardwareControlledMode, 1002
OfflineReason, 1002
OperationMode, 1002
Tlvresult, 1002
unpack_dms_GetSerialNumbers
dms.h, 1367
unpack_dms_GetSerialNumbers_t, 1002
esn, 1002
imei_no, 1003
imeisv_svn, 1003
meid, 1003
unpack_dms_GetUSBComp
dms.h, 1367
unpack_dms_GetUSBComp_t, 1003
NumSupUSBComps, 1003
SupUSBComps, 1003
Tlvresult, 1003
USBComp, 1003
unpack_dms_GetVoiceNumber
dms.h, 1367
unpack_dms_GetVoiceNumber_t, 1003
MIN, 1003
minSize, 1003
Tlvresult, 1003
VoiceNumber, 1003
voiceNumberSize, 1003
unpack_dms_ResetToFactoryDefaults
dms.h, 1368
unpack_dms_ResetToFactoryDefaults_t, 1003

Tlvresult, 1004
unpack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1371
unpack_dms_SLQSDmsSwiGetResetInfo_Ind
dms.h, 1371
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1007
source, 1007
Tlvresult, 1007
type, 1007
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1008
source, 1008
Tlvresult, 1008
type, 1008
unpack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1372
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1008
Tlvresult, 1009
unpack_dms_SLQSGetBandCapability
dms.h, 1372
unpack_dms_SLQSGetBandCapability_t, 1009
bandCapability, 1012
LteBandCapability, 1012
TdsBandCapability, 1012
unpack_dms_SLQSGetERIFile
dms.h, 1373
unpack_dms_SLQSGetERIFile_t, 1012
eriFile, 1012
Tlvresult, 1012
unpack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1373
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 1012
Tlvresult, 1013
unpack_dms_SLQSSwiGetCrashInfo
dms.h, 1373
unpack_dms_SLQSSwiGetCrashInfo_t, 1013
crashInfoParam, 1013
Tlvresult, 1013
unpack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1374
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1013
pGetDyingGaspCfg, 1014
Tlvresult, 1014
unpack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1374
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1014
pGetDyingGaspStatistics, 1014
Tlvresult, 1014
unpack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1374
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1014
carrier, 1015
fwvers, 1015
numEntries, 1015
pCurrImgInfo, 1015
pkgver, 1015
priver, 1015
unpack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1375
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1015
imgType, 1016
logString, 1016
refData, 1016
refString, 1016
ResCode, 1016
Tlvresult, 1016
unpack_dms_SLQSSwiGetHostDevInfo
dms.h, 1375
unpack_dms_SLQSSwiGetHostDevInfo_t, 1017
manString, 1017
modelString, 1017
plasmaIDString, 1017
swVerString, 1017
Tlvresult, 1017
unpack_dms_SLQSSwiGetOSInfo
dms.h, 1375
unpack_dms_SLQSSwiGetOSInfo_t, 1017
nameString, 1018
Tlvresult, 1018
versionString, 1018
unpack_dms_SLQSSwiGetSerialNoExt
dms.h, 1376
unpack_dms_SLQSSwiGetSerialNoExt_t, 1018
meidString, 1018
Tlvresult, 1018
unpack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1376
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1018
Tlvresult, 1019
unpack_dms_SLQSSwiSetHostDevInfo
dms.h, 1376
unpack_dms_SLQSSwiSetHostDevInfo_t, 1019
Tlvresult, 1019
unpack_dms_SLQSSwiSetOSInfo
dms.h, 1377
unpack_dms_SLQSSwiSetOSInfo_t, 1019
Tlvresult, 1020
unpack_dms_SLQSUIMGetState
dms.h, 1377
unpack_dms_SLQSUIMGetState_t, 1020
state, 1020
Tlvresult, 1020
unpack_dms_SetActivationStatusCallback
dms.h, 1368
unpack_dms_SetActivationStatusCallback_t, 1004
Tlvresult, 1004
unpack_dms_SetCrashAction
dms.h, 1368
unpack_dms_SetCrashAction_t, 1004
notused, 1004
unpack_dms_SetCustFeature
dms.h, 1369
unpack_dms_SetCustFeature_t, 1005
Tlvresult, 1005
unpack_dms_SetCustFeaturesV2
dms.h, 1369

unpack_dms_SetCustFeaturesV2_t, 1005
    TlvResult, 1005
unpack_dms_SetEventReport
    dms.h, 1369
unpack_dms_SetEventReport_ind
    dms.h, 1370
unpack_dms_SetEventReport_ind_t, 1005
    ActivationStatusTlv, 1006
    OperatingModeTlv, 1006
    TlvResult, 1006
unpack_dms_SetEventReport_t, 1006
    TlvResult, 1006
unpack_dms_SetFirmwarePreference
    dms.h, 1370
unpack_dms_SetFirmwarePreference_t, 1006
    TlvResult, 1006
unpack_dms_SetPower
    dms.h, 1370
unpack_dms_SetPower_t, 1006
    TlvResult, 1006
unpack_dms_SetUSBComp
    dms.h, 1371
unpack_dms_SetUSBComp_t, 1006
    TlvResult, 1007
unpack_dms_UIMChangePIN
    dms.h, 1377
unpack_dms_UIMGetControlKeyStatus
    dms.h, 1378
unpack_dms_UIMGetControlKeyStatus_t, 1020
    facilityState, 1021
    TlvResult, 1021
    unblockRetriesLeft, 1021
    verifyRetriesLeft, 1021
unpack_dms_UIMGetICCID
    dms.h, 1378
unpack_dms_UIMGetICCID_t, 1021
    String, 1021
    stringSize, 1022
    TlvResult, 1022
unpack_dms_UIMGetPINStatus
    dms.h, 1378
unpack_dms_UIMGetPINStatus_t, 1022
    p1Status, 1023
    p1UnblockRetriesLeft, 1023
    p1VerifyRetriesLeft, 1023
    p2Status, 1023
    p2UnblockRetriesLeft, 1023
    p2VerifyRetriesLeft, 1023
    TlvResult, 1023
unpack_dms_UIMSetControlKeyProtection
    dms.h, 1379
unpack_dms_UIMSetControlKeyProtection_t, 1023
    TlvResult, 1024
    verifyRetriesLeft, 1024
unpack_dms_UIMSetPINProtection
    dms.h, 1379
unpack_dms_UIMSetPINProtection_t, 1024
    TlvResult, 1024
    unblockRetriesLeft, 1024
    verifyRetriesLeft, 1024
unpack_dms_UIMUnblockControlKey
    dms.h, 1379
unpack_dms_UIMUnblockControlKey_t, 1024
    TlvResult, 1025
    unblockRetriesLeft, 1025
unpack_dms_UIMUnblockPIN
    dms.h, 1380
unpack_dms_UIMVerifyPIN
    dms.h, 1380
unpack_dms_ValidateSPC
    dms.h, 1380
unpack_fms_GetImagesPreference
    fms.h, 1383
unpack_fms_GetImagesPreference_t, 1025
    ImageListSize, 1025
    pImageList, 1025
    TlvResult, 1025
unpack_fms_GetStoredImages
    fms.h, 1383
unpack_fms_GetStoredImages_t, 1026
    imageList, 1026
    imagelistSize, 1026
    TlvResult, 1026
unpack_fms_SetImagesPreference
    fms.h, 1383
unpack_fms_SetImagesPreference_t, 1026
    ImageTypes, 1027
    ImageTypesSize, 1027
    TlvResult, 1027
unpack_loc_BestAvailPos_Ind
    loc.h, 1393
unpack_loc_BestAvailPos_Ind_t, 1027
    pAltitudeWrtEllipsoid, 1031
    pAltitudeWrtMeanSeaLevel, 1031
    pGpsTime, 1031
    pHeading, 1031
    pHeadingUnc, 1031
    pHorCirConf, 1031
    pHorEllpConf, 1031
    pHorReliability, 1031
    pHorUncCircular, 1032
    pHorUncEllipseOrientAzimuth, 1032
    pHorUncEllipseSemiMajor, 1032
    pHorUncEllipseSemiMinor, 1032
    pLatitude, 1032
    pLongitude, 1032
    pMagneticDeviation, 1032
    pPrecisionDilution, 1032
    pSensorDataUsage, 1032
    pSpeedHorizontal, 1032
    pSpeedUnc, 1032
    pSpeedVertical, 1032
    pSpeedVerticalUnc, 1032
    pSvUsedforFix, 1032
    pTechnologyMask, 1032
    pTimeSrc, 1032

pTimeUnc, 1032
pTimestampUtc, 1032
pVertConfidence, 1032
pVertReliability, 1032
pVertUnc, 1032
pXid, 1032
status, 1032
Tlvresult, 1032
unpack_loc_Delete_Assist_Data_t, 1032
Tlvresult, 1033
unpack_loc_DeleteAssistData
loc.h, 1393
unpack_loc_DeleteAssistData_Ind
loc.h, 1393
unpack_loc_DeleteAssistData_Ind_t, 1033
status, 1033
Tlvresult, 1034
unpack_loc_EngineState_Ind
loc.h, 1394
unpack_loc_EngineState_Ind_t, 1034
engineState, 1034
Tlvresult, 1034
unpack_loc_EventRegister
loc.h, 1394
unpack_loc_EventRegister_t, 1034
Tlvresult, 1034
unpack_loc_GnssSvInfo_Ind
loc.h, 1394
unpack_loc_GnssSvInfo_Ind_t, 1035
altitudeAssumed, 1035
pSatelliteInfo, 1035
Tlvresult, 1035
unpack_loc_PositionRpt_Ind
loc.h, 1395
unpack_loc_PositionRpt_Ind_t, 1035
pAltitudeAssumed, 1040
pAltitudeWrtEllipsoid, 1040
pAltitudeWrtMeanSeaLevel, 1040
pFixId, 1040
pGpsTime, 1040
pHeading, 1040
pHeadingUnc, 1041
pHorConfidence, 1041
pHorReliability, 1041
pHorUncCircular, 1041
pHorUncEllipseOrientAzimuth, 1041
pHorUncEllipseSemiMajor, 1041
pHorUncEllipseSemiMinor, 1041
pLatitude, 1041
pLeapSeconds, 1041
pLongitude, 1041
pMagneticDeviation, 1041
pPrecisionDilution, 1041
pSensorDataUsage, 1041
pSpeedHorizontal, 1041
pSpeedUnc, 1041
pSpeedVertical, 1041
pSvUsedforFix, 1041
pTechnologyMask, 1041
pTimeSrc, 1041
pTimeUnc, 1041
pTimestampUtc, 1041
pVertConfidence, 1041
pVertReliability, 1041
pVertUnc, 1041
sessionId, 1041
sessionStatus, 1041
Tlvresult, 1041
unpack_loc_SLQSLOCGetBestAvailPos
loc.h, 1397
unpack_loc_SLQSLOCGetBestAvailPos_t, 1044
Tlvresult, 1044
unpack_loc_SLQSLOCInjectPosition
loc.h, 1397
unpack_loc_SLQSLOCInjectSensorData
loc.h, 1397
unpack_loc_SLQSLOCInjectUTCTime
loc.h, 1398
unpack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1398
unpack_loc_SetExtPowerConfig_Ind
loc.h, 1395
unpack_loc_SetExtPowerConfig_Ind_t, 1042
status, 1042
Tlvresult, 1042
unpack_loc_SetExtPowerState
loc.h, 1396
unpack_loc_SetExtPowerState_t, 1042
Tlvresult, 1043
unpack_loc_SetOperationMode
loc.h, 1396
unpack_loc_SetOperationMode_Ind
loc.h, 1396
unpack_loc_SetOperationMode_Ind_t, 1043
status, 1043
Tlvresult, 1043
unpack_loc_SetOperationMode_t, 1043
Tlvresult, 1044
unpack_loc_Start
loc.h, 1398
unpack_loc_Start_t, 1044
Tlvresult, 1044
unpack_loc_Stop
loc.h, 1398
unpack_loc_Stop_t, 1044
Tlvresult, 1045
unpack_nas_GetACCOLC
nas.h, 1423
unpack_nas_GetANAAAAuthenticationStatus
nas.h, 1423
unpack_nas_GetCDMANetworkParameters
nas.h, 1424
unpack_nas_GetCDMANetworkParameters_t, 1045
Application, 1045
Broadcast, 1045
CustomSCP, 1046

