LinuxQMISDK

SLQS04.00.07

Generated by Doxygen 1.8.6

Wed May 24 2017 06:41:02

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 . . . 174
8.114.2.9 cdmaSysIdValid . . . 174

8.114.2.10 isSysPrlMatch . . . 174
8.114.2.11 isSysPrlMatchValid . . . 174
8.114.2.12 MCC . . . 174
8.114.2.13 MNC . . . 174
8.114.2.14 networkID . . . 174
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 . . . 175
8.116 CellDb Struct Reference . . . 175
8.116.1 Detailed Description . . . 175
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 . . . 178
8.120 ChannelRate Struct Reference . . . 178
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 . . . 180
8.122.2.1 ActiveStatus . . . 180
8.122.2.2 ProvisionStatus . . . 180
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 . . . 182
8.124.2 Field Documentation . . . 182
8.124.2.1 ActiveStatus . . . 182
8.124.2.2 ProvisionStatus . . . 182
8.125 COLPResp Struct Reference . . . 182
8.125.1 Detailed Description . . . 182
8.125.2 Field Documentation . . . 183
8.125.2.1 ActiveStatus . . . 183
8.125.2.2 ProvisionStatus . . . 183
8.126 COLRResp Struct Reference . . . 183
8.126.1 Detailed Description . . . 183
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 . . . 184

8.127.2 Field Documentation . . . 185
8.127.2.1 imsRegState . . . 185
8.127.2.2 modemMode . . . 185
8.127.2.3 psState . . . 185
8.127.2.4 systemMode . . . 185
8.127.2.5 temperature . . . 185
8.128 ConnectionStatus Struct Reference . . . 185
8.128.1 Detailed Description . . . 185
8.128.2 Field Documentation . . . 186
8.128.2.1 MDMCallDuration . . . 186
8.128.2.2 MDMConnStatus . . . 186
8.129 connectionStatus Struct Reference . . . 186
8.129.1 Detailed Description . . . 186
8.129.2 Field Documentation . . . 186
8.129.2.1 MDMCallDuration . . . 186
8.129.2.2 MDMConnStatus . . . 187
8.130 connectNumInfo Struct Reference . . . 187
8.130.1 Detailed Description . . . 187
8.130.2 Field Documentation . . . 188
8.130.2.1 callerID . . . 188
8.130.2.2 callerIDLen . . . 188
8.130.2.3 numPlan . . . 188
8.130.2.4 numPresInd . . . 188
8.130.2.5 numType . . . 188
8.130.2.6 screeningInd . . . 188
8.131 CrashInfo Struct Reference . . . 188
8.131.1 Detailed Description . . . 188
8.131.2 Field Documentation . . . 189
8.131.2.1 crashData . . . 189
8.131.2.2 crashId . . . 189
8.131.2.3 crashStrLen . . . 189
8.131.2.4 gcDumpStrLen . . . 189
8.131.2.5 numCrashes . . . 189
8.131.2.6 pCrashString . . . 189
8.131.2.7 pGCDumpString . . . 189
8.132 crashInfoParams Struct Reference . . . 189
8.132.1 Detailed Description . . . 189
8.132.2 Field Documentation . . . 190
8.132.2.1 crashInfo . . . 190
8.132.2.2 crashStatus . . . 190

8.133 CrashInfoParams Struct Reference . . . 190
8.133.1 Detailed Description . . . 190
8.133.2 Field Documentation . . . 190
8.133.2.1 pCrashInfo . . . 190
8.133.2.2 pDevCrashStatus . . . 190
8.134 crashInformation Struct Reference . . . 190
8.134.1 Detailed Description . . . 191
8.134.2 Field Documentation . . . 191
8.134.2.1 crashData . . . 191
8.134.2.2 crashId . . . 191
8.134.2.3 crashString . . . 191
8.134.2.4 crashStrlen . . . 191
8.134.2.5 gcdumpString . . . 191
8.134.2.6 gcdumpStrlen . . . 191
8.134.2.7 numCrashes . . . 191
8.135 CreateProfileIn Struct Reference . . . 191
8.135.1 Detailed Description . . . 192
8.135.2 Field Documentation . . . 192
8.135.2.1 curProfile . . . 192
8.135.2.2 pProfileID . . . 192
8.135.2.3 pProfileType . . . 192
8.136 CreateProfileOut Struct Reference . . . 192
8.136.1 Detailed Description . . . 192
8.136.2 Field Documentation . . . 193
8.136.2.1 pExtErrorCode . . . 193
8.136.2.2 pProfileIndex . . . 193
8.136.2.3 pProfileType . . . 193
8.137 CsgId Struct Reference . . . 193
8.137.1 Detailed Description . . . 193
8.137.2 Field Documentation . . . 193
8.137.2.1 csgId . . . 193
8.137.2.2 TlvPresent . . . 193
8.138 CSGID Struct Reference . . . 193
8.138.1 Detailed Description . . . 194
8.138.2 Field Documentation . . . 194
8.138.2.1 id . . . 194
8.138.2.2 mcc . . . 194
8.138.2.3 mnc . . . 194
8.138.2.4 mncPcsDigits . . . 194
8.138.2.5 rat . . . 194

8.139 CUGInfo Struct Reference . . . 194
8.139.1 Detailed Description . . . 195
8.139.2 Field Documentation . . . 195
8.139.2.1 CUGIndex . . . 195
8.139.2.2 SuppOA . . . 195
8.139.2.3 SuppPrefCUG . . . 195
8.140 curAMRConfig Struct Reference . . . 195
8.140.1 Detailed Description . . . 195
8.140.2 Field Documentation . . . 196
8.140.2.1 gsmAmrStat . . . 196
8.140.2.2 wcdmaAmrStat . . . 196
8.141 CurrDataSysStat Struct Reference . . . 196
8.141.1 Detailed Description . . . 196
8.141.2 Field Documentation . . . 197
8.141.2.1 pCurrNetworkInfo . . . 197
8.141.2.2 pNetworkInfoLen . . . 197
8.141.2.3 pPrefNetwork . . . 197
8.142 currentCatEvent Union Reference . . . 197
8.142.1 Detailed Description . . . 197
8.142.2 Field Documentation . . . 197
8.142.2.1 CatAlphaIdtfr . . . 197
8.142.2.2 CatEndPS . . . 197
8.142.2.3 CatEventLst . . . 197
8.142.2.4 CatEvIDData . . . 197
8.142.2.5 CatRefresh . . . 197
8.143 CurrentImgList Struct Reference . . . 197
8.143.1 Detailed Description . . . 198
8.143.2 Field Documentation . . . 198
8.143.2.1 carrier . . . 198
8.143.2.2 fwvers . . . 198
8.143.2.3 numEntries . . . 198
8.143.2.4 pCurrImgInfo . . . 198
8.143.2.5 pkgver . . . 198
8.143.2.6 priver . . . 198
8.144 currentPLMN Struct Reference . . . 198
8.144.1 Detailed Description . . . 199
8.144.2 Field Documentation . . . 199
8.144.2.1 MCC . . . 199
8.144.2.2 MNC . . . 199
8.144.2.3 netDescr . . . 199

8.144.2.4 netDescrLength . . . 199
8.145 CurrImageInfo Struct Reference . . . 199
8.145.1 Detailed Description . . . 200
8.145.2 Field Documentation . . . 200
8.145.2.1 buildID . . . 200
8.145.2.2 buildIDLen . . . 200
8.145.2.3 imageType . . . 200
8.145.2.4 uniqueID . . . 200
8.146 CurrNetworkInfo Struct Reference . . . 200
8.146.1 Detailed Description . . . 200
8.146.2 Field Documentation . . . 201
8.146.2.1 NetworkType . . . 202
8.146.2.2 RATMask . . . 202
8.146.2.3 SOMask . . . 202
8.147 currNetworkInfo Struct Reference . . . 202
8.147.1 Detailed Description . . . 202
8.147.2 Field Documentation . . . 203
8.147.2.1 NetworkType . . . 203
8.147.2.2 RATMask . . . 203
8.147.2.3 SOMask . . . 203
8.148 custFeaturesInfo Struct Reference . . . 203
8.148.1 Detailed Description . . . 203
8.148.2 Field Documentation . . . 205
8.148.2.1 GpsEnable . . . 205
8.148.2.2 pDHCPRelayEnabled . . . 205
8.148.2.3 pDisableIMSI . . . 205
8.148.2.4 pGPSLPM . . . 205
8.148.2.5 pGPSSel . . . 205
8.148.2.6 pIPFamSupport . . . 205
8.148.2.7 pIsVoiceEnabled . . . 205
8.148.2.8 pRMAutoConnect . . . 205
8.148.2.9 pSMSSupport . . . 206
8.149 custFeaturesSetting Struct Reference . . . 206
8.149.1 Detailed Description . . . 206
8.149.2 Field Documentation . . . 207
8.149.2.1 pDHCPRelayEnabled . . . 207
8.149.2.2 pGPSEnable . . . 207
8.149.2.3 pGPSLPM . . . 207
8.149.2.4 pGPSSel . . . 207
8.149.2.5 pIsVoiceEnabled . . . 207

8.150 custSettingInfo Struct Reference . . . 207
8.150.1 Detailed Description . . . 207
8.150.2 Field Documentation . . . 208
8.150.2.1 cust_attr . . . 208
8.150.2.2 cust_id . . . 208
8.150.2.3 cust_value . . . 208
8.150.2.4 id_length . . . 208
8.150.2.5 value_length . . . 208
8.151 custSettingList Struct Reference . . . 208
8.151.1 Detailed Description . . . 208
8.151.2 Field Documentation . . . 208
8.151.2.1 custSetting . . . 208
8.151.2.2 list_type . . . 209
8.151.2.3 num_instances . . . 209
8.152 dataBearers Struct Reference . . . 209
8.152.1 Detailed Description . . . 209
8.152.2 Field Documentation . . . 209
8.152.2.1 dataBearerMask . . . 209
8.152.2.2 pCurDataBearerTechnology . . . 209
8.152.2.3 pLastCallDataBearerTechnology . . . 209
8.153 DataBearerTech Struct Reference . . . 209
8.153.1 Detailed Description . . . 209
8.153.2 Field Documentation . . . 211
8.153.2.1 ratValue . . . 211
8.153.2.2 soMask . . . 211
8.153.2.3 techType . . . 211
8.154 DataBearerTechExt Struct Reference . . . 211
8.154.1 Detailed Description . . . 212
8.154.2 Field Documentation . . . 212
8.154.2.1 pBearerTech . . . 212
8.154.2.2 pLastBearerTech . . . 212
8.155 dataBearerTechnology Struct Reference . . . 212
8.155.1 Detailed Description . . . 212
8.155.2 Field Documentation . . . 213
8.155.2.1 currentNetwork . . . 213
8.155.2.2 ratMask . . . 213
8.155.2.3 soMask . . . 213
8.156 dataRate Struct Reference . . . 213
8.156.1 Detailed Description . . . 213
8.156.2 Field Documentation . . . 214

8.156.2.1 dataRateMax . . . 214
8.156.2.2 guaranteedRate . . . 214
8.157 dataSrvCapabilities Struct Reference . . . 214
8.157.1 Detailed Description . . . 214
8.157.2 Field Documentation . . . 214
8.157.2.1 dataCapabilities . . . 214
8.157.2.2 dataCapabilitiesLen . . . 215
8.158 DataStatusDetail Struct Reference . . . 215
8.158.1 Detailed Description . . . 215
8.158.2 Field Documentation . . . 216
8.158.2.1 IPAddress . . . 216
8.158.2.2 LastErrCode . . . 216
8.159 DataULongLongTlv Struct Reference . . . 217
8.159.1 Field Documentation . . . 217
8.159.1.1 TlvPresent . . . 217
8.159.1.2 ullData . . . 217
8.160 DataULongTlv Struct Reference . . . 217
8.160.1 Field Documentation . . . 217
8.160.1.1 TlvPresent . . . 217
8.160.1.2 ulData . . . 217
8.161 DcsUsbPortNames Struct Reference . . . 217
8.161.1 Field Documentation . . . 217
8.161.1.1 AtCmdPort . . . 217
8.161.1.2 DmPort . . . 217
8.161.1.3 NmeaPort . . . 217
8.162 delAssistDataStatus Struct Reference . . . 217
8.162.1 Detailed Description . . . 218
8.162.2 Field Documentation . . . 218
8.162.2.1 status . . . 218
8.163 depersonalizationInformation Struct Reference . . . 218
8.163.1 Detailed Description . . . 218
8.163.2 Field Documentation . . . 219
8.163.2.1 ckLen . . . 219
8.163.2.2 ckVal . . . 219
8.163.2.3 feature . . . 219
8.163.2.4 operation . . . 219
8.164 detailSvcInfo Struct Reference . . . 219
8.164.1 Detailed Description . . . 219
8.164.2 Field Documentation . . . 220
8.164.2.1 hdrHybrid . . . 220