ForceRev0, 1046
Protocol, 1046
RegForeignNID, 1046
RegForeignSID, 1046
RegHomeSID, 1046
Roaming, 1046
SCI, 1046
SCM, 1046
unpack_nas_GetHomeNetwork
nas.h, 1424
unpack_nas_GetHomeNetwork3GPP2
nas.h, 1424
unpack_nas_GetHomeNetwork3GPP2_t, 1046
nameSize, 1047
pMCC, 1047
pMNC, 1047
pNID, 1048
pName, 1047
pNw2DescDisp, 1048
pNw2DescEnc, 1048
pNw2DescLen, 1048
pNw2MCC, 1048
pNw2MNC, 1048
pNw2Name, 1048
pSID, 1048
unpack_nas_GetHomeNetwork_t, 1048
mcc, 1048
mnc, 1048
name, 1048
nid, 1048
sid, 1048
unpack_nas_GetNetworkPreference
nas.h, 1425
unpack_nas_GetNetworkPreference_t, 1048
ActiveTechPref, 1049
Duration, 1049
PersistentTechPref, 1049
Tlvresult, 1049
unpack_nas_GetRFInfo
nas.h, 1425
unpack_nas_GetRFInfo_t, 1049
instancesSize, 1050
RFBandInfoElements, 1050
unpack_nas_GetServingNetwork
nas.h, 1425
unpack_nas_GetServingNetwork_t, 1050
CSDomain, 1051
DataCaps, 1051
DataCapsLen, 1051
MCC, 1051
MNC, 1051
Name, 1051
nameSize, 1051
PSDomain, 1051
RAN, 1051
RadioIfaces, 1051
RadioIfacesSize, 1051
RegistrationState, 1051
Roaming, 1051
unpack_nas_GetServingNetworkCapabilities
nas.h, 1425
unpack_nas_GetServingNetworkCapabilities_t, 1051
DataCaps, 1051
DataCapsLen, 1051
unpack_nas_GetSignalStrengths
nas.h, 1426
unpack_nas_GetSignalStrengths_t, 1051
len, 1052
radio, 1052
rssi, 1052
unpack_nas_InitiateDomainAttach
nas.h, 1426
unpack_nas_PerformNetworkScan
nas.h, 1426
unpack_nas_PerformNetworkScan_t, 1052
p3GppNetworkInfoInstances, 1052
p3GppNetworkInstanceSize, 1052
pPCSInstance, 1052
pPCSInstanceSize, 1052
pRATINstance, 1052
pRATInstanceSize, 1052
pScanResult, 1052
unpack_nas_SLQSConfigSigInfo
nas.h, 1429
unpack_nas_SLQSGetErrorRate
nas.h, 1429
unpack_nas_SLQSGetErrorRate_t, 1056
pCDMAFrameErrRate, 1057
pGSMBER, 1057
pHDRPackErrRate, 1057
pWCDMABER, 1057
unpack_nas_SLQSGetNetworkTime
nas.h, 1430
unpack_nas_SLQSGetNetworkTime_t, 1058
p3GPP2TimeInfo, 1058
p3GPPTimeInfo, 1058
unpack_nas_SLQSGetOperatorNameData
nas.h, 1430
unpack_nas_SLQSGetOperatorNameData_t, 1058
pNITZInformation, 1059
pOperatorNameString, 1059
pOperatorPLMNList, 1059
pPLMNNetworkName, 1059
pSrvcProviderName, 1059
unpack_nas_SLQSGetPLMNName
nas.h, 1430
unpack_nas_SLQSGetPLMNName_t, 1059
longName, 1059
longNameCI, 1059
longNameEn, 1059
longNameLen, 1059
longNameSB, 1059
shortName, 1059
shortNameCI, 1059
shortNameEn, 1060
shortNameLen, 1060

shortNameSB, 1060
spn, 1060
spnEncoding, 1060
spnLength, 1060
unpack_nas_SLQSGetServingSystem
nas.h, 1431
unpack_nas_SLQSGetServingSystem_t, 1060
BasestationID, 1061
BasestationLatitude, 1061
BasestationLongitude, 1061
CDMASystemInfoExt, 1061
CallBarStatus, 1061
CellID, 1061
ConcSvcInfo, 1061
CurrentPLMN, 1061
DTMInd, 1061
DataSrvCapabilities, 1061
DefaultRoamInd, 1061
DetailedSvcInfo, 1061
Gpp2TimeZone, 1061
GppNetworkDSTAdjustment, 1062
GppTimeZone, 1062
HdrPersonality, 1062
Lac, 1062
NetworkID, 1062
PRLInd, 1062
RoamIndicatorVal, 1062
RoamingIndicatorList, 1062
ServingSystem, 1062
SystemID, 1062
TrackAreaCode, 1062
unpack_nas_SLQSGetSignalStrength
nas.h, 1431
unpack_nas_SLQSGetSignalStrength_t, 1062
ecioList, 1063
ecioListLen, 1063
errorRateList, 1063
errorRateListLen, 1063
Io, 1063
ltersrp, 1063
ltesnr, 1063
rsrqInfo, 1063
rxSignalStrengthList, 1063
rxSignalStrengthListLen, 1063
signalStrengthReqMask, 1063
sinr, 1063
unpack_nas_SLQSGetSysInfo
nas.h, 1431
unpack_nas_SLQSGetSysInfo_t, 1063
pAddCDMASysInfo, 1065
pAddGSMSysInfo, 1065
pAddHDRSysInfo, 1065
pAddLTESysInfo, 1065
pCDMASrvStatusInfo, 1065
pCDMASysInfo, 1065
pGSMCallBarringSysInfo, 1065
pGSMCipherDomainSysInfo, 1065
pGSMSrvStatusInfo, 1065
pGSMSysInfo, 1065
pHDRSrvStatusInfo, 1065
pHDRSysInfo, 1065
pLTESrvStatusInfo, 1065
pLTESysInfo, 1065
pLTEVoiceSupportSysInfo, 1066
pWCDMASysInfo, 1066
unpack_nas_SLQSGetSysSelectionPref
nas.h, 1432
unpack_nas_SLQSGetSysSelectionPref_t, 1066
pBandPref, 1069
pEmerMode, 1069
pGWAcqOrderPref, 1069
pLTEBandPref, 1069
pModePref, 1069
pNetSelPref, 1069
pPRLPref, 1069
pRoamPref, 1069
pSrvDomainPref, 1069
unpack_nas_SLQSInitiateNetworkRegistration
nas.h, 1432
unpack_nas_SLQSNASSwiGetChannelLock
nas.h, 1435
unpack_nas_SLQSNASSwiGetChannelLock_t, 1075
unpack_nas_SLQSNASSwiSetChannelLock
nas.h, 1436
unpack_nas_SLQSNasConfigSigInfo2
nas.h, 1432
unpack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1432
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1069
pDirNum, 1070
unpack_nas_SLQSNasGetCellLocationInfo
nas.h, 1433
unpack_nas_SLQSNasGetCellLocationInfo_t, 1070
pCDMAInfo, 1071
pLTEInfoInterfreq, 1071
pLTEInfoIntrafreq, 1071
pUMTSInfo, 1071
unpack_nas_SLQSNasGetHDRColorCode
nas.h, 1433
unpack_nas_SLQSNasGetHDRColorCode_t, 1072
pColorCode, 1072
unpack_nas_SLQSNasGetSigInfo
nas.h, 1433
unpack_nas_SLQSNasGetSigInfo_t, 1072
CDMASSInfo, 1072
GSMSSInfo, 1072
HDRSSInfo, 1072
LTESSInfo, 1073
unpack_nas_SLQSNasGetTxRxInfo
nas.h, 1434
unpack_nas_SLQSNasGetTxRxInfo_t, 1073
pRXChain0Info, 1073
pRXChain1Info, 1073
pTXInfo, 1073
unpack_nas_SLQSNasIndicationRegisterExt
nas.h, 1434

unpack_nas_SLQSNasNetworkTimeCallBack_ind
nas.h, 1434
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1073
pDayltSavAdj, 1074
pRadioInterface, 1074
pTimeZone, 1074
universalTime, 1074
unpack_nas_SLQSNasSigInfoCallback_ind
nas.h, 1435
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1074
pRscp, 1075
unpack_nas_SLQSNasSwiIndicationRegister
nas.h, 1435
unpack_nas_SLQSNasSwiModemStatus
nas.h, 1436
unpack_nas_SLQSNasSwiModemStatus_t, 1076
commonInfo, 1076
pLTEInfo, 1076
unpack_nas_SLQSNasSwiOTAMessageCallback_ind
nas.h, 1436
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1076
Info, 1076
Tlvresult, 1076
unpack_nas_SLQSNasSysInfoCallback_ind
nas.h, 1437
unpack_nas_SLQSNasTimerCallback_ind
nas.h, 1437
unpack_nas_SLQSNasTimerCallback_ind_t, 1077
t3396_apn, 1077
t3396_plmn_id, 1077
t3396_val, 1077
unpack_nas_SLQSSetBandPreference
nas.h, 1437
unpack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1437
unpack_nas_SLQSSetSysSelectionPref
nas.h, 1438
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
nas.h, 1438
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1077
Info, 1078
Tlvresult, 1078
unpack_nas_SLQSSwiGetHDRPersonality
nas.h, 1438
unpack_nas_SLQSSwiGetHDRPersonality_t, 1078
pCurrentPersonality, 1078
pPersonalityListLength, 1078
pProtocolSubtypeElement, 1078
unpack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1439
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1078
pAppSubType, 1079
pCurrentPrsnlty, 1079
pPersonalityListLength, 1079
unpack_nas_SLQSSwiGetHRPDStats
nas.h, 1439
unpack_nas_SLQSSwiGetHRPDStats_t, 1079
pPilotSetData, 1080
pUATI, 1080
unpack_nas_SLQSSwiGetLteCQI
nas.h, 1439
unpack_nas_SLQSSwiGetLteCQI_t, 1080
ValidityCW0, 1080
ValidityCW1, 1080
unpack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1440
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1080
pSccRxInfo, 1081
unpack_nas_SLQSSwiNetworkDebug
nas.h, 1440
unpack_nas_SLQSSwiNetworkDebug_t, 1081
pDataStatusDetail, 1081
pDeviceConfigDetail, 1081
pNetworkStat1x, 1081
pNetworkStatEVDO, 1081
pObjectVer, 1081
unpack_nas_SLQSSwiPSDetach
nas.h, 1440
unpack_nas_SLQSSysInfoCallback_ind_t, 1082
pGSMSysInfo, 1084
pHDRSysInfo, 1084
pLTESysInfo, 1084
pSysInfoNoChange, 1084
unpack_nas_SetACCOLC
nas.h, 1427
unpack_nas_SetCDMANetworkParameters
nas.h, 1427
unpack_nas_SetDataCapabilitiesCallback_ind
nas.h, 1427
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1053
dataCaps, 1053
dataCapsSize, 1053
unpack_nas_SetEventReportInd
nas.h, 1428
unpack_nas_SetEventReportInd_t, 1053
RFTlv, 1053
RRTlv, 1053
SLQSSSTlv, 1053
SSTlv, 1053
unpack_nas_SetLURejectCallback
nas.h, 1428
unpack_nas_SetNasLTECphyCaIndCallback_ind
nas.h, 1428
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1053
unpack_nas_SetNetworkPreference
nas.h, 1428
unpack_nas_SetNetworkPreference_t, 1054
Tlvresult, 1055
unpack_nas_SetRFInfoCallback
nas.h, 1428
unpack_nas_SetRoamingIndicatorCallback_ind
nas.h, 1429