8.164.2.2 hdrSrvStatus . . . 220
8.164.2.3 isSysForbidden . . . 220
8.164.2.4 srvCapability . . . 221
8.164.2.5 srvStatus . . . 221
8.165 DeviceConfigDetail Struct Reference . . . 221
8.165.1 Detailed Description . . . 221
8.165.2 Field Documentation . . . 221
8.165.2.1 Chipset . . . 221
8.165.2.2 HWVersion . . . 222
8.165.2.3 QLIC . . . 222
8.165.2.4 Technology . . . 222
8.166 DHCPOption Struct Reference . . . 222
8.166.1 Detailed Description . . . 222
8.166.2 Field Documentation . . . 222
8.166.2.1 optCode . . . 222
8.166.2.2 optValLen . . . 222
8.166.2.3 pOptVal . . . 222
8.167 DHCPOptionList Struct Reference . . . 222
8.167.1 Detailed Description . . . 222
8.167.2 Field Documentation . . . 223
8.167.2.1 numOpt . . . 223
8.167.2.2 pOptions . . . 223
8.168 diagInfo Struct Reference . . . 223
8.168.1 Detailed Description . . . 223
8.168.2 Field Documentation . . . 223
8.168.2.1 diagInfoLen . . . 223
8.168.2.2 diagnosticInfo . . . 223
8.169 dirNum Struct Reference . . . 223
8.169.1 Detailed Description . . . 224
8.169.2 Field Documentation . . . 224
8.169.2.1 dirNum . . . 224
8.169.2.2 dirNumLen . . . 224
8.170 dms_ActivationStatusTlv Struct Reference . . . 224
8.170.1 Detailed Description . . . 224
8.170.2 Field Documentation . . . 225
8.170.2.1 activationStatus . . . 225
8.170.2.2 TlvPresent . . . 225
8.171 dms_OperatingModeTlv Struct Reference . . . 225
8.171.1 Detailed Description . . . 225
8.171.2 Field Documentation . . . 225

8.171.2.1 operatingMode . . . 225
8.171.2.2 TlvPresent . . . 225
8.172 dmsCurrentPRLInfo Struct Reference . . . 225
8.172.1 Detailed Description . . . 225
8.172.2 Field Documentation . . . 226
8.172.2.1 pPRLPreference . . . 226
8.172.2.2 pPRLVersion . . . 226
8.173 DMScustSettingInfo Struct Reference . . . 226
8.173.1 Detailed Description . . . 226
8.173.2 Field Documentation . . . 227
8.173.2.1 cust_attr . . . 227
8.173.2.2 cust_id . . . 227
8.173.2.3 cust_value . . . 227
8.173.2.4 id_length . . . 227
8.173.2.5 value_length . . . 227
8.174 DMScustSettingList Struct Reference . . . 227
8.174.1 Detailed Description . . . 227
8.174.2 Field Documentation . . . 227
8.174.2.1 custSetting . . . 227
8.174.2.2 list_type . . . 227
8.174.2.3 num_instances . . . 227
8.175 DMSgetCustomFeatureV2 Struct Reference . . . 227
8.175.1 Detailed Description . . . 228
8.175.2 Field Documentation . . . 228
8.175.2.1 pCustSettingInfo . . . 228
8.175.2.2 pCustSettingList . . . 228
8.175.2.3 pGetCustomInput . . . 228
8.176 DMSgetCustomInput Struct Reference . . . 228
8.176.1 Detailed Description . . . 228
8.176.2 Field Documentation . . . 228
8.176.2.1 cust_id . . . 228
8.176.2.2 list_type . . . 228
8.177 dmsIndicationRegisterReq Struct Reference . . . 229
8.177.1 Detailed Description . . . 229
8.177.2 Field Documentation . . . 229
8.177.2.1 pSwiGetResetInd . . . 229
8.178 dmsSwiGetResetInfo Struct Reference . . . 229
8.178.1 Detailed Description . . . 229
8.178.2 Field Documentation . . . 230
8.178.2.1 source . . . 230

8.178.2.2 type . . . 230
8.179 Domain Struct Reference . . . 230
8.179.1 Detailed Description . . . 230
8.179.2 Field Documentation . . . 230
8.179.2.1 domainLen . . . 230
8.179.2.2 domainName . . . 230
8.180 DomainNameList Struct Reference . . . 230
8.180.1 Detailed Description . . . 231
8.180.2 Field Documentation . . . 231
8.180.2.1 domain . . . 231
8.180.2.2 numInstances . . . 231
8.181 DRCParams Struct Reference . . . 231
8.181.1 Detailed Description . . . 231
8.181.2 Field Documentation . . . 231
8.181.2.1 DRCCover . . . 231
8.181.2.2 DRCValue . . . 231
8.182 DTMFInfo Struct Reference . . . 231
8.182.1 Detailed Description . . . 232
8.182.2 Field Documentation . . . 232
8.182.2.1 callID . . . 232
8.182.2.2 digitBuff . . . 232
8.182.2.3 digitCnt . . . 232
8.182.2.4 DTMFEvent . . . 232
8.183 DTMFLengths Struct Reference . . . 232
8.183.1 Detailed Description . . . 232
8.183.2 Field Documentation . . . 233
8.183.2.1 DTMFInterdigitInterval . . . 233
8.183.2.2 DTMFPulseWidth . . . 233
8.184 DUNCallInfoInd Struct Reference . . . 233
8.184.1 Field Documentation . . . 233
8.184.1.1 CallEndReason . . . 233
8.184.1.2 ChannelRate . . . 233
8.184.1.3 DataBearerTech . . . 233
8.184.1.4 DormancyStatus . . . 233
8.184.1.5 MdmConnStatus . . . 233
8.184.1.6 RXOKBytesCount . . . 233
8.184.1.7 TXOKBytesCount . . . 233
8.185 dunchannelRate Struct Reference . . . 234
8.185.1 Detailed Description . . . 234
8.185.2 Field Documentation . . . 234

8.185.2.1 CurrChanRxRate . . . 234
8.185.2.2 CurrChanTxRate . . . 234
8.185.2.3 MaxChanRxRate . . . 234
8.185.2.4 MaxChanTxRate . . . 234
8.186 ecioListElement Struct Reference . . . 234
8.186.1 Detailed Description . . . 234
8.186.2 Field Documentation . . . 235
8.186.2.1 ecio . . . 235
8.186.2.2 radioIf . . . 235
8.187 ECIOThresh Struct Reference . . . 235
8.187.1 Detailed Description . . . 235
8.187.2 Field Documentation . . . 235
8.187.2.1 ECIOThresListLen . . . 235
8.187.2.2 pECIOThresList . . . 235
8.188 ECTNum Struct Reference . . . 236
8.188.1 Detailed Description . . . 236
8.188.2 Field Documentation . . . 236
8.188.2.1 ECTCallState . . . 236
8.188.2.2 number . . . 236
8.188.2.3 presentationInd . . . 236
8.189 encryptedPIN1 Struct Reference . . . 236
8.189.1 Detailed Description . . . 236
8.189.2 Field Documentation . . . 237
8.189.2.1 pin1Len . . . 237
8.189.2.2 pin1Val . . . 237
8.190 eriDataparams Struct Reference . . . 237
8.190.1 Detailed Description . . . 237
8.190.2 Field Documentation . . . 237
8.190.2.1 eriData . . . 237
8.190.2.2 eriDataLen . . . 237
8.191 ERIFileparams Struct Reference . . . 237
8.191.1 Detailed Description . . . 238
8.191.2 Field Documentation . . . 238
8.191.2.1 pFile . . . 238
8.191.2.2 pFileSize . . . 238
8.192 errorRateListElement Struct Reference . . . 238
8.192.1 Detailed Description . . . 238
8.192.2 Field Documentation . . . 239
8.192.2.1 errorRate . . . 239
8.192.2.2 radioIf . . . 239

8.193 eTWSPLMNInfoTlv Struct Reference . . . 239
8.193.1 Detailed Description . . . 239
8.193.2 Field Documentation . . . 239
8.193.2.1 ETWSPLMNInfo . . . 239
8.193.2.2 TlvPresent . . . 239
8.194 extDispRecInfo Struct Reference . . . 239
8.194.1 Detailed Description . . . 240
8.194.2 Field Documentation . . . 240
8.194.2.1 dispType . . . 240
8.194.2.2 extDispInfo . . . 240
8.194.2.3 extDispInfoLen . . . 240
8.195 FactorySequenceNumber Struct Reference . . . 240
8.195.1 Detailed Description . . . 240
8.195.2 Field Documentation . . . 240
8.195.2.1 FSNumber . . . 240
8.196 fileAttributes Struct Reference . . . 240
8.196.1 Detailed Description . . . 241
8.196.2 Field Documentation . . . 243
8.196.2.1 fileID . . . 243
8.196.2.2 fileSize . . . 243
8.196.2.3 fileType . . . 243
8.196.2.4 rawLen . . . 243
8.196.2.5 rawValue . . . 243
8.196.2.6 recordCount . . . 243
8.196.2.7 recordSize . . . 243
8.196.2.8 secActivate . . . 243
8.196.2.9 secActivateMask . . . 243
8.196.2.10 secDeactivate . . . 243
8.196.2.11 secDeactivateMask . . . 244
8.196.2.12 secIncrease . . . 244
8.196.2.13 secIncreaseMask . . . 244
8.196.2.14 secRead . . . 244
8.196.2.15 secReadMask . . . 244
8.196.2.16 secWrite . . . 244
8.196.2.17 secWriteMask . . . 244
8.197 fileInfo Struct Reference . . . 244
8.197.1 Detailed Description . . . 244
8.197.2 Field Documentation . . . 244
8.197.2.1 fileID . . . 244
8.197.2.2 path . . . 244

8.197.2.3 pathLen . . . 244
8.198 FirmwareUpdatStat Struct Reference . . . 244
8.198.1 Detailed Description . . . 245
8.198.2 Field Documentation . . . 246
8.198.2.1 pImgType . . . 246
8.198.2.2 pLogString . . . 246
8.198.2.3 pLogStringLen . . . 246
8.198.2.4 pRefData . . . 246
8.198.2.5 pRefString . . . 246
8.198.2.6 pRefStringLen . . . 246
8.198.2.7 ResCode . . . 246
8.199 FMSImageElement Struct Reference . . . 246
8.199.1 Detailed Description . . . 246
8.199.2 Field Documentation . . . 246
8.199.2.1 buildId . . . 247
8.199.2.2 buildIdLength . . . 247
8.199.2.3 imageId . . . 247
8.199.2.4 imageType . . . 247
8.200 FMSImageIdElement Struct Reference . . . 247
8.200.1 Detailed Description . . . 247
8.200.2 Field Documentation . . . 247
8.200.2.1 buildID . . . 247
8.200.2.2 buildIDLength . . . 247
8.200.2.3 failureCount . . . 247
8.200.2.4 imageID . . . 247
8.200.2.5 storageIndex . . . 247
8.201 FMSImageIDEntries Struct Reference . . . 248
8.201.1 Detailed Description . . . 248
8.201.2 Field Documentation . . . 248
8.201.2.1 executingImage . . . 248
8.201.2.2 imageIDElement . . . 248
8.201.2.3 imageIDSize . . . 248
8.201.2.4 imageType . . . 248
8.201.2.5 maxImages . . . 248
8.202 FMSImageList Struct Reference . . . 248
8.202.1 Detailed Description . . . 249
8.202.2 Field Documentation . . . 249
8.202.2.1 imageIDEntries . . . 249
8.202.2.2 listSize . . . 249
8.203 FMSPrefImageList Struct Reference . . . 249

8.203.1 Detailed Description . . . 249
8.203.2 Field Documentation . . . 249
8.203.2.1 listEntries . . . 249
8.203.2.2 listSize . . . 249
8.204 fwinfo_s Struct Reference . . . 249
8.204.1 Detailed Description . . . 250
8.204.2 Field Documentation . . . 250
8.204.2.1 Carrier . . . 250
8.204.2.2 FirmwareID . . . 250
8.204.2.3 GPSCapability . . . 250
8.204.2.4 Region . . . 250
8.204.2.5 Technology . . . 250
8.205 GERANInfo Struct Reference . . . 250
8.205.1 Detailed Description . . . 250
8.205.2 Field Documentation . . . 251
8.205.2.1 arfcn . . . 251
8.205.2.2 bsic . . . 252
8.205.2.3 cellID . . . 252
8.205.2.4 insNmrCellInfo . . . 252
8.205.2.5 lac . . . 252
8.205.2.6 nmrInst . . . 252
8.205.2.7 plmn . . . 252
8.205.2.8 rxLev . . . 252
8.205.2.9 timingAdvance . . . 252
8.206 geranInstInfo Struct Reference . . . 252
8.206.1 Detailed Description . . . 252
8.206.2 Field Documentation . . . 252
8.206.2.1 geranArfcn . . . 252
8.206.2.2 geranBsicBcc . . . 252
8.206.2.3 geranBsicNcc . . . 252
8.206.2.4 geranRssi . . . 253
8.207 getAllCallInformation Struct Reference . . . 253
8.207.1 Detailed Description . . . 253
8.207.2 Field Documentation . . . 253
8.207.2.1 ALS . . . 253
8.207.2.2 Callinfo . . . 253
8.207.2.3 isEmpty . . . 253
8.208 getAllCallRmtPtyName Struct Reference . . . 253
8.208.1 Detailed Description . . . 253
8.208.2 Field Documentation . . . 254

8.208.2.1 callID . . . 254
8.208.2.2 RemotePartyName . . . 254
8.209 getAllCallRmtPtyNum Struct Reference . . . 254
8.209.1 Detailed Description . . . 254
8.209.2 Field Documentation . . . 254
8.209.2.1 callID . . . 254
8.209.2.2 RemotePartyNum . . . 254
8.210 GetAudioPathConfigReq Struct Reference . . . 254
8.210.1 Detailed Description . . . 254
8.210.2 Field Documentation . . . 255
8.210.2.1 Item . . . 255
8.210.2.2 Profile . . . 255
8.211 GetAudioPathConfigResp Struct Reference . . . 255
8.211.1 Detailed Description . . . 255
8.211.2 Field Documentation . . . 256
8.211.2.1 pCodecSTGain . . . 257
8.211.2.2 pDTMFTXGain . . . 257
8.211.2.3 pECMode . . . 257
8.211.2.4 pMICGainSelect . . . 257
8.211.2.5 pNSEnable . . . 257
8.211.2.6 pRXAGCList . . . 257
8.211.2.7 pRXAVCAGCSwitch . . . 257
8.211.2.8 pRXAVCList . . . 257
8.211.2.9 pRXPCMIIRFltr . . . 257
8.211.2.10 pTXAGCList . . . 257
8.211.2.11 pTXAVCSwitch . . . 257
8.211.2.12 pTXGain . . . 257
8.211.2.13 pTXPCMIIRFltr . . . 257
8.212 GetAudioProfileReq Struct Reference . . . 257
8.212.1 Detailed Description . . . 257
8.212.2 Field Documentation . . . 257
8.212.2.1 Generator . . . 257
8.213 GetAudioProfileResp Struct Reference . . . 257
8.213.1 Detailed Description . . . 258
8.213.2 Field Documentation . . . 258
8.213.2.1 EarMute . . . 258
8.213.2.2 MicMute . . . 258
8.213.2.3 Profile . . . 258
8.213.2.4 Volume . . . 258
8.214 GetAudioVolTLBConfigReq Struct Reference . . . 259

8.214.1 Detailed Description . . . 259
8.214.2 Field Documentation . . . 259
8.214.2.1 Generator . . . 259
8.214.2.2 Item . . . 259
8.214.2.3 Profile . . . 259
8.214.2.4 Volume . . . 259
8.215 GetAudioVolTLBConfigResp Struct Reference . . . 259
8.215.1 Detailed Description . . . 260
8.215.2 Field Documentation . . . 260
8.215.2.1 ResCode . . . 260
8.216 getCallFWExtInfo Struct Reference . . . 260
8.216.1 Detailed Description . . . 260
8.216.2 Field Documentation . . . 260
8.216.2.1 CallFWExtInfo . . . 260
8.216.2.2 numInstances . . . 260
8.217 getCallFWInfo Struct Reference . . . 260
8.217.1 Detailed Description . . . 260
8.217.2 Field Documentation . . . 261
8.217.2.1 CallFWInfo . . . 261
8.217.2.2 numInstances . . . 261
8.218 getCustomFeatureV2 Struct Reference . . . 261
8.218.1 Detailed Description . . . 261
8.218.2 Field Documentation . . . 261
8.218.2.1 pCustSettingInfo . . . 261
8.218.2.2 pCustSettingList . . . 261
8.218.2.3 pGetCustomInput . . . 261
8.219 getCustomInput Struct Reference . . . 262
8.219.1 Detailed Description . . . 262
8.219.2 Field Documentation . . . 262
8.219.2.1 cust_id . . . 262
8.219.2.2 list_type . . . 262
8.220 getDUNCallInfoReq Struct Reference . . . 262
8.220.1 Detailed Description . . . 262
8.220.2 Field Documentation . . . 263
8.220.2.1 Mask . . . 263
8.220.2.2 pReportChannelRate . . . 263
8.220.2.3 pReportConnStatus . . . 263
8.220.2.4 pReportDataBearerTech . . . 263
8.220.2.5 pReportDormStatus . . . 263
8.220.2.6 pTransferStatInd . . . 263

8.221 getDUNCallInfoResp Struct Reference . . . 264
8.221.1 Detailed Description . . . 264
8.221.2 Field Documentation . . . 266
8.221.2.1 pCallEndReason . . . 266
8.221.2.2 pChannelRate . . . 266
8.221.2.3 pConnectionStatus . . . 266
8.221.2.4 pDataBearerTech . . . 266
8.221.2.5 pDormancyStatus . . . 266
8.221.2.6 pLastCallDataBearerTech . . . 266
8.221.2.7 pLastCallRXOKBytesCnt . . . 266
8.221.2.8 pLastCallTXOKBytesCnt . . . 266
8.221.2.9 pMdmCallDurationActive . . . 266
8.221.2.10 pRXOKBytesCount . . . 266
8.221.2.11 pTXOKBytesCount . . . 266
8.222 getDyingGaspCfg Struct Reference . . . 266
8.222.1 Detailed Description . . . 267
8.222.2 Field Documentation . . . 267
8.222.2.1 pDestSMSContent . . . 267
8.222.2.2 pDestSMSNum . . . 267
8.223 getDyingGaspStatistics Struct Reference . . . 267
8.223.1 Detailed Description . . . 267
8.223.2 Field Documentation . . . 267
8.223.2.1 pSMSAttemptedFlag . . . 268
8.223.2.2 pTimeStamp . . . 268
8.224 GetErrRateResp Struct Reference . . . 268
8.224.1 Detailed Description . . . 268
8.224.2 Field Documentation . . . 268
8.224.2.1 pCDMAFrameErrRate . . . 268
8.224.2.2 pGSMBER . . . 268
8.224.2.3 pHDRPackErrRate . . . 268
8.224.2.4 pWCDMABER . . . 268
8.225 GetHRPDStatsResp Struct Reference . . . 269
8.225.1 Detailed Description . . . 269
8.225.2 Field Documentation . . . 269
8.225.2.1 pDRCParams . . . 269
8.225.2.2 pPilotSetData . . . 269
8.225.2.3 pUATI . . . 269
8.226 GetIMSSMSConfigParams Struct Reference . . . 269
8.226.1 Detailed Description . . . 269
8.226.2 Field Documentation . . . 270

8.226.2.1 pPhoneCtxtURI . . . 270
8.226.2.2 pPhoneCtxtURILen . . . 270
8.226.2.3 pSettingResp . . . 270
8.226.2.4 pSMSFormat . . . 270
8.226.2.5 pSMSOverIPNwInd . . . 270
8.227 GetIMSUserConfigParams Struct Reference . . . 270
8.227.1 Detailed Description . . . 270
8.227.2 Field Documentation . . . 271
8.227.2.1 pIMSDomain . . . 271
8.227.2.2 pIMSDomainLen . . . 271
8.227.2.3 pSettingResp . . . 271
8.228 GetIMSVoIPConfigResp Struct Reference . . . 271
8.228.1 Detailed Description . . . 271
8.228.2 Field Documentation . . . 273
8.228.2.1 pAmrMode . . . 273
8.228.2.2 pAmrOctetAligned . . . 273
8.228.2.3 pAmrWbEnable . . . 273
8.228.2.4 pAmrWBMode . . . 273
8.228.2.5 pAmrWBOctetAligned . . . 273
8.228.2.6 pMinSessionExpiryTimer . . . 273
8.228.2.7 pRingBackTimer . . . 273
8.228.2.8 pRingingTimer . . . 273
8.228.2.9 pRTPRTCPInactTimer . . . 273
8.228.2.10 pScrAmrEnable . . . 273
8.228.2.11 pScrAmrWbEnable . . . 273
8.228.2.12 pSessionExpiryTimer . . . 273
8.228.2.13 pSettingResp . . . 273
8.229 GetInstIDResp Struct Reference . . . 273
8.229.1 Field Documentation . . . 273
8.229.1.1 pInstanceID . . . 273
8.229.1.2 pIPFamily . . . 274
8.230 GetM2MAudioProfileReq Struct Reference . . . 274
8.230.1 Detailed Description . . . 274
8.230.2 Field Documentation . . . 274
8.230.2.1 pGenerator . . . 274
8.231 GetM2MAudioProfileResp Struct Reference . . . 274
8.231.1 Detailed Description . . . 274
8.231.2 Field Documentation . . . 275
8.231.2.1 CwtMute . . . 275
8.231.2.2 EarMute . . . 275

8.231.2.3 Generator . . . 275
8.231.2.4 MicMute . . . 275
8.231.2.5 Profile . . . 275
8.231.2.6 Volume . . . 275
8.232 GetM2MAudioVolumeReq Struct Reference . . . 275
8.232.1 Detailed Description . . . 275
8.232.2 Field Documentation . . . 276
8.232.2.1 Generator . . . 276
8.232.2.2 Profile . . . 276
8.233 GetM2MAudioVolumeResp Struct Reference . . . 276
8.233.1 Detailed Description . . . 276
8.233.2 Field Documentation . . . 276
8.233.2.1 Level . . . 276
8.234 GetM2MAVMuteReq Struct Reference . . . 276
8.234.1 Detailed Description . . . 276
8.234.2 Field Documentation . . . 276
8.234.2.1 Profile . . . 277
8.235 GetM2MAVMuteResp Struct Reference . . . 277
8.235.1 Detailed Description . . . 277
8.235.2 Field Documentation . . . 277
8.235.2.1 CwtMute . . . 277
8.235.2.2 EarMute . . . 277
8.235.2.3 MicMute . . . 277
8.236 GetM2MSpkrGainReq Struct Reference . . . 277
8.236.1 Detailed Description . . . 277
8.236.2 Field Documentation . . . 278
8.236.2.1 Profile . . . 278
8.237 GetM2MSpkrGainResp Struct Reference . . . 278
8.237.1 Detailed Description . . . 278
8.237.2 Field Documentation . . . 278
8.237.2.1 Value . . . 278
8.238 getMsgWaitingInfo Struct Reference . . . 278
8.238.1 Detailed Description . . . 278
8.238.2 Field Documentation . . . 279
8.238.2.1 msgWaitInfo . . . 279
8.238.2.2 numInstances . . . 279
8.239 GetNetworkTimeResp Struct Reference . . . 279
8.239.1 Detailed Description . . . 279
8.239.2 Field Documentation . . . 279
8.239.2.1 p3GPP2TimeInfo . . . 279

8.239.2.2 p3GPPTimeInfo . . . 279
8.240 GetRegMgrConfigParams Struct Reference . . . 279
8.240.1 Detailed Description . . . 279
8.240.2 Field Documentation . . . 280
8.240.2.1 pIMSTestMode . . . 280
8.240.2.2 pPCSCFPort . . . 280
8.240.2.3 pPriCSCFPortName . . . 280
8.240.2.4 pPriCSCFPortNameLen . . . 280
8.240.2.5 pSettingResp . . . 280
8.241 GetSessionIDResp Struct Reference . . . 280
8.241.1 Field Documentation . . . 280
8.241.1.1 pSessionIDv4 . . . 280
8.241.1.2 pSessionIDv6 . . . 280
8.242 GetSIPConfigResp Struct Reference . . . 280
8.242.1 Detailed Description . . . 281
8.242.2 Field Documentation . . . 281
8.242.2.1 pSettingResp . . . 281
8.242.2.2 pSigCompEnabled . . . 281
8.242.2.3 pSIPLocalPort . . . 281
8.242.2.4 pSubscribeTimer . . . 281
8.242.2.5 pTimerSIPReg . . . 281
8.242.2.6 pTimerT1 . . . 281
8.242.2.7 pTimerT2 . . . 281
8.242.2.8 pTimerTf . . . 281
8.243 GnssData Struct Reference . . . 282
8.243.1 Detailed Description . . . 282
8.243.2 Field Documentation . . . 283
8.243.2.1 mask . . . 283
8.244 gnssSvInfoNotification Struct Reference . . . 283
8.244.1 Detailed Description . . . 284
8.244.2 Field Documentation . . . 284
8.244.2.1 bAltitudeAssumed . . . 284
8.244.2.2 pSatelliteInfo . . . 284
8.245 GPRSQoS Struct Reference . . . 284
8.245.1 Detailed Description . . . 284
8.245.2 Field Documentation . . . 285
8.245.2.1 delayClass . . . 285
8.245.2.2 meanThroughputClass . . . 285
8.245.2.3 peakThroughputClass . . . 285
8.245.2.4 precedenceClass . . . 285