unpack_nas_SetRoamingIndicatorCallback_ind_t, 1055
roaming, 1056
unpack_nas_SetServingSystemCallback_ind
nas.h, 1429
unpack_nas_SetServingSystemCallback_ind_t, 1056
SSInfo, 1056
Tlvresult, 1056
unpack_nas_SlqsGetLTECphyCAInfo
nas.h, 1430
unpack_nas_SlqsGetLTECphyCAInfo_t, 1057
LTECphyCaInfo, 1057
Tlvresult, 1057
unpack_omaDmConfigTlv_t, 1084
alertmsg, 1085
alertmsglength, 1085
state, 1085
userInputReq, 1085
userInputTimeout, 1085
unpack_omaDmFotaTlv_t, 1085
description, 1087
descriptionlength, 1087
fwdloadsize, 1087
fwloadComplete, 1087
namelength, 1087
package_name, 1087
sessionType, 1087
severity, 1087
state, 1087
updateCompleteStatus, 1087
userInputReq, 1087
userInputTimeout, 1087
version, 1087
versionlength, 1087
unpack_omaDmNotificationsTlv_t, 1087
notification, 1088
sessionStatus, 1088
unpack_pds_GetPDSDefaults
pds.h, 1443
unpack_pds_GetPDSDefaults_t, 1088
pAccuracy, 1088
pInterval, 1088
pOperation, 1088
pTimeout, 1088
unpack_pds_GetPDSState
pds.h, 1444
unpack_pds_GetPDSState_t, 1088
pEnabledStatus, 1089
pTrackingStatus, 1089
unpack_pds_GetPortAutomaticTracking
pds.h, 1444
unpack_pds_GetPortAutomaticTracking_t, 1089
pbAuto, 1089
unpack_pds_SetPDSDefaults
pds.h, 1444
unpack_pds_SetPDSState
pds.h, 1445
unpack_pds_SetPortAutomaticTracking
pds.h, 1445
unpack_qmi_t, 1089
msgid, 1090
type, 1090
xid, 1090
unpack_qos_IPv4Addr_t, 1090
addr, 1091
subnetMask, 1091
unpack_qos_IPv6Addr_t, 1091
addr, 1091
prefixLen, 1091
unpack_qos_IPv6TrafCls_t, 1091
mask, 1092
val, 1092
unpack_qos_Port_t, 1092
port, 1093
range, 1093
unpack_qos_QosFlowInfo_t, 1093
BearerID, 1094
is_RxQFlowGranted_Available, 1094
is_TxQFlowGranted_Available, 1094
NumRxFilters, 1094
NumTxFilters, 1094
QFlowState, 1094
RxQFilter, 1094
RxQFlowGranted, 1094
TxQFilter, 1094
TxQFlowGranted, 1094
unpack_qos_QosFlowInfoState_t, 1094
id, 1095
isNewFlow, 1095
state, 1095
unpack_qos_SLQSQosGetNetworkStatus
qos.h, 1873
unpack_qos_SLQSQosGetNetworkStatus_t, 1095
NWQoSStatus, 1095
unpack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 1874
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1095
ambr_dl, 1096
ambr_dl_ext, 1096
ambr_dl_ext2, 1096
ambr_ul, 1096
ambr_ul_ext, 1096
ambr_ul_ext2, 1096
apnId, 1096
unpack_qos_SLQSQosSwiReadDataStats
qos.h, 1874
unpack_qos_SLQSQosSwiReadDataStats_t, 1097
apnId, 1098
numQosFlow, 1098
qosFlow, 1098
total_rx_bytes, 1098
total_rx_pkt, 1098
total_tx_bytes, 1098
total_tx_pkt, 1098
unpack_qos_SLQSSetQosEventCallback
qos.h, 1875

unpack_qos_SLQSSetQosEventCallback_ind
qos.h, 1875
unpack_qos_SLQSSetQosEventCallback_ind_t, 1098
NumFlows, 1098
QosFlowInfo, 1098
unpack_qos_SLQSSetQosNWStatusCallback_ind
qos.h, 1875
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1098
status, 1099
unpack_qos_SLQSSetQosPriEventCallback_ind
qos.h, 1876
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 1099
event, 1099
unpack_qos_SLQSSetQosStatusCallback_ind
qos.h, 1877
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1099
event, 1100
id, 1100
reason, 1100
status, 1100
unpack_qos_Tos_t, 1109
mask, 1109
val, 1109
unpack_qos_dataRate_t, 1090
dataRateMax, 1090
guaranteedRate, 1090
unpack_qos_pktErrRate_t, 1092
exponent, 1092
multiplier, 1092
unpack_qos_swiQosFilter_t, 1101
EspSpi, 1103
IPv4DstAddr, 1103
IPv4SrcAddr, 1103
IPv4Tos, 1103
IPv6DstAddr, 1103
IPv6Label, 1103
IPv6SrcAddr, 1103
IPv6TrafCls, 1103
Id, 1103
index, 1103
is_EspSpi_Available, 1103
is_IPv4DstAddr_Available, 1103
is_IPv4SrcAddr_Available, 1103
is_IPv4Tos_Available, 1103
is_IPv6DstAddr_Available, 1103
is_IPv6Label_Available, 1103
is_IPv6SrcAddr_Available, 1103
is_IPv6TrafCls_Available, 1103
is_Id_Available, 1103
is_NxtHdrProto_Available, 1103
is_Precedence_Available, 1103
is_TCPDstPort_Available, 1103
is_TCPSrcPort_Available, 1103
is_TranDstPort_Available, 1104
is_TranSrcPort_Available, 1104
is_UDPDstPort_Available, 1104
is_UDPSrcPort_Available, 1104
NxtHdrProto, 1104
Precedence, 1104
TCPDstPort, 1104
TCPSrcPort, 1104
TranDstPort, 1104
TranSrcPort, 1104
UDPDstPort, 1104
UDPSrcPort, 1104
version, 1104
unpack_qos_swiQosFlow_t, 1104
DataRate, 1107
index, 1107
is_DataRate_Available, 1107
is_Jitter_Available, 1107
is_Latency_Available, 1107
is_LteQci_Available, 1107
is_MaxAllowedPktSz_Available, 1107
is_MinPolicedPktSz_Available, 1107
is_PktErrRate_Available, 1107
is_ProfileId3GPP2_Available, 1107
is_TokenBucket_Available, 1107
is_TrafficClass_Available, 1107
is_val_3GPP2Pri_Available, 1107
is_val_3GPPImCn_Available, 1107
is_val_3GPPSigInd_Available, 1107
Jitter, 1107
Latency, 1107
LteQci, 1108
MaxAllowedPktSz, 1108
MinPolicedPktSz, 1108
PktErrRate, 1108
ProfileId3GPP2, 1108
TokenBucket, 1108
TrafficClass, 1108
val_3GPP2Pri, 1108
val_3GPPImCn, 1108
val_3GPPResResidualBER, 1108
val_3GPPSigInd, 1108
val_3GPPTraHdlPri, 1108
unpack_qos_tokenBucket_t, 1108
bucketSz, 1108
peakRate, 1108
tokenRate, 1108
unpack_result_code_only
common.h, 1330
unpack_rms_GetSMSWake
rms.h, 1878
unpack_rms_GetSMSWake_t, 1110
enabled, 1110
wake_mask, 1110
unpack_rms_SetSMSWake
rms.h, 1879
unpack_sar_SLQSGetRfSarState
sar.h, 1880
unpack_sar_SLQSGetRfSarState_t, 1111
pSarRFState, 1112
unpack_sar_SLQSSetRfSarState

sar.h, 1880
unpack_sms_GetSMSCAddress
sms.h, 1895
unpack_sms_GetSMSCAddress_t, 1112
addressSize, 1112
pSMSCAddress, 1112
pSMSCType, 1113
typeSize, 1113
unpack_sms_SLQSDeleteSMS
sms.h, 1897
unpack_sms_SLQSDeleteSMS_t, 1115
unpack_sms_SLQSGetIndicationRegister
sms.h, 1897
unpack_sms_SLQSGetIndicationRegister_t, 1115
pGetIndicationRegInfo, 1115
unpack_sms_SLQSGetMessageWaiting
sms.h, 1897
unpack_sms_SLQSGetMessageWaiting_t, 1115
pGetMsgWaitingInfoResp, 1115
unpack_sms_SLQSGetSMS
sms.h, 1898
unpack_sms_SLQSGetSMS_t, 1116
message, 1116
messageFormat, 1116
messageSize, 1116
messageTag, 1116
unpack_sms_SLQSGetSMSList
sms.h, 1898
unpack_sms_SLQSGetSMSList_t, 1117
messageList, 1117
messageListSize, 1117
unpack_sms_SLQSGetSmsBroadcastConfig
sms.h, 1898
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1116
pBroadcastConfig, 1117
pCDMABroadcastConfig, 1117
unpack_sms_SLQSGetTransLayerInfo
sms.h, 1899
unpack_sms_SLQSGetTransLayerInfo_t, 1117
pGetTransLayerInfo, 1118
unpack_sms_SLQSGetTransNWRegInfo
sms.h, 1899
unpack_sms_SLQSGetTransNWRegInfo_t, 1118
unpack_sms_SLQSModifySMSStatus
sms.h, 1899
unpack_sms_SLQSModifySMSStatus_t, 1118
unpack_sms_SLQSSendAsyncSMS
sms.h, 1900
unpack_sms_SLQSSetIndicationRegister
sms.h, 1900
unpack_sms_SLQSSetSmsBroadcastActivation
sms.h, 1900
unpack_sms_SLQSSetSmsBroadcastConfig
sms.h, 1901
unpack_sms_SLQSSetSmsStorage
sms.h, 1901
unpack_sms_SLQSSmsGetMaxStorageSize
sms.h, 1901
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1118
pMaxStorageSizeResp, 1119
unpack_sms_SLQSSmsGetMessageProtocol
sms.h, 1902
unpack_sms_SLQSSmsGetMessageProtocol_t, 1119
pMessageProtocol, 1119
unpack_sms_SLQSSmsSetRoutes
sms.h, 1902
unpack_sms_SLQSSwiGetSMSStorage
sms.h, 1902
unpack_sms_SLQSSwiGetSMSStorage_t, 1119
pSmsStorage, 1120
unpack_sms_SLQSWmsMemoryFullCallBack_ind
sms.h, 1903
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1120
messageMode, 1120
storageType, 1120
unpack_sms_SaveSMS
sms.h, 1895
unpack_sms_SaveSMS_t, 1113
pMessageIndex, 1113
unpack_sms_SendSMS
sms.h, 1895
unpack_sms_SendSMS_t, 1113
messageFailureCode, 1113
messageID, 1113
unpack_sms_SetNewSMSCallback
sms.h, 1896
unpack_sms_SetNewSMSCallback_ind
sms.h, 1896
unpack_sms_SetNewSMSCallback_ind_t, 1113
ETWSTlv, 1114
IMSTlv, 1114
MMTlv, 1114
NewMMTlv, 1114
SMSCTlv, 1114
TRMessageTlv, 1115
unpack_sms_SetNewSMSCallback_t, 1115
unpack_sms_SetSMSCAddress
sms.h, 1896
unpack_swiloc_SwiLocGetAutoStart
swiloc.h, 1905
unpack_swiloc_SwiLocGetAutoStart_t, 1120
fix_rate, 1121
fix_rate_reported, 1122
fix_type, 1122
fix_type_reported, 1122
function, 1122
function_reported, 1122
max_dist, 1122
max_dist_reported, 1122
max_time, 1122
max_time_reported, 1122
unpack_swiloc_SwiLocSetAutoStart
swiloc.h, 1906
unpack_swioma_SLQSOMADMAlertCallback
swioma.h, 1911