8.245.2.5 reliabilityClass . . . 285
8.246 GPRSRequestedQoS Struct Reference . . . 285
8.246.1 Detailed Description . . . 285
8.246.2 Field Documentation . . . 286
8.246.2.1 delayClass . . . 286
8.246.2.2 meanThroughputClass . . . 286
8.246.2.3 peakThroughputClass . . . 286
8.246.2.4 precedenceClass . . . 286
8.246.2.5 reliabilityClass . . . 286
8.247 GPSStateInfo Struct Reference . . . 286
8.247.1 Detailed Description . . . 286
8.247.2 Field Documentation . . . 289
8.247.2.1 Altitude . . . 289
8.247.2.2 EngineState . . . 289
8.247.2.3 glo_almanac_sv_msk . . . 289
8.247.2.4 glo_ephemeris_sv_msk . . . 289
8.247.2.5 glo_health_sv_msk . . . 289
8.247.2.6 glo_visible_sv_msk . . . 289
8.247.2.7 gps_almanac_sv_msk . . . 289
8.247.2.8 gps_ephemeris_sv_msk . . . 289
8.247.2.9 gps_health_sv_msk . . . 289
8.247.2.10 gps_visible_sv_msk . . . 289
8.247.2.11 HorizontalUncertainty . . . 289
8.247.2.12 iono_valid . . . 289
8.247.2.13 Latitude . . . 289
8.247.2.14 Longitude . . . 289
8.247.2.15 sbas_almanac_sv_msk . . . 289
8.247.2.16 sbas_ephemeris_sv_msk . . . 289
8.247.2.17 sbas_health_sv_msk . . . 290
8.247.2.18 sbas_visible_sv_msk . . . 290
8.247.2.19 Time_uncert_ms . . . 290
8.247.2.20 TimeStmp_gps_week . . . 290
8.247.2.21 TimeStmp_tow_ms . . . 290
8.247.2.22 ValidMask . . . 290
8.247.2.23 VerticalUncertainty . . . 290
8.247.2.24 xtra_start_gps_minutes . . . 290
8.247.2.25 xtra_start_gps_week . . . 290
8.247.2.26 xtra_valid_duration_hours . . . 290
8.248 gpsTime_s Struct Reference . . . 290
8.248.1 Detailed Description . . . 290

8.248.2 Field Documentation . . . 290
8.248.2.1 gpsTimeOfWeekMs . . . 290
8.248.2.2 gpsWeek . . . 290
8.249 gsmCellInfo Struct Reference . . . 290
8.249.1 Detailed Description . . . 291
8.249.2 Field Documentation . . . 291
8.249.2.1 arfcn . . . 291
8.249.2.2 band1900 . . . 291
8.249.2.3 bsicId . . . 291
8.249.2.4 cellIdValid . . . 291
8.249.2.5 rssi . . . 291
8.249.2.6 srxlev . . . 291
8.250 GSMRSSIThresh Struct Reference . . . 292
8.250.1 Detailed Description . . . 292
8.250.2 Field Documentation . . . 292
8.250.2.1 GSMRSSIThreshListLen . . . 292
8.250.2.2 pGSMRSSIThreshList . . . 292
8.251 GSMSrvStatusInfo Struct Reference . . . 292
8.251.1 Detailed Description . . . 292
8.251.2 Field Documentation . . . 293
8.251.2.1 isPrefDataPath . . . 293
8.251.2.2 srvStatus . . . 293
8.251.2.3 trueSrvStatus . . . 293
8.252 GSMSysInfo Struct Reference . . . 293
8.252.1 Detailed Description . . . 293
8.252.2 Field Documentation . . . 295
8.252.2.1 cellId . . . 295
8.252.2.2 cellIdValid . . . 295
8.252.2.3 dtmSupp . . . 295
8.252.2.4 dtmSuppValid . . . 295
8.252.2.5 egprsSupp . . . 295
8.252.2.6 egprsSuppValid . . . 295
8.252.2.7 lac . . . 296
8.252.2.8 lacValid . . . 296
8.252.2.9 MCC . . . 296
8.252.2.10 MNC . . . 296
8.252.2.11 networkIdValid . . . 296
8.252.2.12 regRejectInfoValid . . . 296
8.252.2.13 rejCause . . . 296
8.252.2.14 rejectSrvDomain . . . 296

8.252.2.15 sysInfoGSM . . . 296
8.253 gyroAcceptReady_s Struct Reference . . . 296
8.253.1 Detailed Description . . . 296
8.253.2 Field Documentation . . . 296
8.253.2.1 batchPerSec . . . 296
8.253.2.2 injectEnable . . . 297
8.253.2.3 samplesPerBatch . . . 297
8.254 gyroTempAcceptReady_s Struct Reference . . . 297
8.254.1 Detailed Description . . . 297
8.254.2 Field Documentation . . . 297
8.254.2.1 batchPerSec . . . 297
8.254.2.2 injectEnable . . . 297
8.254.2.3 samplesPerBatch . . . 297
8.255 HDRECIOThresh Struct Reference . . . 297
8.255.1 Detailed Description . . . 298
8.255.2 Field Documentation . . . 298
8.255.2.1 HDRECIOThreshListLen . . . 298
8.255.2.2 pHDRECIOThreshList . . . 298
8.256 HDRIOThresh Struct Reference . . . 298
8.256.1 Detailed Description . . . 298
8.256.2 Field Documentation . . . 298
8.256.2.1 HDRIOThreshListLen . . . 298
8.256.2.2 pHDRIOThreshList . . . 298
8.257 HDRPersonalityInd Struct Reference . . . 298
8.257.1 Field Documentation . . . 299
8.257.1.1 pCurrentPersonality . . . 299
8.257.1.2 pPersonalityListLength . . . 299
8.257.1.3 pProtocolSubtypeElement . . . 299
8.258 HDRPersonalityResp Struct Reference . . . 299
8.258.1 Detailed Description . . . 299
8.258.2 Field Documentation . . . 299
8.258.2.1 pCurrentPersonality . . . 299
8.258.2.2 pPersonalityListLength . . . 299
8.258.2.3 pProtocolSubtypeElement . . . 299
8.259 HDRProtSubtypResp Struct Reference . . . 299
8.259.1 Detailed Description . . . 300
8.259.2 Field Documentation . . . 300
8.259.2.1 pAppSubType . . . 300
8.259.2.2 pCurrentPrsnlty . . . 300
8.259.2.3 pPersonalityListLength . . . 300

8.259.2.4 pProtoSubTypElmnt . . . 300
8.260 HDRRSSIThresh Struct Reference . . . 300
8.260.1 Detailed Description . . . 300
8.260.2 Field Documentation . . . 301
8.260.2.1 HDRRSSIThreshListLen . . . 301
8.260.2.2 pHDRRSSIThreshList . . . 301
8.261 HDRSINRThresh Struct Reference . . . 301
8.261.1 Detailed Description . . . 301
8.261.2 Field Documentation . . . 301
8.261.2.1 HDRSINRThresListLen . . . 301
8.261.2.2 pHDRSINRThresList . . . 301
8.262 HDRSINRThreshold Struct Reference . . . 301
8.262.1 Detailed Description . . . 302
8.262.2 Field Documentation . . . 302
8.262.2.1 HDRSINRThreshListLen . . . 302
8.262.2.2 pHDRSINRThreshList . . . 302
8.263 HDRSSInfo Struct Reference . . . 302
8.263.1 Detailed Description . . . 302
8.263.2 Field Documentation . . . 303
8.263.2.1 ecio . . . 303
8.263.2.2 io . . . 303
8.263.2.3 rssi . . . 303
8.263.2.4 sinr . . . 303
8.264 hdrSSInfo Struct Reference . . . 303
8.264.1 Detailed Description . . . 303
8.264.2 Field Documentation . . . 304
8.264.2.1 ecio . . . 304
8.264.2.2 io . . . 304
8.264.2.3 rssi . . . 304
8.264.2.4 sinr . . . 304
8.265 HDRSysInfo Struct Reference . . . 304
8.265.1 Detailed Description . . . 305
8.265.2 Field Documentation . . . 306
8.265.2.1 hdrActiveProt . . . 306
8.265.2.2 hdrActiveProtValid . . . 306
8.265.2.3 hdrPersonality . . . 306
8.265.2.4 hdrPersonalityValid . . . 306
8.265.2.5 is856SysId . . . 306
8.265.2.6 is856SysIdValid . . . 306
8.265.2.7 isSysPrlMatch . . . 306

8.265.2.8 isSysPrlMatchValid . . . 306
8.265.2.9 sysInfoHDR . . . 306
8.266 homeSIDNID Struct Reference . . . 306
8.266.1 Detailed Description . . . 306
8.266.2 Field Documentation . . . 307
8.266.2.1 numInstances . . . 307
8.266.2.2 SidNid . . . 307
8.267 hotSwapStatus Struct Reference . . . 307
8.267.1 Detailed Description . . . 307
8.267.2 Field Documentation . . . 307
8.267.2.1 hotSwap . . . 307
8.267.2.2 hotSwapLength . . . 307
8.268 image_info_t Struct Reference . . . 307
8.268.1 Detailed Description . . . 307
8.268.2 Field Documentation . . . 308
8.268.2.1 buildID . . . 308
8.268.2.2 buildIDLen . . . 308
8.268.2.3 imageType . . . 308
8.268.2.4 uniqueID . . . 308
8.269 ImageElement Struct Reference . . . 308
8.269.1 Detailed Description . . . 308
8.269.2 Field Documentation . . . 309
8.269.2.1 buildId . . . 309
8.269.2.2 buildIdLength . . . 309
8.269.2.3 imageId . . . 309
8.269.2.4 imageType . . . 309
8.270 ImageIdElement Struct Reference . . . 309
8.270.1 Detailed Description . . . 309
8.270.2 Field Documentation . . . 309
8.270.2.1 buildID . . . 310
8.270.2.2 buildIDLength . . . 310
8.270.2.3 failureCount . . . 310
8.270.2.4 imageID . . . 310
8.270.2.5 storageIndex . . . 310
8.271 ImageIDEntries Struct Reference . . . 310
8.271.1 Detailed Description . . . 310
8.271.2 Field Documentation . . . 310
8.271.2.1 executingImage . . . 310
8.271.2.2 imageIDElement . . . 310
8.271.2.3 imageIDSize . . . 310

8.271.2.4 imageType . . . 310
8.271.2.5 maxImages . . . 311
8.272 ImageList Struct Reference . . . 311
8.272.1 Detailed Description . . . 311
8.272.2 Field Documentation . . . 311
8.272.2.1 imageIDEntries . . . 311
8.272.2.2 listSize . . . 311
8.273 IMSAIndRegisterInfo Struct Reference . . . 311
8.273.1 Detailed Description . . . 311
8.273.2 Field Documentation . . . 312
8.273.2.1 pPdpStatusConfig . . . 312
8.273.2.2 pRatHandoverStatusConfig . . . 312
8.273.2.3 pRegStatusConfig . . . 312
8.273.2.4 pServiceStatusConfig . . . 312
8.274 imsaPdpStatusInfo Struct Reference . . . 312
8.274.1 Detailed Description . . . 312
8.274.2 Field Documentation . . . 313
8.274.2.1 connetionState . . . 313
8.274.2.2 pFailErrorCode . . . 313
8.275 imsaRatStatusInfo Struct Reference . . . 313
8.275.1 Detailed Description . . . 313
8.275.2 Field Documentation . . . 313
8.275.2.1 pErrorCodeStr . . . 313
8.275.2.2 pRATStatus . . . 313
8.275.2.3 pSrcRAT . . . 313
8.275.2.4 pTgtRAT . . . 314
8.276 IMSARegistrationStatus Struct Reference . . . 314
8.276.1 Detailed Description . . . 314
8.276.2 Field Documentation . . . 314
8.276.2.1 pImsRegErrCode . . . 314
8.276.2.2 pImsRegStatus . . . 314
8.276.2.3 pNewImsRegStatus . . . 314
8.277 imsaRegStatusInfo Struct Reference . . . 314
8.277.1 Detailed Description . . . 315
8.277.2 Field Documentation . . . 315
8.277.2.1 pbIMSRegistered . . . 315
8.277.2.2 pImsRegStatus . . . 315
8.277.2.3 pRegStatusErrorCode . . . 315
8.278 IMSAServiceStatus Struct Reference . . . 315
8.278.1 Detailed Description . . . 315

8.278.2 Field Documentation . . . 317
8.278.2.1 pSmsServiceRat . . . 317
8.278.2.2 pSmsServiceStatus . . . 317
8.278.2.3 pUtServiceRat . . . 317
8.278.2.4 pUtServiceStatus . . . 317
8.278.2.5 pVoipServiceRat . . . 317
8.278.2.6 pVoipServiceStatus . . . 317
8.278.2.7 pVsServiceRat . . . 317
8.278.2.8 pVsServiceStatus . . . 317
8.278.2.9 pVtServiceRat . . . 317
8.278.2.10 pVtServiceStatus . . . 317
8.279 IMSASupportedFieldsResp Struct Reference . . . 317
8.279.1 Detailed Description . . . 317
8.279.2 Field Documentation . . . 318
8.279.2.1 pIndFieldsList . . . 318
8.279.2.2 pReqFieldsList . . . 318
8.279.2.3 pRespFieldsList . . . 318
8.280 IMSASupportedMsgInfo Struct Reference . . . 318
8.280.1 Detailed Description . . . 318
8.280.2 Field Documentation . . . 318
8.280.2.1 pSupportedMsgList . . . 318
8.281 imsaSvcStatusInfo Struct Reference . . . 318
8.281.1 Detailed Description . . . 318
8.281.2 Field Documentation . . . 319
8.281.2.1 pSMSSvcRAT . . . 319
8.281.2.2 pSMSSvcStatus . . . 319
8.281.2.3 pUTSvcRAT . . . 319
8.281.2.4 pUTSvcStatus . . . 319
8.281.2.5 pVOIPSvcRAT . . . 319
8.281.2.6 pVOIPSvcStatus . . . 319
8.281.2.7 pVTSvcRAT . . . 319
8.281.2.8 pVTSvcStatus . . . 319
8.282 imsCfgIndRegisterInfo Struct Reference . . . 319
8.282.1 Detailed Description . . . 319
8.282.2 Field Documentation . . . 320
8.282.2.1 pRegMgrConfigEvents . . . 320
8.282.2.2 pSIPConfigEvents . . . 320
8.282.2.3 pSMSConfigEvents . . . 320
8.282.2.4 pUserConfigEvents . . . 320
8.282.2.5 pVoIPConfigEvents . . . 320

8.283 imsRegMgrConfigInfo Struct Reference . . . 320
8.283.1 Detailed Description . . . 321
8.283.2 Field Documentation . . . 321
8.283.2.1 pCSCFPortName . . . 321
8.283.2.2 pIMSTestMode . . . 321
8.283.2.3 pPriCSCFPort . . . 321
8.284 imsSIPConfigInfo Struct Reference . . . 321
8.284.1 Detailed Description . . . 321
8.284.2 Field Documentation . . . 322
8.284.2.1 pSigCompEnabled . . . 322
8.284.2.2 pSIPLocalPort . . . 322
8.284.2.3 pSubscribeTimer . . . 322
8.284.2.4 pTimerSIPReg . . . 322
8.284.2.5 pTimerT1 . . . 322
8.284.2.6 pTimerT2 . . . 322
8.284.2.7 pTimerTf . . . 322
8.285 imsSMSConfigInfo Struct Reference . . . 322
8.285.1 Detailed Description . . . 322
8.285.2 Field Documentation . . . 323
8.285.2.1 pPhoneCtxtURI . . . 323
8.285.2.2 pSMSFormat . . . 323
8.285.2.3 pSMSOverIPNwInd . . . 323
8.286 imsUserConfigInfo Struct Reference . . . 323
8.286.1 Detailed Description . . . 323
8.286.2 Field Documentation . . . 323
8.286.2.1 pIMSDomain . . . 323
8.287 imsVoIPConfigInfo Struct Reference . . . 323
8.287.1 Detailed Description . . . 324
8.287.2 Field Documentation . . . 325
8.287.2.1 pAmrMode . . . 325
8.287.2.2 pAmrOctetAligned . . . 325
8.287.2.3 pAmrWbEnable . . . 325
8.287.2.4 pAmrWBMode . . . 325
8.287.2.5 pAmrWBOctetAligned . . . 325
8.287.2.6 pMinSessionExpiryTimer . . . 325
8.287.2.7 pRingBackTimer . . . 326
8.287.2.8 pRingingTimer . . . 326
8.287.2.9 pRTPRTCPInactTimer . . . 326
8.287.2.10 pScrAmrEnable . . . 326
8.287.2.11 pScrAmrWbEnable . . . 326

8.287.2.12 pSessionExpiryTimer . . . 326
8.288 IndFieldsList Struct Reference . . . 326
8.288.1 Detailed Description . . . 326
8.288.2 Field Documentation . . . 326
8.288.2.1 indicationFields . . . 326
8.288.2.2 indicationFieldsLen . . . 326
8.289 infoInterFreq Struct Reference . . . 326
8.289.1 Detailed Description . . . 326
8.289.2 Field Documentation . . . 327
8.289.2.1 cell_resel_priority . . . 327
8.289.2.2 cellInterFreqParams . . . 327
8.289.2.3 cells_len . . . 327
8.289.2.4 earfcn . . . 327
8.289.2.5 threshXHigh . . . 327
8.289.2.6 threshXLow . . . 327
8.290 IOThresh Struct Reference . . . 327
8.290.1 Detailed Description . . . 328
8.290.2 Field Documentation . . . 328
8.290.2.1 IOThresListLen . . . 328
8.290.2.2 pIOThresList . . . 328
8.291 IPv4Addr Struct Reference . . . 328
8.291.1 Detailed Description . . . 328
8.291.2 Field Documentation . . . 328
8.291.2.1 addr . . . 328
8.291.2.2 subnetMask . . . 328
8.292 IPv4Info Struct Reference . . . 328
8.292.1 Detailed Description . . . 329
8.292.2 Field Documentation . . . 329
8.292.2.1 address . . . 329
8.292.2.2 port . . . 329
8.292.2.3 TlvPresent . . . 329
8.293 IPv6Addr Struct Reference . . . 329
8.293.1 Detailed Description . . . 329
8.293.2 Field Documentation . . . 329
8.293.2.1 addr . . . 329
8.293.2.2 prefixLen . . . 329
8.294 IPV6AddressInfo Struct Reference . . . 330
8.294.1 Detailed Description . . . 330
8.294.2 Field Documentation . . . 330
8.294.2.1 IPAddressV6 . . . 330

8.294.2.2 IPV6PrefixLen . . . 330
8.295 ipv6AddressInfo Struct Reference . . . 330
8.295.1 Detailed Description . . . 330
8.295.2 Field Documentation . . . 330
8.295.2.1 IPAddressV6 . . . 330
8.295.2.2 IPV6PrefixLen . . . 330
8.296 IPV6GWAddressInfo Struct Reference . . . 330
8.296.1 Detailed Description . . . 331
8.296.2 Field Documentation . . . 331
8.296.2.1 gwAddressV6 . . . 331
8.296.2.2 gwV6PrefixLen . . . 331
8.297 IPv6Info Struct Reference . . . 331
8.297.1 Detailed Description . . . 331
8.297.2 Field Documentation . . . 331
8.297.2.1 address . . . 332
8.297.2.2 port . . . 332
8.297.2.3 TlvPresent . . . 332
8.298 IPv6TrafCls Struct Reference . . . 332
8.298.1 Detailed Description . . . 332
8.298.2 Field Documentation . . . 332
8.298.2.1 mask . . . 332
8.298.2.2 val . . . 332
8.299 LibPackGPRSRequestedQoS Struct Reference . . . 332
8.299.1 Detailed Description . . . 332
8.299.2 Field Documentation . . . 333
8.299.2.1 delayClass . . . 333
8.299.2.2 meanThroughputClass . . . 333
8.299.2.3 peakThroughputClass . . . 333
8.299.2.4 precedenceClass . . . 333
8.299.2.5 reliabilityClass . . . 333
8.300 LibpackProfile3GPP Struct Reference . . . 333
8.300.1 Detailed Description . . . 334
8.300.2 Field Documentation . . . 337
8.300.2.1 pAddrAllocPref . . . 337
8.300.2.2 pAPNClass . . . 337
8.300.2.3 pAPNDisabledFlag . . . 337
8.300.2.4 pAPNName . . . 337
8.300.2.5 pAPNnameSize . . . 338
8.300.2.6 pAuthenticationPref . . . 338
8.300.2.7 pGPRSMinimumQoS . . . 338

8.300.2.8 pGPRSRequestedQos . . . 338
8.300.2.9 pImCnFlag . . . 338
8.300.2.10 pIPv4AddrPref . . . 338
8.300.2.11 pIPv6AddPref . . . 338
8.300.2.12 pPassword . . . 338
8.300.2.13 pPasswordSize . . . 338
8.300.2.14 pPcscfAddrUsingDhcp . . . 338
8.300.2.15 pPcscfAddrUsingPCO . . . 338
8.300.2.16 pPDNInactivTimeout . . . 338
8.300.2.17 pPdpAccessConFlag . . . 338
8.300.2.18 pPdpContext . . . 338
8.300.2.19 pPdpDataCompType . . . 338
8.300.2.20 pPdpHdrCompType . . . 338
8.300.2.21 pPDPtype . . . 338
8.300.2.22 pPriDNSIPv4AddPref . . . 338
8.300.2.23 pPriDNSIPv6addpref . . . 338
8.300.2.24 pPrimaryID . . . 338
8.300.2.25 pProfilename . . . 338
8.300.2.26 pProfilenameSize . . . 338
8.300.2.27 pQosClassID . . . 338
8.300.2.28 pSecDNSIPv4AddPref . . . 338
8.300.2.29 pSecDNSIPv6addpref . . . 338
8.300.2.30 pSecondaryFlag . . . 338
8.300.2.31 pTFTID1Params . . . 338
8.300.2.32 pTFTID2Params . . . 338
8.300.2.33 pUMTSMinQoS . . . 339
8.300.2.34 pUMTSMinQosSigInd . . . 339
8.300.2.35 pUMTSReqQoS . . . 339
8.300.2.36 pUMTSReqQoSSigInd . . . 339
8.300.2.37 pUsername . . . 339
8.300.2.38 pUsernameSize . . . 339
8.301 LibpackProfile3GPP2 Struct Reference . . . 339
8.301.1 Detailed Description . . . 339
8.301.2 Field Documentation . . . 343
8.301.2.1 pAllowLinger . . . 343
8.301.2.2 pAPNClass3GPP2 . . . 343
8.301.2.3 pAPNEnabled3GPP2 . . . 343
8.301.2.4 pApnString . . . 343
8.301.2.5 pApnStringSize . . . 343
8.301.2.6 pAppPriority . . . 343

8.301.2.7 pAppType . . . 343
8.301.2.8 pAuthPassword . . . 343
8.301.2.9 pAuthPasswordSize . . . 343
8.301.2.10 pAuthProtocol . . . 343
8.301.2.11 pAuthRetryCount . . . 343
8.301.2.12 pAuthTimeout . . . 343
8.301.2.13 pDataMode . . . 343
8.301.2.14 pDataRate . . . 343
8.301.2.15 pIpcpAckTimeout . . . 343
8.301.2.16 pIpcpCreqRetryCount . . . 343
8.301.2.17 pIsPcscfAddressNedded . . . 343
8.301.2.18 pLcpAckTimeout . . . 343
8.301.2.19 pLcpCreqRetryCount . . . 344
8.301.2.20 pNegoDnsSrvrPref . . . 344
8.301.2.21 pPDNInactivTimeout3GPP2 . . . 344
8.301.2.22 pPdnType . . . 344
8.301.2.23 pPppSessCloseTimer1x . . . 344
8.301.2.24 pPppSessCloseTimerDO . . . 344
8.301.2.25 pPrimaryV4DnsAddress . . . 344
8.301.2.26 pPriV6DnsAddress . . . 344
8.301.2.27 pRATType . . . 344
8.301.2.28 pSecondaryV4DnsAddress . . . 344
8.301.2.29 pSecV6DnsAddress . . . 344
8.301.2.30 pUserId . . . 344
8.301.2.31 pUserIdSize . . . 344
8.302 LibPackprofile_3GPP Struct Reference . . . 344
8.302.1 Detailed Description . . . 345
8.302.2 Field Documentation . . . 348
8.302.2.1 pAddrAllocPref . . . 348
8.302.2.2 pAPNClass . . . 348
8.302.2.3 pAPNDisabledFlag . . . 348
8.302.2.4 pAPNName . . . 348
8.302.2.5 pAPNnameSize . . . 349
8.302.2.6 pAuthenticationPref . . . 349
8.302.2.7 pGPRSMinimumQoS . . . 349
8.302.2.8 pGPRSRequestedQos . . . 349
8.302.2.9 pImCnFlag . . . 349
8.302.2.10 pIPv4AddrPref . . . 349
8.302.2.11 pIPv6AddPref . . . 349
8.302.2.12 pPassword . . . 349