unpack_swioma_SLQSOMADMAlertCallback_ind
swioma.h, 1912
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1122
eventType, 1122
unpack_swioma_SLQSOMADMCancelSession
swioma.h, 1912
unpack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 1913
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1123
Date, 1125
DateLength, 1125
PkgDescLength, 1125
PkgDescription, 1125
PkgName, 1125
PkgNameLength, 1125
RetryCount, 1125
SessionState, 1125
SessionType, 1125
Severity, 1125
Source, 1125
SourceLength, 1125
Status, 1125
Time, 1125
TimeLength, 1125
UpdateCompleteStatus, 1125
unpack_swioma_SLQSOMADMGetSettings
swioma.h, 1913
unpack_swioma_SLQSOMADMGetSettings_t, 1125
Autosdm, 1127
FOTAdownload, 1127
FwAutoCheck, 1127
unpack_swioma_SLQSOMADMSendSelection
swioma.h, 1914
unpack_swioma_SLQSOMADMSetSettings
swioma.h, 1914
unpack_swioma_SLQSOMADMStartSession
swioma.h, 1914
unpack_swioma_SLQSOMADMStartSession_t, 1127
FwAvailability, 1127
unpack_uim_ChangePin
uim.h, 1926
unpack_uim_ChangePin_t, 1127
pEncryptedPIN1, 1128
pIndicationToken, 1128
pRemainingRetries, 1128
Tlvresult, 1128
unpack_uim_GetCardStatus
uim.h, 1926
unpack_uim_GetCardStatus_t, 1128
pCardStatus, 1128
pHotSwapStatus, 1129
Tlvresult, 1129
unpack_uim_ReadTransparent
uim.h, 1926
unpack_uim_ReadTransparent_t, 1129
pCardResult, 1129
pEncryptedData, 1129
pIndicationToken, 1129
pReadResult, 1129
Tlvresult, 1129
unpack_uim_SLQSUIMAuthenticate
uim.h, 1927
unpack_uim_SLQSUIMAuthenticate_t, 1131
pAuthenticateResult, 1131
pCardResult, 1131
pIndicationToken, 1131
unpack_uim_SLQSUIMDepersonalization
uim.h, 1928
unpack_uim_SLQSUIMDepersonalization_t, 1131
pRemainingRetries, 1132
unpack_uim_SLQSUIMEventRegister
uim.h, 1928
unpack_uim_SLQSUIMEventRegister_t, 1132
eventMask, 1132
unpack_uim_SLQSUIMGetConfiguration
uim.h, 1928
unpack_uim_SLQSUIMGetConfiguration_t, 1132
pAutoSelection, 1133
pHaltSubscription, 1133
pPersonalizationStatus, 1133
unpack_uim_SLQSUIMGetFileAttributes
uim.h, 1929
unpack_uim_SLQSUIMGetFileAttributes_t, 1133
pCardResult, 1134
pFileAttributes, 1134
pIndicationToken, 1134
unpack_uim_SLQSUIMGetSlotsStatus
uim.h, 1929
unpack_uim_SLQSUIMGetSlotsStatus_t, 1134
pNumberOfPhySlot, 1134
pUimSlotsStatus, 1134
unpack_uim_SLQSUIMPowerDown
uim.h, 1930
unpack_uim_SLQSUIMPowerUp
uim.h, 1930
unpack_uim_SLQSUIMRefreshComplete
uim.h, 1930
unpack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 1930
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 1134
pRefreshEvent, 1135
unpack_uim_SLQSUIMRefreshOK
uim.h, 1931
unpack_uim_SLQSUIMRefreshRegister
uim.h, 1931
unpack_uim_SLQSUIMReset
uim.h, 1931
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind
uim.h, 1932
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t, 1135
unpack_uim_SLQSUIMSwitchSlot
uim.h, 1932
unpack_uim_SetPinProtection
uim.h, 1927

unpack_uim_SetPinProtection_t, 1130
pEncryptedPIN1, 1130
pIndicationToken, 1130
pRemainingRetries, 1130
Tlvresult, 1130
unpack_uim_SetUimSlotStatusChangeCallback_ind
uim.h, 1927
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 1130
bNumberOfPhySlots, 1131
slotsstatusChange, 1131
unpack_uim_UnblockPin
uim.h, 1933
unpack_uim_UnblockPin_t, 1135
pEncryptedPIN1, 1136
pIndicationToken, 1136
pRemainingRetries, 1136
Tlvresult, 1136
unpack_uim_VerifyPin
uim.h, 1933
unpack_uim_VerifyPin_t, 1136
pEncryptedPIN1, 1136
pIndicationToken, 1136
pRemainingRetries, 1136
Tlvresult, 1136
unpack_voice_AnswerUSSD
voice.h, 1949
unpack_voice_CancelUSSD
voice.h, 1950
unpack_voice_OriginateUSSD
voice.h, 1950
unpack_voice_SLQSOriginateUSSD
voice.h, 1950
unpack_voice_SLQSOriginateUSSD_t, 1136
pAlphaIDInfo, 1137
pCCSuppsType, 1137
pCallId, 1137
pCcResultType, 1137
pUSSDInfo, 1137
pfailureCause, 1137
unpack_voice_SLQSVoiceALSSelectLine
voice.h, 1951
unpack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 1951
unpack_voice_SLQSVoiceAnswerCall
voice.h, 1951
unpack_voice_SLQSVoiceAnswerCall_t, 1137
pCallId, 1137
unpack_voice_SLQSVoiceBindSubscription
voice.h, 1952
unpack_voice_SLQSVoiceBurstDTMF
voice.h, 1952
unpack_voice_SLQSVoiceBurstDTMF_t, 1137
pCallID, 1138
unpack_voice_SLQSVoiceDialCall
voice.h, 1952
unpack_voice_SLQSVoiceDialCall_t, 1138
pAlphaIDInfo, 1138
pCCResultType, 1139
pCCSUPSType, 1139
pCallID, 1138
unpack_voice_SLQSVoiceEndCall
voice.h, 1953
unpack_voice_SLQSVoiceEndCall_t, 1139
pCallId, 1139
unpack_voice_SLQSVoiceGetAllCallInfo
voice.h, 1953
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1139
pArrAlertingPattern, 1141
pArrAlertingType, 1141
pArrAlphaID, 1141
pArrCallEndReason, 1141
pArrCallInfo, 1141
pArrCalledPartyNum, 1141
pArrConnectPartyNum, 1141
pArrDiagInfo, 1141
pArrRedirPartyNum, 1141
pArrRemotePartyName, 1141
pArrRemotePartyNum, 1141
pArrSvcOption, 1141
pArrUUSInfo, 1142
pVoicePrivacy, 1142
unpack_voice_SLQSVoiceGetCLIP
voice.h, 1955
unpack_voice_SLQSVoiceGetCLIP_t, 1149
pAlphaIDInfo, 1150
pCCResType, 1150
pCallID, 1150
pFailCause, 1150
unpack_voice_SLQSVoiceGetCLIR
voice.h, 1955
unpack_voice_SLQSVoiceGetCLIR_t, 1150
pAlphaIDInfo, 1151
pCCResType, 1151
pCallID, 1151
pFailCause, 1151
unpack_voice_SLQSVoiceGetCNAP
voice.h, 1955
unpack_voice_SLQSVoiceGetCNAP_t, 1151
pAlphaIDInfo, 1152
pCCResType, 1152
pCallID, 1152
pFailCause, 1152
unpack_voice_SLQSVoiceGetCOLP
voice.h, 1956
unpack_voice_SLQSVoiceGetCOLP_t, 1152
pAlphaIDInfo, 1153
pCCResType, 1153
pCallID, 1153
pFailCause, 1154
unpack_voice_SLQSVoiceGetCOLR
voice.h, 1956
unpack_voice_SLQSVoiceGetCOLR_t, 1154
pAlphaIDInfo, 1155
pCCResType, 1155
pCallID, 1155

pFailCause, 1155
unpack_voice_SLQSVoiceGetCallBarring
voice.h, 1953
unpack_voice_SLQSVoiceGetCallBarring_t, 1142
pAlphaIDInfo, 1143
pCCResType, 1143
pCallID, 1143
pFailCause, 1143
pSvcClass, 1143
unpack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 1954
unpack_voice_SLQSVoiceGetCallForwardingStatus_t, 1143
pAlphaIDInfo, 1144
pCCResType, 1144
pCallID, 1144
pFailCause, 1144
unpack_voice_SLQSVoiceGetCallInfo
voice.h, 1954
unpack_voice_SLQSVoiceGetCallInfo_t, 1144
pAlertType, 1147
pAlertingPattern, 1147
pAlphaIDInfo, 1147
pCallInfo, 1147
pConnectNumInfo, 1147
pDiagInfo, 1147
pOTASPStatus, 1147
pRemotePartyName, 1147
pRemotePartyNum, 1147
pSrvOpt, 1147
pUUSInfo, 1147
pVoicePrivacy, 1147
unpack_voice_SLQSVoiceGetCallWaiting
voice.h, 1954
unpack_voice_SLQSVoiceGetCallWaiting_t, 1147
pAlphaIDInfo, 1148
pCCResType, 1148
pCallID, 1148
pFailCause, 1148
pSvcClass, 1148
unpack_voice_SLQSVoiceGetConfig
voice.h, 1956
unpack_voice_SLQSVoiceGetConfig_t, 1155
pAirTimerCnt, 1156
pAutoAnswerStat, 1156
pCurAMRConfig, 1156
pCurPrefVoiceSO, 1156
pCurVoiceDomainPref, 1156
pCurVoicePrivacyPref, 1157
pCurrTTYMode, 1156
pRoamTimerCnt, 1157
unpack_voice_SLQSVoiceIndicationRegister
voice.h, 1957
unpack_voice_SLQSVoiceManageCalls
voice.h, 1957
unpack_voice_SLQSVoiceManageCalls_t, 1157
pFailCause, 1157
unpack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 1957
unpack_voice_SLQSVoiceSendFlash
voice.h, 1958
unpack_voice_SLQSVoiceSendFlash_t, 1157
pCallID, 1157
unpack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 1958
unpack_voice_SLQSVoiceSetCallBarringPassword_t, 1157
pAlphaIDInfo, 1158
pCCResType, 1158
pCallID, 1158
pFailCause, 1159
unpack_voice_SLQSVoiceSetConfig
voice.h, 1958
unpack_voice_SLQSVoiceSetConfig_t, 1159
pAirTimerStatus, 1160
pAutoAnsStatus, 1160
pPrefVoiceSOStatus, 1160
pRoamTimerStatus, 1160
pTTYConfigStatus, 1160
pVoiceDomainPrefStatus, 1160
unpack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 1959
unpack_voice_SLQSVoiceSetSUPSService
voice.h, 1959
unpack_voice_SLQSVoiceSetSUPSService_t, 1160
pCallID, 1161
pFailCause, 1161
unpack_voice_SLQSVoiceStartContDTMF
voice.h, 1959
unpack_voice_SLQSVoiceStartContDTMF_t, 1161
pCallID, 1162
unpack_voice_SLQSVoiceStopContDTMF
voice.h, 1960
unpack_voice_SLQSVoiceStopContDTMF_t, 1162
callID, 1162
unpack_wds_DHCPv4ClientLease_ind
wds.h, 1985
unpack_wds_DHCPv4ClientLease_ind_t, 1162
DHCPv4LeaseOptTlv, 1162
DHCPv4LeaseStateTlv, 1162
IPv4AddrTlv, 1163
ProfileIdTlv, 1163
unpack_wds_DHCPv4ClientLeaseChange
wds.h, 1985
unpack_wds_GetAutoconnect
wds.h, 1985
unpack_wds_GetAutoconnect_t, 1163
psetting, 1163
unpack_wds_GetByteTotals
wds.h, 1986
unpack_wds_GetByteTotals_t, 1163
pRXTotalBytes, 1163
pTXTotalBytes, 1163
unpack_wds_GetConnectionRate
wds.h, 1986
unpack_wds_GetConnectionRate_t, 1164