8.302.2.13 pPasswordSize . . . 349
8.302.2.14 pPcscfAddrUsingDhcp . . . 349
8.302.2.15 pPcscfAddrUsingPCO . . . 349
8.302.2.16 pPDNInactivTimeout . . . 349
8.302.2.17 pPdpAccessConFlag . . . 349
8.302.2.18 pPdpContext . . . 349
8.302.2.19 pPdpDataCompType . . . 349
8.302.2.20 pPdpHdrCompType . . . 349
8.302.2.21 pPDPtype . . . 349
8.302.2.22 pPriDNSIPv4AddPref . . . 349
8.302.2.23 pPriDNSIPv6addpref . . . 349
8.302.2.24 pPrimaryID . . . 349
8.302.2.25 pProfilename . . . 349
8.302.2.26 pProfilenameSize . . . 349
8.302.2.27 pQosClassID . . . 349
8.302.2.28 pSecDNSIPv4AddPref . . . 349
8.302.2.29 pSecDNSIPv6addpref . . . 349
8.302.2.30 pSecondaryFlag . . . 349
8.302.2.31 pTFTID1Params . . . 349
8.302.2.32 pTFTID2Params . . . 349
8.302.2.33 pUMTSMinQoS . . . 350
8.302.2.34 pUMTSMinQosSigInd . . . 350
8.302.2.35 pUMTSReqQoS . . . 350
8.302.2.36 pUMTSReqQoSSigInd . . . 350
8.302.2.37 pUsername . . . 350
8.302.2.38 pUsernameSize . . . 350
8.303 LibPackprofile_3GPP2 Struct Reference . . . 350
8.303.1 Detailed Description . . . 350
8.303.2 Field Documentation . . . 354
8.303.2.1 pAllowLinger . . . 354
8.303.2.2 pAPNClass3GPP2 . . . 354
8.303.2.3 pAPNEnabled3GPP2 . . . 354
8.303.2.4 pApnString . . . 354
8.303.2.5 pApnStringSize . . . 354
8.303.2.6 pAppPriority . . . 354
8.303.2.7 pAppType . . . 354
8.303.2.8 pAuthPassword . . . 354
8.303.2.9 pAuthPassword_tSize . . . 354
8.303.2.10 pAuthProtocol . . . 354
8.303.2.11 pAuthRetryCount . . . 354

8.303.2.12 pAuthTimeout . . . 354
8.303.2.13 pDataMode . . . 354
8.303.2.14 pDataRate . . . 354
8.303.2.15 pIpcpAckTimeout . . . 354
8.303.2.16 pIpcpCreqRetryCount . . . 354
8.303.2.17 pIsPcscfAddressNedded . . . 354
8.303.2.18 pLcpAckTimeout . . . 354
8.303.2.19 pLcpCreqRetryCount . . . 355
8.303.2.20 pNegoDnsSrvrPref . . . 355
8.303.2.21 pPDNInactivTimeout3GPP2 . . . 355
8.303.2.22 pPdnType . . . 355
8.303.2.23 pPppSessCloseTimer1x . . . 355
8.303.2.24 pPppSessCloseTimerDO . . . 355
8.303.2.25 pPrimaryV4DnsAddress . . . 355
8.303.2.26 pPriV6DnsAddress . . . 355
8.303.2.27 pRATType . . . 355
8.303.2.28 pSecondaryV4DnsAddress . . . 355
8.303.2.29 pSecV6DnsAddress . . . 355
8.303.2.30 pUserId . . . 355
8.303.2.31 pUserIdSize . . . 355
8.304 LibPackQosClassID Struct Reference . . . 355
8.304.1 Detailed Description . . . 355
8.304.2 Field Documentation . . . 356
8.304.2.1 gDlBitRate . . . 356
8.304.2.2 gUlBitRate . . . 356
8.304.2.3 maxDlBitRate . . . 356
8.304.2.4 maxUlBitRate . . . 356
8.304.2.5 QCI . . . 356
8.305 LibPackTFTIDParams Struct Reference . . . 356
8.305.1 Detailed Description . . . 356
8.305.2 Field Documentation . . . 357
8.305.2.1 destPortRangeEnd . . . 357
8.305.2.2 destPortRangeStart . . . 357
8.305.2.3 eValid . . . 357
8.305.2.4 filterId . . . 357
8.305.2.5 flowLabel . . . 357
8.305.2.6 IPSECSPI . . . 357
8.305.2.7 ipVersion . . . 358
8.305.2.8 nextHeader . . . 358
8.305.2.9 pSourceIP . . . 358

8.305.2.10 sourceIPMask . . . 358
8.305.2.11 srcPortRangeEnd . . . 358
8.305.2.12 srcPortRangeStart . . . 358
8.305.2.13 tosMask . . . 358
8.306 LibPackUMTSQoS Struct Reference . . . 358
8.306.1 Detailed Description . . . 358
8.306.2 Field Documentation . . . 360
8.306.2.1 deliveryErrSDU . . . 360
8.306.2.2 grntDownlinkBitrate . . . 360
8.306.2.3 grntUplinkBitrate . . . 360
8.306.2.4 maxDownlinkBitrate . . . 360
8.306.2.5 maxSDUSize . . . 360
8.306.2.6 maxUplinkBitrate . . . 360
8.306.2.7 qosDeliveryOrder . . . 360
8.306.2.8 resBerRatio . . . 360
8.306.2.9 sduErrorRatio . . . 360
8.306.2.10 trafficClass . . . 360
8.306.2.11 trafficPriority . . . 360
8.306.2.12 transferDelay . . . 360
8.307 LibPackUMTSReqQoSSigInd Struct Reference . . . 360
8.307.1 Detailed Description . . . 360
8.307.2 Field Documentation . . . 361
8.307.2.1 SigInd . . . 361
8.307.2.2 UMTSReqQoS . . . 361
8.308 lineCtrlInfo Struct Reference . . . 361
8.308.1 Detailed Description . . . 361
8.308.2 Field Documentation . . . 361
8.308.2.1 polarityIncluded . . . 361
8.308.2.2 pwrDenialTime . . . 361
8.308.2.3 revPolarity . . . 361
8.308.2.4 toggleMode . . . 361
8.309 loc_BdsSV Struct Reference . . . 361
8.309.1 Detailed Description . . . 362
8.309.2 Field Documentation . . . 362
8.309.2.1 id . . . 362
8.309.2.2 mask . . . 362
8.310 loc_BdsSVInfo Struct Reference . . . 362
8.310.1 Detailed Description . . . 362
8.310.2 Field Documentation . . . 362
8.310.2.1 len . . . 363

8.310.2.2 pSV . . . 363
8.311loc_CellDb Struct Reference . . . 363
8.311.1 Detailed Description . . . 363
8.311.2 Field Documentation . . . 363
8.311.2.1 mask . . . 363
8.312loc_ClkInfo Struct Reference . . . 363
8.312.1 Detailed Description . . . 363
8.312.2 Field Documentation . . . 364
8.312.2.1 mask . . . 364
8.313loc_GnssData Struct Reference . . . 364
8.313.1 Detailed Description . . . 365
8.313.2 Field Documentation . . . 365
8.313.2.1 mask . . . 366
8.314loc_gpsTime Struct Reference . . . 366
8.314.1 Detailed Description . . . 366
8.314.2 Field Documentation . . . 366
8.314.2.1 gpsTimeOfWeekMs . . . 366
8.314.2.2 gpsWeek . . . 366
8.315loc_IPv4Info Struct Reference . . . 366
8.315.1 Detailed Description . . . 366
8.315.2 Field Documentation . . . 367
8.315.2.1 address . . . 367
8.315.2.2 port . . . 367
8.315.2.3 TlvPresent . . . 367
8.316loc_IPv6Info Struct Reference . . . 367
8.316.1 Detailed Description . . . 367
8.316.2 Field Documentation . . . 367
8.316.2.1 address . . . 367
8.316.2.2 port . . . 367
8.316.2.3 TlvPresent . . . 367
8.317loc_LocApplicationInfo Struct Reference . . . 367
8.317.1 Detailed Description . . . 368
8.317.2 Field Documentation . . . 368
8.317.2.1 appNameLength . . . 368
8.317.2.2 appProviderLength . . . 368
8.317.2.3 appVersionLength . . . 368
8.317.2.4 appVersionValid . . . 368
8.317.2.5 pAppName . . . 368
8.317.2.6 pAppProvider . . . 368
8.317.2.7 pAppVersion . . . 368

8.318loc_precisionDilution Struct Reference . . . 368
8.318.1 Detailed Description . . . 369
8.318.2 Field Documentation . . . 369
8.318.2.1 HDOP . . . 369
8.318.2.2 PDOP . . . 369
8.318.2.3 VDOP . . . 369
8.319loc_satelliteInfo Struct Reference . . . 369
8.319.1 Detailed Description . . . 369
8.319.2 Field Documentation . . . 371
8.319.2.1 azimuth . . . 371
8.319.2.2 elevation . . . 371
8.319.2.3 gnssSvId . . . 371
8.319.2.4 healthStatus . . . 371
8.319.2.5 snr . . . 371
8.319.2.6 svInfoMask . . . 371
8.319.2.7 svListLen . . . 371
8.319.2.8 svStatus . . . 371
8.319.2.9 system . . . 371
8.319.2.10validMask . . . 371
8.320loc_sensorDataUsage Struct Reference . . . 371
8.320.1 Detailed Description . . . 371
8.320.2 Field Documentation . . . 372
8.320.2.1 aidingIndicatorMask . . . 372
8.320.2.2 usageMask . . . 372
8.321loc_SV Struct Reference . . . 372
8.321.1 Detailed Description . . . 372
8.321.2 Field Documentation . . . 373
8.321.2.1 id . . . 373
8.321.2.2 mask . . . 373
8.321.2.3 system . . . 373
8.322loc_SVInfo Struct Reference . . . 373
8.322.1 Detailed Description . . . 373
8.322.2 Field Documentation . . . 373
8.322.2.1 len . . . 373
8.322.2.2 pSV . . . 373
8.323loc_svUsedforFix Struct Reference . . . 373
8.323.1 Detailed Description . . . 374
8.323.2 Field Documentation . . . 374
8.323.2.1 gnssSvUsedList . . . 374
8.323.2.2 gnssSvUsedList_len . . . 374

8.324 loc_urlAddr Struct Reference . . . 374
8.324.1 Detailed Description . . . 374
8.324.2 Field Documentation . . . 374
8.324.2.1 address . . . 374
8.324.2.2 TlvPresent . . . 374
8.325 LocApplicationInfo Struct Reference . . . 375
8.325.1 Detailed Description . . . 375
8.325.2 Field Documentation . . . 375
8.325.2.1 appNameLength . . . 375
8.325.2.2 appProviderLength . . . 375
8.325.2.3 appVersionLength . . . 375
8.325.2.4 appVersionValid . . . 376
8.325.2.5 pAppName . . . 376
8.325.2.6 pAppProvider . . . 376
8.325.2.7 pAppVersion . . . 376
8.326 LocDelAssDataReq Struct Reference . . . 376
8.326.1 Detailed Description . . . 376
8.326.2 Field Documentation . . . 376
8.326.2.1 pBdsSVInfo . . . 376
8.326.2.2 pCellDb . . . 376
8.326.2.3 pClkInfo . . . 376
8.326.2.4 pGnssData . . . 377
8.326.2.5 pSVInfo . . . 377
8.327 LOCEventRegisterReqResp Struct Reference . . . 377
8.327.1 Detailed Description . . . 377
8.327.2 Field Documentation . . . 379
8.327.2.1 eventRegister . . . 379
8.328 LOCExtPowerStateReqResp Struct Reference . . . 379
8.328.1 Detailed Description . . . 379
8.328.2 Field Documentation . . . 379
8.328.2.1 extPowerState . . . 379
8.329 LocGetServerReq Struct Reference . . . 379
8.329.1 Detailed Description . . . 379
8.329.2 Field Documentation . . . 380
8.329.2.1 pServerAddrTypeMask . . . 380
8.329.2.2 serverType . . . 380
8.330 LocInjectPositionReq Struct Reference . . . 380
8.330.1 Detailed Description . . . 380
8.330.2 Field Documentation . . . 383
8.330.2.1 pAltitudeSrcInfo . . . 383

8.330.2.2 pAltitudeWrtEllipsoid . . . 383
8.330.2.3 pAltitudeWrtMeanSeaLevel . . . 383
8.330.2.4 pHorConfidence . . . 384
8.330.2.5 pHorReliability . . . 384
8.330.2.6 pHorUncCircular . . . 384
8.330.2.7 pLatitude . . . 384
8.330.2.8 pLongitude . . . 384
8.330.2.9 pPositionSrc . . . 384
8.330.2.10 pRawHorConfidence . . . 384
8.330.2.11 pRawHorUncCircular . . . 384
8.330.2.12 pTimestampAge . . . 384
8.330.2.13 pTimestampUtc . . . 384
8.330.2.14 pVertConfidence . . . 384
8.330.2.15 pVertReliability . . . 384
8.330.2.16 pVertUnc . . . 384
8.331 LocInjectSensorDataReq Struct Reference . . . 384
8.331.1 Detailed Description . . . 384
8.331.2 Field Documentation . . . 385
8.331.2.1 pAcceleroData . . . 385
8.331.2.2 pAcceleroTempData . . . 385
8.331.2.3 pAcceleroTimeSrc . . . 385
8.331.2.4 pGyroData . . . 385
8.331.2.5 pGyroTempData . . . 385
8.331.2.6 pGyroTimeSrc . . . 385
8.331.2.7 pOpaqueIdentifier . . . 385
8.332 LocSetCradleMountReq Struct Reference . . . 385
8.332.1 Detailed Description . . . 386
8.332.2 Field Documentation . . . 386
8.332.2.1 pConfidence . . . 386
8.332.2.2 state . . . 386
8.333 LOCStartReq Struct Reference . . . 386
8.333.1 Detailed Description . . . 386
8.333.2 Field Documentation . . . 387
8.333.2.1 pApplicationInfo . . . 387
8.333.2.2 pConfigAltitudeAssumed . . . 387
8.333.2.3 pHorizontalAccuracyLvl . . . 388
8.333.2.4 pIntermediateReportState . . . 388
8.333.2.5 pMinIntervalTime . . . 388
8.333.2.6 pRecurrenceType . . . 388
8.333.2.7 SessionId . . . 388