currentChannelRXRate, 1164
currentChannelTXRate, 1164
maxChannelRXRate, 1164
maxChannelTXRate, 1164
unpack_wds_GetDataBearerTechnology
wds.h, 1986
unpack_wds_GetDataBearerTechnology_t, 1164
pDataBearer, 1165
unpack_wds_GetDefaultProfile
wds.h, 1987
unpack_wds_GetDefaultProfile_t, 1165
apnname, 1166
apnsize, 1166
auth, 1166
ipaddr, 1166
ipaddrv6, 1166
name, 1166
namesize, 1166
pdptype, 1166
pridns, 1166
pridnsv6, 1166
secdns, 1166
secdnsv6, 1166
username, 1166
usersize, 1166
unpack_wds_GetDefaultProfileNum
wds.h, 1987
unpack_wds_GetDefaultProfileNum_t, 1166
index, 1167
unpack_wds_GetDormancyState
wds.h, 1987
unpack_wds_GetDormancyState_t, 1167
dormancyState, 1167
unpack_wds_GetLastMobileIPError
wds.h, 1988
unpack_wds_GetLastMobileIPError_t, 1167
error, 1167
unpack_wds_GetMobileIP
wds.h, 1988
unpack_wds_GetMobileIP_t, 1167
mipMode, 1168
unpack_wds_GetMobileIPProfile
wds.h, 1988
unpack_wds_GetMobileIPProfile_t, 1168
AAASPI, 1168
AAAState, 1168
address, 1168
enabled, 1168
HASPI, 1168
HAState, 1168
NAI, 1168
naiSize, 1168
primaryHA, 1168
revTunneling, 1169
secondaryHA, 1169
unpack_wds_GetPacketStatistics
wds.h, 1989
unpack_wds_GetPacketStatistics_t, 1169
pRXDroppedCount, 1170
pRXOKBytesLastCall, 1170
pRXOkBytesCount, 1170
pRXPacketErrors, 1170
pRXPacketOverflows, 1170
pRXPacketSuccesses, 1170
pTXDroppedCount, 1170
pTXOKBytesLastCall, 1170
pTXOkBytesCount, 1170
pTXPacketErrors, 1170
pTXPacketOverflows, 1170
pTXPacketSuccesses, 1170
unpack_wds_GetPacketStatus
wds.h, 1989
unpack_wds_GetPacketStatus_t, 1170
rXDroppedCount, 1171
rXOKBytesLastCall, 1171
rXOkBytesCount, 1171
rXPacketErrors, 1171
rXPacketOverflows, 1171
rXPacketSuccesses, 1171
tXDroppedCount, 1171
tXOKBytesLastCall, 1171
tXOkBytesCount, 1171
tXPacketErrors, 1171
tXPacketOverflows, 1171
tXPacketSuccesses, 1171
unpack_wds_GetSessionDuration
wds.h, 1989
unpack_wds_GetSessionDuration_t, 1171
callDuration, 1172
unpack_wds_GetSessionState
wds.h, 1990
unpack_wds_GetSessionState_t, 1172
connectionStatus, 1172
unpack_wds_RMSetTransferStatistics
wds.h, 1990
unpack_wds_RMSetTransferStatistics_t, 1172
unpack_wds_RMTransferStatistics_ind
wds.h, 1990
unpack_wds_SLQSCreateProfile
wds.h, 1993
unpack_wds_SLQSCreateProfile_t, 1172
pCreateProfileOut, 1172
pProfileID, 1172
Tlvresult, 1173
unpack_wds_SLQSDeleteProfile
wds.h, 1993
unpack_wds_SLQSDeleteProfile_t, 1173
extendedErrorCode, 1173
unpack_wds_SLQSGet3GPPConfigItem
wds.h, 1993
unpack_wds_SLQSGet3GPPConfigItem_t, 1173
_3gppRelease, 1174
defaultPDNEnabled, 1174
LTEAttachProfile, 1174
profileList, 1174
unpack_wds_SLQSGetCurrDataSystemStat

wds.h, 1994
unpack_wds_SLQSGetCurrDataSystemStat_t, 1174
currNetworkInfo, 1175
networkInfoLen, 1175
prefNetwork, 1175
unpack_wds_SLQSGetCurrentChannelRate
wds.h, 1994
unpack_wds_SLQSGetCurrentChannelRate_t, 1175
max_channel_rx_rate, 1176
max_channel_tx_rate, 1176
unpack_wds_SLQSGetDUNCallInfo
wds.h, 1995
unpack_wds_SLQSGetDUNCallInfo_t, 1176
callEndReason, 1177
channelRate, 1177
connectionStatus, 1177
dataBearerTech, 1177
dormancyStatus, 1177
lastCallDataBearerTech, 1177
mdmCallDurationActive, 1177
rxOKBytesCount, 1177
txOKBytesCount, 1177
unpack_wds_SLQSGetDataBearerTechnology
wds.h, 1994
unpack_wds_SLQSGetDataBearerTechnology_t, 1176
curDataBearerTechnology, 1176
dataBearerMask, 1176
lastCallDataBearerTechnology, 1176
unpack_wds_SLQSGetProfileSettings
wds.h, 1995
unpack_wds_SLQSGetProfileSettings_t, 1177
pProfileSettings, 1177
ProfileType, 1178
Tlvresult, 1178
unpack_wds_SLQSGetRuntimeSettings
wds.h, 1995
unpack_wds_SLQSGetRuntimeSettings_t, 1178
APNName, 1179
Authentication, 1179
DomainList, 1179
GPRSGrantedQoS, 1179
GWAddressV4, 1179
IMCNflag, 1179
IPFamilyPreference, 1179
IPV6AddrInfo, 1179
IPV6GWAddrInfo, 1179
IPv4, 1179
Mtu, 1179
PDPType, 1179
PrimaryDNSV4, 1179
PrimaryDNSV6, 1179
ProfileID, 1179
ProfileName, 1179
SecondaryDNSV4, 1179
SecondaryDNSV6, 1180
ServerAddrList, 1180
SubnetMaskV4, 1180
Technology, 1180
UMTSGrantedQoS, 1180
Username, 1180
unpack_wds_SLQSModifyProfile
wds.h, 1996
unpack_wds_SLQSModifyProfile_t, 1180
pExtErrorCode, 1180
unpack_wds_SLQSResetPacketStatics
wds.h, 1996
unpack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 1998
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1183
pHwConfig, 1183
unpack_wds_SLQSSGetLoopback
wds.h, 1998
unpack_wds_SLQSSGetLoopback_t, 1184
ByteLoopbackMode, 1184
ByteLoopbackMultiplier, 1184
unpack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 1999
unpack_wds_SLQSSSetLoopback
wds.h, 1999
unpack_wds_SLQSSet3GPPConfigItem
wds.h, 1996
unpack_wds_SLQSSetIPFamilyPreference
wds.h, 1997
unpack_wds_SLQSSetIPFamilyPreference_t, 1180
Tlvresult, 1180
unpack_wds_SLQSSetPacketSrvStatusCallback
wds.h, 1997
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1180
bearerID, 1181
conn_status, 1181
ipFamily, 1181
reconfigReqd, 1181
sessionEndReason, 1181
techName, 1181
verboseSessnEndReason, 1181
verboseSessnEndReasonType, 1181
unpack_wds_SLQSSetWdsEventCallback
wds.h, 1997
unpack_wds_SLQSSetWdsEventCallback_ind
wds.h, 1998
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1181
currDBTechAvail, 1182
currNWInfo, 1182
dBTechAvail, 1182
dBTechnology, 1183
dBtechExtAvail, 1183
dBtechnologyExt, 1183
dataSysStatAvail, 1182
dormancyStatAvail, 1183
dormancyStatus, 1183
mipStatus, 1183
mipstatAvail, 1183
netInfoLen, 1183
prefNetwork, 1183
ratMask, 1183

rx_bytes, 1183
rx_pkts, 1183
soMask, 1183
tx_bytes, 1183
tx_pkts, 1183
xferStatAvail, 1183
unpack_wds_SLQSStartDataSession
wds.h, 1999
unpack_wds_SLQSStartDataSession_t, 1184
pFailureReason, 1185
pVerboseFailReasonType, 1185
pVerboseFailureReason, 1185
psid, 1185
unpack_wds_SLQSStopDataSession
wds.h, 2000
unpack_wds_SLQSWdsGoActive
wds.h, 2000
unpack_wds_SLQSWdsGoDormant
wds.h, 2000
unpack_wds_SLQSWdsSetEventReport
wds.h, 2001
unpack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 2001
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1185
apnName, 1186
bearerId, 1186
contextId, 1186
ipv4Address, 1186
ipv4GWAddress, 1186
ipv6Address, 1186
ipv6GWAddress, 1186
unpack_wds_SetAutoconnect
wds.h, 1991
unpack_wds_SetDefaultProfile
wds.h, 1991
unpack_wds_SetDefaultProfileNum
wds.h, 1991
unpack_wds_SetMobileIP
wds.h, 1992
unpack_wds_SetMobileIPParameters
wds.h, 1992
unpack_wds_SetMobileIPProfile
wds.h, 1992
unpack_wds_SetMobileIPProfile_t, 1172
UnpackQmiProfileInfo
wds.h, 1966
unpackWdsProfileParam, 1187
SlqsProfile3GPP, 1187
SlqsProfile3GPP2, 1187
upLink
NSSAudioCtrl, 568
UpdateCompleteStatus
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1125
updateCompleteStatus
omaDmFotaTlv, 573
unpack_omaDmFotaTlv_t, 1087
upgrade_mc77xx_fw
qaGobiApiFms.h, 1602
UpgradeFirmware2k
qaGobiApiFms.h, 1602
upinRetries
slotInf, 836
slotInfo, 838
uim_slotInfo, 956
upinState
slotInf, 837
slotInfo, 838
uim_slotInfo, 956
UpkQmiCbkCatEventReportInd
qaCbkCatEventReportInd.h, 1447
UpkQmiCbkSwiOmaDmEventReportInd
qaCbkSwiOmaDmEventReportInd.h, 1448
UpkQmiCbkSwiOmaDmEventReportIndExt
qaCbkSwiOmaDmEventReportInd.h, 1448
UpkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 1861
UpkQmiNasPerformNetworkScan
qaNasPerformNetworkScan.h, 1862
upukRetries
slotInf, 837
slotInfo, 838
uim_slotInfo, 956
usageMask
loc_sensorDataUsage, 368
sensorDataUsage_s, 800
t_sensor, 912
User Identity Module Service (UIM), 52
userData
SMSAsyncRawSend_s, 869
userInputReq
omaDmConfigTlv, 569
omaDmConfigTlvExt, 571
omaDmFotaTlv, 573
unpack_omaDmConfigTlv_t, 1085
unpack_omaDmFotaTlv_t, 1087
userInputTimeout
omaDmConfigTlv, 569
omaDmConfigTlvExt, 571
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
unpack_omaDmConfigTlv_t, 1085
unpack_omaDmFotaTlv_t, 1087
Username
unpack_wds_SLQSGetRuntimeSettings_t, 1180
username
unpack_wds_GetDefaultProfile_t, 1166
usersize
unpack_wds_GetDefaultProfile_t, 1166
ussDCS
pack_voice_SLQSOriginateUSSD_t, 661
USSInfo, 1192
voice_USSInfo, 1230
ussData
pack_voice_SLQSOriginateUSSD_t, 661

USSInfo, 1192
voice_USSInfo, 1229
ussLen
pack_voice_SLQSOriginateUSSD_t, 661
USSInfo, 1192
voice_USSInfo, 1230
uusInfo
allCallsUUSInfo, 118
voice_allCallsUUSInfo, 1196

v4sessionId
qaQmiInterfaceInfo, 735
ssdatasession_params, 889
WdsRunTimeSettings, 1321
v6sessionId
qaQmiInterfaceInfo, 735
ssdatasession_params, 889
WdsRunTimeSettings, 1321
VOICE_SUPS_SRV_CLASS_DATA
qaGobiApiVoice.h, 1801
VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC
qaGobiApiVoice.h, 1802
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC
qaGobiApiVoice.h, 1802
VOICE_SUPS_SRV_CLASS_FAX
qaGobiApiVoice.h, 1801
VOICE_SUPS_SRV_CLASS_NONE
qaGobiApiVoice.h, 1801
VOICE_SUPS_SRV_CLASS_PACKETACCESS
qaGobiApiVoice.h, 1802
VOICE_SUPS_SRV_CLASS_PADACCESS
qaGobiApiVoice.h, 1802
VOICE_SUPS_SRV_CLASS_SMS
qaGobiApiVoice.h, 1802
VOICE_SUPS_SRV_CLASS_VOICE
qaGobiApiVoice.h, 1801
VDOP
loc_precisionDilution, 365
precisionDilution_s, 718
VOICE_SRV
qaGobiApiCbk.h, 1463
val
IPv6TrafCls, 328
Tos, 927
unpack_qos_IPv6TrafCls_t, 1092
unpack_qos_Tos_t, 1109
val_3GPP2Pri
unpack_qos_swiQosFlow_t, 1108
val_3GPPImCn
unpack_qos_swiQosFlow_t, 1108
val_3GPPResResidualBER
unpack_qos_swiQosFlow_t, 1108
val_3GPPSigInd
unpack_qos_swiQosFlow_t, 1108
val_3GPPTraHdlPri
unpack_qos_swiQosFlow_t, 1108
ValidMask
GPSStateInfo, 288
validMask
loc_satelliteInfo, 367
satelliteInfo, 795
ValidateSPC
qaGobiApiDms.h, 1583
ValidityCW0
LteCQIParm, 383
unpack_nas_SLQSSwiGetLteCQI_t, 1080
ValidityCW1
LteCQIParm, 383
unpack_nas_SLQSSwiGetLteCQI_t, 1080
Value
GetM2MSpkrGainResp, 276
SetM2MSpkrGainReq, 822
value
pack_dms_UIMSetPINProtection_t, 587
pack_dms_UIMVerifyPIN_t, 589
value_length
custSettingInfo, 206
DMScustSettingInfo, 224
pack_dms_SetCustFeaturesV2_t, 581
setCustomSettingV2, 812
verbFailReason
ssdatasession_params, 889
verbFailReasonType
ssdatasession_params, 889
verboseSessnEndReason
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-
t, 1181
verboseSessnEndReasonType
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-
t, 1181
verifyLeft
personalizationStatus, 707
remainingRetries, 777
uim_personalizationStatus, 948
uim_remainingRetries, 952
verifyPIN
pack_uim_VerifyPin_t, 660
UIMVerifyPinReq, 980
verifyRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1021
unpack_dms_UIMSetControlKeyProtection_t, 1024
unpack_dms_UIMSetPINProtection_t, 1024
verifyUIMPIN, 1194
pinID, 1194
pinLen, 1194
pinVal, 1194
version
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
swiQosFilter, 904
unpack_omaDmFotaTlv_t, 1087
unpack_qos_swiQosFilter_t, 1104
versionString
pack_dms_SLQSSwiSetOSInfo_t, 584
unpack_dms_SLQSSwiGetOSInfo_t, 1018

versionlength
omaDmFotaTlv, 573
omaDmFotaTlvExt, 575
unpack_omaDmFotaTlv_t, 1087
vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 600
vertReliability
pack_loc_SLQSLOCInjectPosition_t, 600
vertUnc
pack_loc_SLQSLOCInjectPosition_t, 600
VerticalUncertainty
GPSStateInfo, 288
VirtStream
nas_protocolSubtypeElement, 485
protocolSubtypeElement, 733
Voice Service (VOICE), 50
voice.h, 1933
MAXVOICEUSSDLENGTH, 1937
pack_voice_AnswerUSSD, 1938
pack_voice_CancelUSSD, 1938
pack_voice_OriginateUSSD, 1938
pack_voice_SLQSOriginateUSSD, 1939
pack_voice_SLQSVoiceALSSelectLine, 1939
pack_voice_SLQSVoiceALSSetLineSwitching, 1939
pack_voice_SLQSVoiceAnswerCall, 1940
pack_voice_SLQSVoiceBindSubscription, 1940
pack_voice_SLQSVoiceBurstDTMF, 1940
pack_voice_SLQSVoiceDialCall, 1941
pack_voice_SLQSVoiceEndCall, 1941
pack_voice_SLQSVoiceGetAllCallInfo, 1942
pack_voice_SLQSVoiceGetCLIP, 1943
pack_voice_SLQSVoiceGetCLIR, 1944
pack_voice_SLQSVoiceGetCNAP, 1944
pack_voice_SLQSVoiceGetCOLP, 1944
pack_voice_SLQSVoiceGetCOLR, 1945
pack_voice_SLQSVoiceGetCallBarring, 1942
pack_voice_SLQSVoiceGetCallForwardingStatus, 1942
pack_voice_SLQSVoiceGetCallInfo, 1943
pack_voice_SLQSVoiceGetCallWaiting, 1943
pack_voice_SLQSVoiceGetConfig, 1945
pack_voice_SLQSVoiceIndicationRegister, 1945
pack_voice_SLQSVoiceManageCalls, 1946
pack_voice_SLQSVoiceOrigUSSDNoWait, 1946
pack_voice_SLQSVoiceSendFlash, 1947
pack_voice_SLQSVoiceSetCallBarringPassword, 1947
pack_voice_SLQSVoiceSetConfig, 1947
pack_voice_SLQSVoiceSetPreferredPrivacy, 1948
pack_voice_SLQSVoiceSetSUPSService, 1948
pack_voice_SLQSVoiceStartContDTMF, 1948
pack_voice_SLQSVoiceStopContDTMF, 1949
unpack_voice_AnswerUSSD, 1949
unpack_voice_CancelUSSD, 1950
unpack_voice_OriginateUSSD, 1950
unpack_voice_SLQSOriginateUSSD, 1950
unpack_voice_SLQSVoiceALSSelectLine, 1951
unpack_voice_SLQSVoiceALSSetLineSwitching, 1951
unpack_voice_SLQSVoiceAnswerCall, 1951
unpack_voice_SLQSVoiceBindSubscription, 1952
unpack_voice_SLQSVoiceBurstDTMF, 1952
unpack_voice_SLQSVoiceDialCall, 1952
unpack_voice_SLQSVoiceEndCall, 1953
unpack_voice_SLQSVoiceGetAllCallInfo, 1953
unpack_voice_SLQSVoiceGetCLIP, 1955
unpack_voice_SLQSVoiceGetCLIR, 1955
unpack_voice_SLQSVoiceGetCNAP, 1955
unpack_voice_SLQSVoiceGetCOLP, 1956
unpack_voice_SLQSVoiceGetCOLR, 1956
unpack_voice_SLQSVoiceGetCallBarring, 1953
unpack_voice_SLQSVoiceGetCallForwarding-Status, 1954
unpack_voice_SLQSVoiceGetCallInfo, 1954
unpack_voice_SLQSVoiceGetCallWaiting, 1954
unpack_voice_SLQSVoiceGetConfig, 1956
unpack_voice_SLQSVoiceIndicationRegister, 1957
unpack_voice_SLQSVoiceManageCalls, 1957
unpack_voice_SLQSVoiceOrigUSSDNoWait, 1957
unpack_voice_SLQSVoiceSendFlash, 1958
unpack_voice_SLQSVoiceSetCallBarringPassword, 1958
unpack_voice_SLQSVoiceSetConfig, 1958
unpack_voice_SLQSVoiceSetPreferredPrivacy, 1959
unpack_voice_SLQSVoiceSetSUPSService, 1959
unpack_voice_SLQSVoiceStartContDTMF, 1959
unpack_voice_SLQSVoiceStopContDTMF, 1960
voice_CLIPResp, 1212
ActiveStatus, 1213
ProvisionStatus, 1213
voice_CLIRResp, 1213
ActiveStatus, 1214
ProvisionStatus, 1214
voice_CNAPResp, 1214
ActiveStatus, 1214
ProvisionStatus, 1214
voice_COLPResp, 1215
ActiveStatus, 1215
ProvisionStatus, 1215
voice_COLRResp, 1215
ActiveStatus, 1216
ProvisionStatus, 1216
voice_CUGInfo, 1218
CUGIndex, 1218
SuppOA, 1218
SuppPrefCUG, 1218
voice_DTMFLengths, 1220
DTMFInterdigitInterval, 1220
DTMFPulseWidth, 1220
voice_USSInfo, 1229
ussDCS, 1230
ussData, 1229
ussLen, 1230
voice_UUSInfo, 1230

UUSData, 1231
UUSDatalen, 1231
UUSDcs, 1231
UUSType, 1231
voice_airTimer, 1194
airTimerValue, 1195
namID, 1195
voice_allCallsAlphaIDInfo, 1195
AlphaIDInfo, 1195
callID, 1195
voice_allCallsDiagInfo, 1195
callID, 1196
DiagInfo, 1196
voice_allCallsUUSInfo, 1196
callID, 1196
uusInfo, 1196
voice_alphaIDInfo, 1196
alphaDcs, 1197
alphaLen, 1197
alphaText, 1197
voice_arrAlertingPattern, 1197
alertingPattern, 1198
callID, 1198
numInstances, 1198
voice_arrAlertingType, 1198
AlertingType, 1199
callID, 1199
numInstances, 1199
voice_arrAlphaID, 1199
allCallsAlphaIDInfoArr, 1199
numInstances, 1199
voice_arrCallEndReason, 1200
callEndReason, 1200
callID, 1200
numInstances, 1200
voice_arrCallInfo, 1200
getAllCallInfo, 1201
numInstances, 1201
voice_arrCalledPartyNum, 1199
CalledPartyNum, 1200
numInstances, 1200
voice_arrConnectPartyNum, 1201
ConnectedPartyNum, 1201
numInstances, 1201
voice_arrDiagInfo, 1202
DiagInfo, 1202
numInstances, 1202
voice_arrRedirPartyNum, 1202
numInstances, 1202
RedirPartyNum, 1203
voice_arrRemotePartyName, 1203
GetAllCallRmtPtyName, 1203
numInstances, 1203
voice_arrRemotePartyNum, 1203
numInstances, 1204
RmtPtyNum, 1204
voice_arrSvcOption, 1204
callID, 1204
numInstances, 1204
srvOption, 1204
voice_arrUUSInfo, 1204
AllCallsUUSInfo, 1205
numInstances, 1205
voice_burstDTMFInfo, 1205
digitCnt, 1205
pCallID, 1205
pDigitBuff, 1205
voice_callFWExtInfo, 1207
noReplyTimer, 1209
numLen, 1209
numPlan, 1209
numType, 1209
number, 1209
PI, 1209
SI, 1209
SvcClass, 1209
SvcStatus, 1209
voice_callFWInfo, 1209
noReplyTimer, 1210
numLen, 1210
number, 1210
SvcClass, 1210
SvcStatus, 1210
voice_callFwdTypeAndPlan, 1206
numberPlan, 1207
numberType, 1207
voice_callInfo, 1210
callID, 1211
callState, 1211
callType, 1212
direction, 1212
mode, 1212
voice_calledPartySubAdd, 1206
extBit, 1206
oddEvenInd, 1206
subAddr, 1206
subAddrLen, 1206
subAddrType, 1206
voice_ccSUPSType, 1212
reason, 1212
svcType, 1212
voice_connectNumInfo, 1216
callerID, 1217
callerIDLen, 1217
numPlan, 1217
numPresInd, 1217
numType, 1218
screeningInd, 1218
voice_curAMRConfig, 1218
gsmAmrStat, 1219
wcdmaAmrStat, 1219
voice_diagInfo, 1219
diagInfoLen, 1219
diagnosticInfo, 1219
voice_getAllCallInformation, 1220
ALS, 1221

CallInfo, 1221
isEmpty, 1221
voice_getAllCallRmtPtyName, 1221
callID, 1221
RemotePartyName, 1221
voice_getAllCallRmtPtyNum, 1221
callID, 1222
RemotePartyNum, 1222
voice_getCallFWExtInfo, 1222
CallFWExtInfo, 1222
numInstances, 1222
voice_getCallFWInfo, 1223
CallFWInfo, 1223
numInstances, 1223
voice_peerNumberInfo, 1223
callID, 1225
numLen, 1225
numPI, 1225
numPlan, 1225
numSI, 1225
numType, 1225
number, 1225
voice_prefVoiceSO, 1225
evrcCapability, 1227
homeOrigVoiceSO, 1227
homePageVoiceSO, 1227
namID, 1227
roamOrigVoiceSO, 1227
voice_remotePartyName, 1227
callerName, 1227
codingScheme, 1228
nameLen, 1228
namePI, 1228
voice_remotePartyNum, 1228
numLen, 1228
presentationInd, 1228
remPartyNumber, 1228
voice_roamTimer, 1228
namID, 1229
roamTimerValue, 1229
voiceALSSelectLineInfo, 1231
lineValue, 1231
voiceALSSetLineSwitchInfo, 1231
switchOption, 1231
voiceAnswerCall, 1232
pCallId, 1232
voiceBindSubscriptionInfo, 1232
subsType, 1232
voiceBurstDTMFInfo, 1232
BurstDTMFInfo, 1233
pBurstDTMFLengths, 1233
voiceCallInfoReq, 1233
callID, 1233
voiceCallInfoResp, 1233
pAlertType, 1236
pAlertingPattern, 1236
pAlphaIDInfo, 1236
pCallInfo, 1236
pConnectNumInfo, 1236
pDiagInfo, 1236
pOTASPStatus, 1236
pRemotePartyName, 1236
pRemotePartyNum, 1236
pSrvOpt, 1236
pUUSInfo, 1236
pVoicePrivacy, 1236
voiceCallRequestParams, 1236
callNumber, 1238
pCLIRType, 1238
pCUGInfo, 1238
pCallPartySubAdd, 1238
pCallType, 1238
pEmergencyCategory, 1238
pSvcType, 1238
pUUSInFo, 1238
voiceCallResponseParams, 1238
pAlphaIDInfo, 1239
pCCResultType, 1239
pCCSUPSType, 1239
pCallID, 1239
voiceContDTMFinfo, 1239
DTMFdigit, 1239
pCallID, 1239
voiceDTMFEventInfo, 1240
DTMFInformation, 1240
pOffLength, 1240
pOnLength, 1240
voiceFlashInfo, 1240
pCallID, 1241
pFlashPayLd, 1241
pFlashType, 1241
voiceGetAllCallInfo, 1241
pArrAlertingPattern, 1243
pArrAlertingType, 1243
pArrAlphaID, 1243
pArrCallEndReason, 1243
pArrCallInfo, 1243
pArrCalledPartyNum, 1243
pArrConnectPartyNum, 1243
pArrDiagInfo, 1243
pArrRedirPartyNum, 1243
pArrRemotePartyName, 1243
pArrRemotePartyNum, 1243
pArrSvcOption, 1243
pArrUUSInfo, 1244
pOTASPStatus, 1244
pVoicePrivacy, 1244
voiceGetCLIPResp, 1249
pAlphaIDInfo, 1250
pCCResType, 1250
pCCSUPSType, 1250
pCLIPResp, 1250
pCallID, 1250
pFailCause, 1250
voiceGetCLIRResp, 1250
pAlphaIDInfo, 1251