8.334 LOCStopReq Struct Reference . . . 388
8.334.1 Detailed Description . . . 388
8.334.2 Field Documentation . . . 388
8.334.2.1 sessionId . . . 388
8.335 LteCQIParm Struct Reference . . . 388
8.335.1 Detailed Description . . . 388
8.335.2 Field Documentation . . . 389
8.335.2.1 CQIValueCW0 . . . 389
8.335.2.2 CQIValueCW1 . . . 389
8.335.2.3 ValidityCW0 . . . 389
8.335.2.4 ValidityCW1 . . . 389
8.336 lteEARFCN Struct Reference . . . 389
8.336.1 Detailed Description . . . 389
8.336.2 Field Documentation . . . 389
8.336.2.1 earfcn0 . . . 390
8.336.2.2 earfcn1 . . . 390
8.336.2.3 status . . . 390
8.337 lteGsmCellInfo Struct Reference . . . 390
8.337.1 Detailed Description . . . 390
8.337.2 Field Documentation . . . 391
8.337.2.1 cellReselPriority . . . 391
8.337.2.2 cells_len . . . 391
8.337.2.3 GsmCellInfo . . . 391
8.337.2.4 nccPermitted . . . 391
8.337.2.5 threshGsmHigh . . . 391
8.337.2.6 threshGsmLow . . . 391
8.338 LTEInfo Struct Reference . . . 391
8.338.1 Detailed Description . . . 391
8.338.2 Field Documentation . . . 393
8.338.2.1 band . . . 393
8.338.2.2 bandwidth . . . 393
8.338.2.3 emmConnState . . . 393
8.338.2.4 emmState . . . 393
8.338.2.5 emmSubState . . . 393
8.338.2.6 RXChan . . . 393
8.338.2.7 TXChan . . . 393
8.339 LTEInfoInterfreq Struct Reference . . . 393
8.339.1 Detailed Description . . . 393
8.339.2 Field Documentation . . . 393
8.339.2.1 freqsLen . . . 393

8.339.2.2 InfoInterfreq . . . 393
8.339.2.3 ueInIdle . . . 393
8.340 LTEInfoIntrafreq Struct Reference . . . 393
8.340.1 Detailed Description . . . 394
8.340.2 Field Documentation . . . 395
8.340.2.1 CellParams . . . 395
8.340.2.2 cellReselPriority . . . 395
8.340.2.3 cellsLen . . . 395
8.340.2.4 earfcn . . . 395
8.340.2.5 globalCellId . . . 395
8.340.2.6 plmn . . . 395
8.340.2.7 servingCellId . . . 395
8.340.2.8 sIntraSearch . . . 395
8.340.2.9 sNonIntraSearch . . . 395
8.340.2.10 tac . . . 395
8.340.2.11 threshServingLow . . . 395
8.340.2.12 ueInIdle . . . 396
8.341 LTEInfoNeighboringGSM Struct Reference . . . 396
8.341.1 Detailed Description . . . 396
8.341.2 Field Documentation . . . 396
8.341.2.1 freqsLen . . . 396
8.341.2.2 LteGsmCellInfo . . . 396
8.341.2.3 ueInIdle . . . 396
8.342 LTEInfoNeighboringWCDMA Struct Reference . . . 396
8.342.1 Detailed Description . . . 396
8.342.2 Field Documentation . . . 397
8.342.2.1 freqsLen . . . 397
8.342.2.2 LTEWCDMACellInfo . . . 397
8.342.2.3 ueInIdle . . . 397
8.343 LteNasReleaseInfo_s Struct Reference . . . 397
8.343.1 Detailed Description . . . 397
8.343.2 Field Documentation . . . 397
8.343.2.1 nas_major . . . 397
8.343.2.2 nas_minor . . . 397
8.343.2.3 nas_release . . . 398
8.344 ltePCI Struct Reference . . . 398
8.344.1 Detailed Description . . . 398
8.344.2 Field Documentation . . . 398
8.344.2.1 earfcn . . . 398
8.344.2.2 pci . . . 398

8.344.2.3 status . . . 398
8.345 IteRsrpinformation Struct Reference . . . 398
8.345.1 Detailed Description . . . 398
8.345.2 Field Documentation . . . 399
8.345.2.1 rsrplevel . . . 399
8.346 LTERSRPThresh Struct Reference . . . 399
8.346.1 Detailed Description . . . 399
8.346.2 Field Documentation . . . 399
8.346.2.1 LTERSRPThreshListLen . . . 399
8.346.2.2 pLTERSRPThreshList . . . 399
8.347 LTERSRQThresh Struct Reference . . . 399
8.347.1 Detailed Description . . . 399
8.347.2 Field Documentation . . . 400
8.347.2.1 LTERSRQThreshListLen . . . 400
8.347.2.2 pLTERSRQThreshList . . . 400
8.348 LTERSSIThresh Struct Reference . . . 400
8.348.1 Detailed Description . . . 400
8.348.2 Field Documentation . . . 400
8.348.2.1 LTERSSIThreshListLen . . . 400
8.348.2.2 pLTERSSIThreshList . . . 400
8.349 LteSccRxInfoResp Struct Reference . . . 400
8.349.1 Detailed Description . . . 400
8.349.2 Field Documentation . . . 400
8.349.2.1 pSccRxInfo . . . 401
8.350 LTESigRptCfg Struct Reference . . . 401
8.350.1 Detailed Description . . . 401
8.350.2 Field Documentation . . . 401
8.350.2.1 avgPeriod . . . 401
8.350.2.2 rptRate . . . 401
8.351 LTESigRptConfig Struct Reference . . . 401
8.351.1 Detailed Description . . . 402
8.351.2 Field Documentation . . . 402
8.351.2.1 avgPeriod . . . 402
8.351.2.2 rptRate . . . 402
8.352 IteSnrinformation Struct Reference . . . 402
8.352.1 Detailed Description . . . 402
8.352.2 Field Documentation . . . 403
8.352.2.1 snrlevel . . . 403
8.353 LTESNRThresh Struct Reference . . . 403
8.353.1 Detailed Description . . . 403

8.353.2 Field Documentation . . . 403
8.353.2.1 LTESNRThresListLen . . . 403
8.353.2.2 pLTESNRThresList . . . 403
8.354 LTESNRThreshold Struct Reference . . . 403
8.354.1 Detailed Description . . . 404
8.354.2 Field Documentation . . . 404
8.354.2.1 LTESNRThreshListLen . . . 404
8.354.2.2 pLTESNRThreshList . . . 404
8.355 LTESSInfo Struct Reference . . . 404
8.355.1 Detailed Description . . . 404
8.355.2 Field Documentation . . . 405
8.355.2.1 rsrp . . . 405
8.355.2.2 rsrq . . . 405
8.355.2.3 rssi . . . 405
8.355.2.4 snr . . . 405
8.356 lteSSInfo Struct Reference . . . 405
8.356.1 Detailed Description . . . 405
8.356.2 Field Documentation . . . 405
8.356.2.1 rsrp . . . 405
8.356.2.2 rsrq . . . 405
8.356.2.3 rssi . . . 405
8.356.2.4 snr . . . 405
8.357 LTESysInfo Struct Reference . . . 406
8.357.1 Detailed Description . . . 406
8.357.2 Field Documentation . . . 407
8.357.2.1 cellId . . . 407
8.357.2.2 cellIdValid . . . 407
8.357.2.3 lac . . . 407
8.357.2.4 lacValid . . . 407
8.357.2.5 MCC . . . 408
8.357.2.6 MNC . . . 408
8.357.2.7 networkIdValid . . . 408
8.357.2.8 regRejectInfoValid . . . 408
8.357.2.9 rejCause . . . 408
8.357.2.10 rejectSrvDomain . . . 408
8.357.2.11 sysInfoLTE . . . 408
8.357.2.12 tac . . . 408
8.357.2.13 tacValid . . . 408
8.358 lteWcdmaCellInfo Struct Reference . . . 408
8.358.1 Detailed Description . . . 408

8.358.2 Field Documentation . . . 409
8.358.2.1 cellReselPriority . . . 409
8.358.2.2 cellsLen . . . 409
8.358.2.3 threshXhigh . . . 409
8.358.2.4 threshXlow . . . 409
8.358.2.5 uarfcn . . . 409
8.358.2.6 WCDMACellInfo . . . 409
8.359 messageModeTlv Struct Reference . . . 409
8.359.1 Detailed Description . . . 409
8.359.2 Field Documentation . . . 409
8.359.2.1 MessageModeInfo . . . 409
8.359.2.2 TlvPresent . . . 409
8.360 messageWaitingInfoContent Struct Reference . . . 409
8.360.1 Detailed Description . . . 410
8.360.2 Field Documentation . . . 410
8.360.2.1 activeInd . . . 410
8.360.2.2 msgCount . . . 410
8.360.2.3 msgType . . . 410
8.361 minBasedIMSI Struct Reference . . . 410
8.361.1 Detailed Description . . . 410
8.361.2 Field Documentation . . . 411
8.361.2.1 imsiM1112 . . . 411
8.361.2.2 imsiMS1 . . . 411
8.361.2.3 imsiMS2 . . . 411
8.361.2.4 mccM . . . 411
8.362 mitigationDevList Struct Reference . . . 411
8.362.1 Detailed Description . . . 411
8.362.2 Field Documentation . . . 411
8.362.2.1 maxMitigationLevel . . . 411
8.362.2.2 mitigationDevId . . . 412
8.362.2.3 mitigationDevIdLen . . . 412
8.363 MNRInfo Struct Reference . . . 412
8.363.1 Detailed Description . . . 412
8.363.2 Field Documentation . . . 412
8.363.2.1 mcc . . . 412
8.363.2.2 mnc . . . 412
8.363.2.3 rat . . . 412
8.364 ModifyProfileIn Struct Reference . . . 412
8.364.1 Detailed Description . . . 412
8.364.2 Field Documentation . . . 413

8.364.2.1 curProfile . . . 413
8.364.2.2 pProfileID . . . 413
8.364.2.3 pProfileType . . . 413
8.365 ModifyProfileOut Struct Reference . . . 413
8.365.1 Detailed Description . . . 413
8.365.2 Field Documentation . . . 413
8.365.2.1 pExtErrorCode . . . 413
8.366 msgWaitingInfo Struct Reference . . . 413
8.366.1 Detailed Description . . . 414
8.366.2 Field Documentation . . . 414
8.366.2.1 msgWaitInfo . . . 414
8.366.2.2 numInstances . . . 414
8.367 namName Struct Reference . . . 414
8.367.1 Detailed Description . . . 414
8.367.2 Field Documentation . . . 414
8.367.2.1 namName . . . 414
8.367.2.2 namNameLen . . . 414
8.368 nas_acqOrderPref Struct Reference . . . 415
8.368.1 Detailed Description . . . 415
8.368.2 Field Documentation . . . 415
8.368.2.1 acqOrdeLen . . . 415
8.368.2.2 pAcqOrder . . . 415
8.369 nas_ActPilotPNElement Struct Reference . . . 415
8.369.1 Detailed Description . . . 415
8.369.2 Field Documentation . . . 415
8.369.2.1 ActSetPilotPN . . . 416
8.369.2.2 ActSetPilotPNStrength . . . 416
8.370 nas_AddCDMASysInfo Struct Reference . . . 416
8.370.1 Detailed Description . . . 416
8.370.2 Field Documentation . . . 416
8.370.2.1 geoSysIdx . . . 416
8.370.2.2 regPrd . . . 416
8.371 nas_AddSysInfo Struct Reference . . . 416
8.371.1 Detailed Description . . . 416
8.371.2 Field Documentation . . . 417
8.371.2.1 cellBroadcastCap . . . 417
8.371.2.2 geoSysIdx . . . 417
8.372 nas_CallBarringSysInfo Struct Reference . . . 417
8.372.1 Detailed Description . . . 417
8.372.2 Field Documentation . . . 417