pCCResType, 1252
pCCSUPSType, 1252
pCLIRResp, 1252
pCallID, 1251
pFailCause, 1252
voiceGetCNAPResp, 1252
pAlphaIDInfo, 1253
pCCResType, 1253
pCCSUPSType, 1253
pCNAPResp, 1253
pCallID, 1253
pFailCause, 1253
voiceGetCOLPResp, 1253
pAlphaIDInfo, 1254
pCCResType, 1254
pCCSUPSType, 1254
pCOLPResp, 1254
pCallID, 1254
pFailCause, 1254
voiceGetCOLRResp, 1254
pAlphaIDInfo, 1255
pCCResType, 1255
pCCSUPSType, 1255
pCOLRResp, 1255
pCallID, 1255
pFailCause, 1255
voiceGetCallBarringReq, 1244
pSvcClass, 1244
reason, 1244
voiceGetCallBarringResp, 1244
pAlphaIDInfo, 1245
pCCResType, 1246
pCCSUPSType, 1246
pCallID, 1246
pFailCause, 1246
pSvcClass, 1246
voiceGetCallFWReq, 1246
pSvcClass, 1246
Reason, 1246
voiceGetCallFWResp, 1246
pAlphaIDInfo, 1248
pCCResType, 1248
pCCSUPSType, 1248
pCallID, 1248
pFailCause, 1248
pGetCallFWExtInfo, 1248
pGetCallFWInfo, 1248
voiceGetCallWaitInfo, 1248
pAlphaIDInfo, 1249
pCCResType, 1249
pCCSUPSType, 1249
pCallID, 1249
pFailCause, 1249
pSvcClass, 1249
voiceGetConfigReq, 1255
pAMRStatus, 1257
pAirTimer, 1257
pAutoAnswer, 1257
pNamID, 1257
pPrefVoicePrivacy, 1257
pPrefVoiceSO, 1257
pRoamTimer, 1257
pTTYMode, 1257
pVoiceDomainPref, 1257
voiceGetConfigResp, 1257
pAirTimerCnt, 1259
pAutoAnswerStat, 1259
pCurAMRConfig, 1259
pCurPrefVoiceSO, 1259
pCurVoiceDomainPref, 1259
pCurVoicePrivacyPref, 1259
pCurrTTYMode, 1259
pRoamTimerCnt, 1259
voiceIndicationRegisterInfo, 1259
pRegDTMFEvents, 1260
pRegVoicePrivacyEvents, 1260
pSuppsNotifEvents, 1260
voiceInfoRec, 1260
callID, 1262
pCLIRCause, 1262
pCallWaitInd, 1262
pCalledPartyInfo, 1262
pCallerIDInfo, 1262
pCallerNameInfo, 1262
pCallingPartyInfo, 1262
pConnectNumInfo, 1262
pDispInfo, 1262
pExtDispInfo, 1262
pExtDispRecInfo, 1262
pLineCtrlInfo, 1262
pNSSAudioCtrl, 1262
pNSSRelease, 1262
pRedirNumInfo, 1262
pSignalInfo, 1262
voiceManageCallsReq, 1262
pCallID, 1263
SUPSType, 1263
voiceManageCallsResp, 1263
pFailCause, 1264
VoiceNumber
unpack_dms_GetVoiceNumber_t, 1003
voiceNumberSize
unpack_dms_GetVoiceNumber_t, 1003
voiceOTASPStatusInfo, 1264
callID, 1265
OTASPStatus, 1265
voiceOrigUSSDNoWaitInfo, 1264
USSInformation, 1264
voicePrivacy
voicePrivacyInfo, 1266
voicePrivacyInfo, 1265
callID, 1266
voicePrivacy, 1266
voiceSUPSInfo, 1278
pAlphaIDInfo, 1280
pCLIPstatus, 1280

pCLIRstatus, 1280
pCNAPstatus, 1280
pCOLPstatus, 1280
pCOLRstatus, 1280
pCallBarPasswd, 1280
pCallFWNum, 1280
pCallFWTimerVal, 1280
pCallFwdInfo, 1280
pCallID, 1280
pDataSrc, 1280
pFailCause, 1280
pNewPwdData, 1280
pReason, 1280
pSvcClass, 1280
pUSSInfo, 1280
SUPSInformation, 1280
voiceSUPSNotification, 1280
callID, 1282
notifType, 1282
pCUGIndex, 1282
pECTNum, 1282
voiceSetAllCallStatusCbkInfo, 1266
arrCallInfomation, 1268
pArrAlertingPattern, 1268
pArrAlertingType, 1268
pArrAlphaID, 1268
pArrCallEndReason, 1268
pArrCalledPartyNum, 1268
pArrConnectPartyNum, 1268
pArrDiagInfo, 1268
pArrRedirPartyNum, 1268
pArrRemotePartyName, 1268
pArrRemotePartyNum, 1268
pArrSvcOption, 1268
voiceSetCallBarringPwdInfo, 1268
newPasswd, 1269
newPasswdAgain, 1269
oldPasswd, 1269
Reason, 1269
voiceSetCallBarringPwdResp, 1269
pAlphaIDInfo, 1270
pCCResType, 1270
pCCSUPSType, 1270
pCallID, 1270
pFailCause, 1270
voiceSetConfigReq, 1270
pAirTimerConfig, 1271
pAutoAnswer, 1272
pPrefVoiceDomain, 1272
pPrefVoiceSO, 1272
pRoamTimerConfig, 1272
pTTYMode, 1272
voiceSetConfigResp, 1272
pAirTimerStatus, 1273
pAutoAnsStatus, 1273
pPrefVoiceSOStatus, 1273
pRoamTimerStatus, 1273
pTTYConfigStatus, 1273
pVoiceDomainPrefStatus, 1273
voiceSetPrefPrivacy, 1273
privacyPref, 1274
voiceSetSUPSServiceReq, 1274
pCallBarringPasswd, 1276
pCallForwardingNumber, 1276
pCallFwdTypeAndPlan, 1276
pServiceClass, 1276
pTimerVal, 1276
reason, 1276
voiceSvc, 1276
voiceSetSUPSServiceResp, 1276
pAlphaIDInfo, 1277
pCCResultType, 1277
pCCSUPSType, 1277
pCallID, 1277
pFailCause, 1277
voiceStopContDTMFinfo, 1277
callID, 1278
voiceSvc
pack_voice_SLQSVoiceSetSUPSService_t, 677
voiceSetSUPSServiceReq, 1276
VolValue
SetAudioVolTLBConfigReq, 811
Volume
GetAudioProfileResp, 256
GetAudioVolTLBConfigReq, 257
GetM2MAudioProfileResp, 273
SetAudioProfileReq, 810
SetAudioVolTLBConfigReq, 811
voteForInit
registerRefresh, 777
uim_registerRefresh, 952

WAMSParaChanged
_SwiAvmsEventInd, 100
wAlertMsgLength
_QmiSwiAvmsEventReportConfigResp, 74
WCDMACellInfo
lteWcdmaCellInfo, 403
nas_lteWcdmaCellInfo, 465
WCDMAECIOThresh, 1283
pWCDMAECIOThreshList, 1283
WCDMAECIOThreshListLen, 1283
WCDMAECIOThreshListLen
nas_WCDMAECIOThresh, 516
WCDMAECIOThresh, 1283
WCDMAInfoLTENeighborCell, 1283
UMTSLTENbrCell, 1284
umtsLTENbrCellLen, 1284
wcdmaRRCState, 1284
WCDMARSSIThresh, 1288
pWCDMARSSIThreshList, 1289
WCDMARSSIThreshListLen, 1289
WCDMARSSIThreshListLen
nas_WCDMARSSIThresh, 517
WCDMARSSIThresh, 1289
WCDMASSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1073

WCDMASysInfo, 1289
cellId, 1291
cellIdValid, 1292
hsCallStatus, 1292
hsCallStatusValid, 1292
hsInd, 1292
hsIndValid, 1292
lac, 1292
lacValid, 1292
MCC, 1292
MNC, 1292
networkIdValid, 1292
psc, 1292
pscValid, 1292
regRejectInfoValid, 1292
rejCause, 1292
rejectSrvDomain, 1292
sysInfoWCDMA, 1292
WDS_IsGobiDevice
qaGobiApiWds.h, 1860
WDS_SRV
qaGobiApiCbk.h, 1463
WDSGetLoopbackData, 1316
ByteLoopbackMode, 1317
ByteLoopbackMultiplier, 1317
WDSSWICurrentChannelRates, 1323
current_channel_rx_rate, 1324
current_channel_tx_rate, 1324
max_channel_rx_rate, 1324
max_channel_tx_rate, 1324
WDSSetLoopbackData, 1323
pLoopbackMode, 1323
pLoopbackMultiplier, 1323
wDescriptionLength
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
wErrorCode
_QmiSwiAvmsEventReportDataSessionStatus-Resp, 75
wHTTPStatus
_QmiSwiAvmsEventReportHTTPStatusResp, 76
wNameLength
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
WORD
SwiDataTypes.h, 1904
wSessionStatus
_QmiSwiAvmsEventReportNotificationResp, 78
wUpdateCompeteStatus
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
wUserInputTimeout
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
_QmiSwiAvmsEventReportConnectionRequest-Resp, 75
wVersionLength
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
wWamsChangeMask
_QmiSwiAvmsEventReportWAMSParamChange-Resp, 79
wake_mask
pack_rms_SetSMSWake_t, 633
unpack_rms_GetSMSWake_t, 1110
wcdmaAmrStat
curAMRConfig, 195
voice_curAMRConfig, 1219
wcdmaCellInfo, 1282
cpich_ecno, 1282
cpich_rscp, 1282
psc, 1282
srxlev, 1283
wcdmaLongMsgDecodingParams, 1284
Date, 1285
pIsUDHPresent, 1285
pMessage, 1285
pPartNum, 1286
pReferenceNum, 1286
pScAddr, 1286
pScAddrLength, 1286
pSenderAddr, 1286
pSenderAddrLength, 1286
pTextMsg, 1286
pTextMsgLength, 1286
pTotalNum, 1286
Time, 1286
wcdmaMsgDecodingParams, 1286
Date, 1287
pMessage, 1287
pScAddr, 1287
pScAddrLength, 1287
pSenderAddr, 1287
pSenderAddrLength, 1287
pTextMsg, 1287
pTextMsgLength, 1287
Time, 1287
wcdmaMsgEncodingParams, 1287
alphabet, 1288
messageSize, 1288
pDestAddr, 1288
pPDUMessage, 1288
pTextMsg, 1288
wcdmaRRCState
nas_WCDMAInfoLTENeighborCell, 517
WCDMAInfoLTENeighborCell, 1284
wcdmaUARFCN, 1292
status, 1293
uarfcn, 1293
wds.h, 1960
BYT_STAT_STAT_MASK, 1966
PACK_WDS_IPV4, 1966
PACK_WDS_IPV6, 1966
pack_wds_DHCPv4ClientLeaseChange, 1966
pack_wds_GetAutoconnect, 1966

pack_wds_GetByteTotals, 1967
pack_wds_GetConnectionRate, 1967
pack_wds_GetDataBearerTechnology, 1967
pack_wds_GetDefaultProfile, 1968
pack_wds_GetDefaultProfileNum, 1968
pack_wds_GetDormancyState, 1969
pack_wds_GetLastMobileIPError, 1969
pack_wds_GetMobileIP, 1969
pack_wds_GetMobileIPProfile, 1970
pack_wds_GetPacketStatistics, 1970
pack_wds_GetPacketStatus, 1971
pack_wds_GetSessionDuration, 1971
pack_wds_GetSessionState, 1971
pack_wds_RMSetTransferStatistics, 1972
pack_wds_SLQSCreateProfile, 1975
pack_wds_SLQSDeleteProfile, 1975
pack_wds_SLQSGet3GPPConfigItem, 1976
pack_wds_SLQSGetCurrDataSystemStat, 1976
pack_wds_SLQSGetCurrentChannelRate, 1976
pack_wds_SLQSGetDUNCallInfo, 1977
pack_wds_SLQSGetDataBearerTechnology, 1977
pack_wds_SLQSGetProfileSettings, 1978
pack_wds_SLQSGetRuntimeSettings, 1978
pack_wds_SLQSModifyProfile, 1979
pack_wds_SLQSResetPacketStatics, 1979
pack_wds_SLQSSGetDHCPv4ClientConfig, 1981
pack_wds_SLQSSGetLoopback, 1981
pack_wds_SLQSSSetDHCPv4ClientConfig, 1981
pack_wds_SLQSSSetLoopback, 1982
pack_wds_SLQSSet3GPPConfigItem, 1979
pack_wds_SLQSSetIPFamilyPreference, 1980
pack_wds_SLQSSetWdsEventCallback, 1980
pack_wds_SLQSStartDataSession, 1982
pack_wds_SLQSStopDataSession, 1983
pack_wds_SLQSWdsGoActive, 1983
pack_wds_SLQSWdsGoDormant, 1983
pack_wds_SLQSWdsSetEventReport, 1984
pack_wds_SLQSWdsSwiPDPRuntimeSettings, 1984
pack_wds_SetAutoconnect, 1972
pack_wds_SetDefaultProfile, 1973
pack_wds_SetDefaultProfileNum, 1973
pack_wds_SetMobileIP, 1973
pack_wds_SetMobileIPParameters, 1974
pack_wds_SetMobileIPProfile, 1974
unpack_wds_DHCPv4ClientLease_ind, 1985
unpack_wds_DHCPv4ClientLeaseChange, 1985
unpack_wds_GetAutoconnect, 1985
unpack_wds_GetByteTotals, 1986
unpack_wds_GetConnectionRate, 1986
unpack_wds_GetDataBearerTechnology, 1986
unpack_wds_GetDefaultProfile, 1987
unpack_wds_GetDefaultProfileNum, 1987
unpack_wds_GetDormancyState, 1987
unpack_wds_GetLastMobileIPError, 1988
unpack_wds_GetMobileIP, 1988
unpack_wds_GetMobileIPProfile, 1988
unpack_wds_GetPacketStatistics, 1989
unpack_wds_GetPacketStatus, 1989
unpack_wds_GetSessionDuration, 1989
unpack_wds_GetSessionState, 1990
unpack_wds_RMSetTransferStatistics, 1990
unpack_wds_RMTransferStatistics_ind, 1990
unpack_wds_SLQSCreateProfile, 1993
unpack_wds_SLQSDeleteProfile, 1993
unpack_wds_SLQSGet3GPPConfigItem, 1993
unpack_wds_SLQSGetCurrDataSystemStat, 1994
unpack_wds_SLQSGetCurrentChannelRate, 1994
unpack_wds_SLQSGetDUNCallInfo, 1995
unpack_wds_SLQSGetDataBearerTechnology, 1994
unpack_wds_SLQSGetProfileSettings, 1995
unpack_wds_SLQSGetRuntimeSettings, 1995
unpack_wds_SLQSModifyProfile, 1996
unpack_wds_SLQSResetPacketStatics, 1996
unpack_wds_SLQSSGetDHCPv4ClientConfig, 1998
unpack_wds_SLQSSGetLoopback, 1998
unpack_wds_SLQSSSetDHCPv4ClientConfig, 1999
unpack_wds_SLQSSSetLoopback, 1999
unpack_wds_SLQSSet3GPPConfigItem, 1996
unpack_wds_SLQSSetIPFamilyPreference, 1997
unpack_wds_SLQSSetPacketSrvStatusCallback, 1997
unpack_wds_SLQSSetWdsEventCallback, 1997
unpack_wds_SLQSSetWdsEventCallback_ind, 1998
unpack_wds_SLQSStartDataSession, 1999
unpack_wds_SLQSStopDataSession, 2000
unpack_wds_SLQSWdsGoActive, 2000
unpack_wds_SLQSWdsGoDormant, 2000
unpack_wds_SLQSWdsSetEventReport, 2001
unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 2001
unpack_wds_SetAutoconnect, 1991
unpack_wds_SetDefaultProfile, 1991
unpack_wds_SetDefaultProfileNum, 1991
unpack_wds_SetMobileIP, 1992
unpack_wds_SetMobileIPParameters, 1992
unpack_wds_SetMobileIPProfile, 1992
UnpackQmiProfileInfo, 1966
wds_DHCPLeaseOptTlv, 1295
numOpt, 1295
optList, 1295
optListData, 1295
TlvPresent, 1295
wds_DHCPLeaseStateTlv, 1295
leaseState, 1295
TlvPresent, 1295
wds_DHCPOpt, 1295
optCode, 1296
optValLen, 1296
pOptVal, 1296
wds_DHCPProfileIdTlv, 1296
profileId, 1296

profileType, 1296
TlvPresent, 1296
wds_DHCPv4HWConfig, 1296
chaddr, 1296
chaddrLen, 1296
hwType, 1297
wds_DHCPv4Option, 1297
optCode, 1297
optVal, 1297
optValLen, 1297
wds_DHCPv4OptionList, 1297
numOpt, 1298
pOptList, 1298
wds_DHCPv4ProfileId, 1298
profileId, 1298
profileType, 1298
wds_DataULongLongTlv, 1294
TlvPresent, 1294
ullData, 1294
wds_DataULongTlv, 1294
TlvPresent, 1295
ulData, 1295
wds_Domain, 1298
domainLen, 1299
domainName, 1299
wds_DomainNameList, 1299
domain, 1299
numInstances, 1299
wds_GPRSQoS, 1299
delayClass, 1300
meanThroughputClass, 1300
peakThroughputClass, 1300
precedenceClass, 1300
reliabilityClass, 1300
wds_IPV6AddressInfo, 1300
IPAddressV6, 1301
IPV6PrefixLen, 1301
wds_IPV6GWAddressInfo, 1301
gwAddressV6, 1301
gwV6PrefixLen, 1301
wds_IPv4AdTlv, 1300
IPv4Addr, 1300
TlvPresent, 1300
wds_PCSCFFQDNAddress, 1301
fqdnAddr, 1302
fqdnLen, 1302
wds_PCSCFFQDNAddressList, 1302
numInstances, 1302
pcsfFQDNAddress, 1302
wds_PCSCFIPv4ServerAddressList, 1302
numInstances, 1303
pscsfIPv4Addr, 1303
wds_ProfileIdentifier, 1303
profileIndex, 1303
profileType, 1303
wds_TrStatInd, 1304
statsMask, 1304
statsPeriod, 1304
wds_UMTSMinQoS, 1304
deliveryErrSDU, 1306
grntDownlinkBitrate, 1306
grntUplinkBitrate, 1307
maxDownlinkBitrate, 1307
maxSDUSize, 1307
maxUplinkBitrate, 1307
qosDeliveryOrder, 1307
resBerRatio, 1307
sduErrorRatio, 1307
trafficClass, 1307
trafficPriority, 1307
transferDelay, 1307
wds_currNetworkInfo, 1293
NetworkType, 1294
RATMask, 1294
SOMask, 1294
wds_profileInfo, 1303
SlqsProfile3GPP, 1304
SlqsProfile3GPP2, 1304
WdsByteTotals, 1307
ByteTotalsElmntsV4, 1308
ByteTotalsElmntsV6, 1308
pV4sessionId, 1308
pV6sessionId, 1308
WdsByteTotalsElmnts, 1308
pRXTotalBytes, 1308
pTXTotalBytes, 1308
WdsClientLeaseChange, 1308
pEnableNotification, 1309
WdsConnectionRate, 1309
ConnRateElmntsV4, 1309
ConnRateElmntsV6, 1309
pV4sessionId, 1309
pV6sessionId, 1309
WdsConnectionRateElmnts, 1310
pCurrentChannelRXRate, 1310
pCurrentChannelTXRate, 1310
pMaxChannelRXRate, 1310
pMaxChannelTXRate, 1310
WdsDHCPv4ClientLeaseInd, 1310
pIPv4Addr, 1311
pLeaseState, 1311
pOptList, 1311
pProfileId, 1311
WdsDHCPv4Config, 1311
pHwConfig, 1312
pProfileId, 1312
pRequestOptionList, 1312
WdsDHCPv4HWConfig, 1312
chaddr, 1313
chaddrLen, 1313
hwType, 1313
WdsDHCPv4Option, 1314
optCode, 1314
optVal, 1314
optValLen, 1314
WdsDHCPv4OptionList, 1314

numOpt, 1315
pOptList, 1315
WdsDHCPv4ProfileId, 1315
profileId, 1315
profileType, 1315
wdsDhcpv4HwConfig, 1312
chaddr, 1312
chaddrLen, 1312
hwType, 1312
wdsDhcpv4Option, 1313
optCode, 1313
optVal, 1313
optValLen, 1313
wdsDhcpv4OptionList, 1315
numOpt, 1315
pOptList, 1315
wdsDhcpv4ProfileId, 1316
profileId, 1316
profileType, 1316
WdsIpAddressInfoReq, 1317
ip, 1317
pv4sessionId, 1317
pv6sessionId, 1317
WdsPktStatisticsElmnts, 1317
pRXDroppedCount, 1318
pRXOKBytesLastCall, 1318
pRXOkBytesCount, 1318
pRXPacketErrors, 1318
pRXPacketOverflows, 1319
pRXPacketSuccesses, 1319
pTXDroppedCount, 1319
pTXOKBytesLastCall, 1319
pTXOkBytesCount, 1319
pTXPacketErrors, 1319
pTXPacketOverflows, 1319
pTXPacketSuccesses, 1319
WdsPktStatisticsReq, 1319
pStatMask, 1319
WdsPktStatisticsResp, 1319
pV4sessionId, 1320
pV6sessionId, 1320
PktStatElmntsV4, 1320
PktStatElmntsV6, 1320
WdsProfileParam, 1320
SlqsProfile3GPP, 1320
SlqsProfile3GPP2, 1320
WdsRunTimeSettings, 1320
rts, 1321
v4sessionId, 1321
v6sessionId, 1321
wdsSetEventReportReq, 1321
pCurrChannelRateInd, 1323
pCurrDataBearerTechInd, 1323
pCurrPrefDataSysInd, 1323
pDataBearerTechInd, 1323
pDataCallStatusChangeInd, 1323
pDataSystemStatusChangeInd, 1323
pDormancyStatusInd, 1323
pEVDOPageMonPerChangeInd, 1323
pMIPStatusInd, 1323
pTransferStatInd, 1323
Wireless Data Service (WDS), 38

xAxis
sensorData, 798
sensorData_t, 799
xferStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1183
xid
pack_loc_SLQSLOCGetBestAvailPos_t, 595
pack_qmi_t, 631
unpack_qmi_t, 1090
xtra_start_gps_minutes
GPSStateInfo, 288
xtra_start_gps_week
GPSStateInfo, 288
xtra_valid_duration_hours
GPSStateInfo, 288

yAxis
sensorData, 798
sensorData_t, 799
year
nas_timeInfo, 508
nas_UniversalTime, 514
timeInfo, 922
UniversalTime, 991

zAxis
sensorData, 798
sensorData_t, 799