8.372.2.1 csBarStatus . . . 417
8.372.2.2 psBarStatus . . . 418
8.373 nas_callBarStatus Struct Reference . . . 418
8.373.1 Detailed Description . . . 418
8.373.2 Field Documentation . . . 418
8.373.2.1 csBarStatus . . . 418
8.373.2.2 psBarStatus . . . 418
8.374 nas_CDMAChannel Struct Reference . . . 418
8.374.1 Detailed Description . . . 419
8.374.2 Field Documentation . . . 419
8.374.2.1 priChA . . . 419
8.374.2.2 priChB . . . 419
8.374.2.3 secChA . . . 419
8.374.2.4 secChB . . . 419
8.375 nas_CDMAECIOThresh Struct Reference . . . 419
8.375.1 Detailed Description . . . 419
8.375.2 Field Documentation . . . 419
8.375.2.1 CDMAECIOThreshListLen . . . 420
8.375.2.2 pCDMAECIOThreshList . . . 420
8.376 nas_CDMAInfo Struct Reference . . . 420
8.376.1 Detailed Description . . . 420
8.376.2 Field Documentation . . . 420
8.376.2.1 baseId . . . 420
8.376.2.2 baseLat . . . 420
8.376.2.3 baseLong . . . 421
8.376.2.4 nid . . . 421
8.376.2.5 refpn . . . 421
8.376.2.6 sid . . . 421
8.377 nas_CDMARSSIThresh Struct Reference . . . 421
8.377.1 Detailed Description . . . 421
8.377.2 Field Documentation . . . 421
8.377.2.1 CDMARSSIThreshListLen . . . 421
8.377.2.2 pCDMARSSIThreshList . . . 421
8.378 nas_CDMASysInfo Struct Reference . . . 421
8.378.1 Detailed Description . . . 422
8.378.2 Field Documentation . . . 424
8.378.2.1 baseId . . . 424
8.378.2.2 baseLat . . . 424
8.378.2.3 baseLong . . . 424
8.378.2.4 bsInfoValid . . . 424

8.378.2.5 bsPRev . . . 424
8.378.2.6 bsPRevValid . . . 424
8.378.2.7 ccsSupported . . . 424
8.378.2.8 ccsSupportedValid . . . 424
8.378.2.9 cdmaSysIdValid . . . 424
8.378.2.10 csSysPrlMatch . . . 424
8.378.2.11 isSysPrlMatchValid . . . 424
8.378.2.12 MCC . . . 425
8.378.2.13 MNC . . . 425
8.378.2.14 networkID . . . 425
8.378.2.15 networkIdValid . . . 425
8.378.2.16 packetZone . . . 425
8.378.2.17 packetZoneValid . . . 425
8.378.2.18 pRevInUse . . . 425
8.378.2.19 pRevInUseValid . . . 425
8.378.2.20 sysInfoCDMA . . . 425
8.378.2.21 systemID . . . 425
8.379 nas_CDMASysInfoExt Struct Reference . . . 425
8.379.1 Detailed Description . . . 425
8.379.2 Field Documentation . . . 425
8.379.2.1 imsi_11_12 . . . 425
8.379.2.2 MCC . . . 425
8.380 nas_cellParams Struct Reference . . . 425
8.380.1 Detailed Description . . . 426
8.380.2 Field Documentation . . . 426
8.380.2.1 pci . . . 426
8.380.2.2 rsrp . . . 426
8.380.2.3 rsrq . . . 426
8.380.2.4 rssi . . . 426
8.380.2.5 srxlev . . . 426
8.381 nas_CommInfo Struct Reference . . . 426
8.381.1 Detailed Description . . . 427
8.381.2 Field Documentation . . . 428
8.381.2.1 imsRegState . . . 428
8.381.2.2 modemMode . . . 428
8.381.2.3 psState . . . 428
8.381.2.4 systemMode . . . 428
8.381.2.5 temperature . . . 428
8.382 nas_CSGID Struct Reference . . . 428
8.382.1 Detailed Description . . . 428

8.382.2 Field Documentation . . . 429
8.382.2.1 id . . . 429
8.382.2.2 mcc . . . 429
8.382.2.3 mnc . . . 429
8.382.2.4 mncPcsDigits . . . 429
8.382.2.5 rat . . . 429
8.383 nas_currentPLMN Struct Reference . . . 429
8.383.1 Detailed Description . . . 429
8.383.2 Field Documentation . . . 430
8.383.2.1 MCC . . . 430
8.383.2.2 MNC . . . 430
8.383.2.3 netDescr . . . 430
8.383.2.4 netDescrLength . . . 430
8.384 nas_dataSrvCapabilities Struct Reference . . . 430
8.384.1 Detailed Description . . . 430
8.384.2 Field Documentation . . . 431
8.384.2.1 dataCapabilities . . . 431
8.384.2.2 dataCapabilitiesLen . . . 431
8.385 nas_DataStatusDetail Struct Reference . . . 431
8.385.1 Detailed Description . . . 431
8.385.2 Field Documentation . . . 432
8.385.2.1 IPAddress . . . 432
8.385.2.2 LastErrCode . . . 432
8.386 nas_detailSvcInfo Struct Reference . . . 433
8.386.1 Detailed Description . . . 433
8.386.2 Field Documentation . . . 434
8.386.2.1 hdrHybrid . . . 434
8.386.2.2 hdrSrvStatus . . . 434
8.386.2.3 isSysForbidden . . . 434
8.386.2.4 srvCapability . . . 434
8.386.2.5 srvStatus . . . 434
8.387 nas_DeviceConfigDetail Struct Reference . . . 434
8.387.1 Detailed Description . . . 434
8.387.2 Field Documentation . . . 435
8.387.2.1 Chipset . . . 435
8.387.2.2 HWVersion . . . 435
8.387.2.3 QLIC . . . 435
8.387.2.4 Technology . . . 435
8.388 nas_dirNum Struct Reference . . . 435
8.388.1 Detailed Description . . . 435

8.388.2 Field Documentation . . . 435
8.388.2.1 dirNum . . . 435
8.388.2.2 dirNumLen . . . 435
8.389 nas_DRCParams Struct Reference . . . 435
8.389.1 Detailed Description . . . 436
8.389.2 Field Documentation . . . 436
8.389.2.1 DRCCover . . . 436
8.389.2.2 DRCValue . . . 436
8.390 nas_ecioListElement Struct Reference . . . 436
8.390.1 Detailed Description . . . 436
8.390.2 Field Documentation . . . 436
8.390.2.1 ecio . . . 436
8.390.2.2 radioIf . . . 437
8.391 nas_ECIOThresh Struct Reference . . . 437
8.391.1 Detailed Description . . . 437
8.391.2 Field Documentation . . . 437
8.391.2.1 ECIOThresListLen . . . 437
8.391.2.2 pECIOThresList . . . 437
8.392 nas_errorRateListElement Struct Reference . . . 437
8.392.1 Detailed Description . . . 437
8.392.2 Field Documentation . . . 438
8.392.2.1 errorRate . . . 438
8.392.2.2 radioIf . . . 438
8.393 nas_GERANInfo Struct Reference . . . 438
8.393.1 Detailed Description . . . 439
8.393.2 Field Documentation . . . 440
8.393.2.1 arfcn . . . 440
8.393.2.2 bsic . . . 440
8.393.2.3 cellID . . . 440
8.393.2.4 insNmrCellInfo . . . 440
8.393.2.5 lac . . . 440
8.393.2.6 nmrInst . . . 440
8.393.2.7 plmn . . . 440
8.393.2.8 rxLev . . . 440
8.393.2.9 timingAdvance . . . 440
8.394 nas_geranInstInfo Struct Reference . . . 440
8.394.1 Detailed Description . . . 440
8.394.2 Field Documentation . . . 440
8.394.2.1 geranArfcn . . . 440
8.394.2.2 geranBsicBcc . . . 441

8.394.2.3 geranBsicNcc . . . 441
8.394.2.4 geranRssi . . . 441
8.395 nas_gsmCellInfo Struct Reference . . . 441
8.395.1 Detailed Description . . . 441
8.395.2 Field Documentation . . . 441
8.395.2.1 arfcn . . . 442
8.395.2.2 band1900 . . . 442
8.395.2.3 bsicId . . . 442
8.395.2.4 cellIdValid . . . 442
8.395.2.5 rssi . . . 442
8.395.2.6 srxlev . . . 442
8.396 nas_GSMRSSIThresh Struct Reference . . . 442
8.396.1 Detailed Description . . . 442
8.396.2 Field Documentation . . . 442
8.396.2.1 GSMRSSIThreshListLen . . . 442
8.396.2.2 pGSMRSSIThreshList . . . 442
8.397 nas_GSMSrvStatusInfo Struct Reference . . . 442
8.397.1 Detailed Description . . . 442
8.397.2 Field Documentation . . . 443
8.397.2.1 isPrefDataPath . . . 443
8.397.2.2 srvStatus . . . 443
8.397.2.3 trueSrvStatus . . . 443
8.398 nas_GSMSysInfo Struct Reference . . . 443
8.398.1 Detailed Description . . . 444
8.398.2 Field Documentation . . . 445
8.398.2.1 cellId . . . 445
8.398.2.2 cellIdValid . . . 445
8.398.2.3 dtmSupp . . . 445
8.398.2.4 dtmSuppValid . . . 446
8.398.2.5 egprsSupp . . . 446
8.398.2.6 egprsSuppValid . . . 446
8.398.2.7 lac . . . 446
8.398.2.8 lacValid . . . 446
8.398.2.9 MCC . . . 446
8.398.2.10 MNC . . . 446
8.398.2.11 networkIdValid . . . 446
8.398.2.12 regRejectInfoValid . . . 446
8.398.2.13 rejCause . . . 446
8.398.2.14 rejectSrvDomain . . . 446
8.398.2.15 sysInfoGSM . . . 446

8.399 nas_HDRECIOThresh Struct Reference . . . 446
8.399.1 Detailed Description . . . 446
8.399.2 Field Documentation . . . 446
8.399.2.1 HDRECIOThreshListLen . . . 446
8.399.2.2 pHDRECIOThreshList . . . 446
8.400 nas_HDRIOThresh Struct Reference . . . 446
8.400.1 Detailed Description . . . 447
8.400.2 Field Documentation . . . 447
8.400.2.1 HDRIOThreshListLen . . . 447
8.400.2.2 pHDRIOThreshList . . . 447
8.401 nas_HDRRSSIThresh Struct Reference . . . 447
8.401.1 Detailed Description . . . 447
8.401.2 Field Documentation . . . 447
8.401.2.1 HDRRSSIThreshListLen . . . 447
8.401.2.2 pHDRRSSIThreshList . . . 448
8.402 nas_HDRSINRThresh Struct Reference . . . 448
8.402.1 Detailed Description . . . 448
8.402.2 Field Documentation . . . 448
8.402.2.1 HDRSINRThresListLen . . . 448
8.402.2.2 pHDRSINRThresList . . . 448
8.403 nas_HDRSINRThreshold Struct Reference . . . 448
8.403.1 Detailed Description . . . 448
8.403.2 Field Documentation . . . 449
8.403.2.1 HDRSINRThreshListLen . . . 449
8.403.2.2 pHDRSINRThreshList . . . 449
8.404 nas_HDRSysInfo Struct Reference . . . 449
8.404.1 Detailed Description . . . 449
8.404.2 Field Documentation . . . 451
8.404.2.1 hdrActiveProt . . . 451
8.404.2.2 hdrActiveProtValid . . . 451
8.404.2.3 hdrPersonality . . . 451
8.404.2.4 hdrPersonalityValid . . . 451
8.404.2.5 is856SysId . . . 451
8.404.2.6 is856SysIdValid . . . 451
8.404.2.7 isSysPrlMatch . . . 451
8.404.2.8 isSysPrlMatchValid . . . 451
8.404.2.9 sysInfoHDR . . . 451
8.405 nas_homeSIDNID Struct Reference . . . 451
8.405.1 Detailed Description . . . 451
8.405.2 Field Documentation . . . 451

8.405.2.1 numInstances . . . 451
8.405.2.2 SidNid . . . 451
8.406 nas_infoInterFreq Struct Reference . . . 452
8.406.1 Detailed Description . . . 452
8.406.2 Field Documentation . . . 452
8.406.2.1 cell_resel_priority . . . 452
8.406.2.2 cellInterFreqParams . . . 452
8.406.2.3 cells_len . . . 452
8.406.2.4 earfcn . . . 453
8.406.2.5 threshXHigh . . . 453
8.406.2.6 threshXLow . . . 453
8.407 nas_IOThresh Struct Reference . . . 453
8.407.1 Detailed Description . . . 453
8.407.2 Field Documentation . . . 453
8.407.2.1 IOThresListLen . . . 453
8.407.2.2 pIOThresList . . . 453
8.408 nas_lteEARFCN Struct Reference . . . 453
8.408.1 Detailed Description . . . 453
8.408.2 Field Documentation . . . 454
8.408.2.1 earfcn0 . . . 454
8.408.2.2 earfcn1 . . . 454
8.408.2.3 status . . . 454
8.409 nas_lteGsmCellInfo Struct Reference . . . 454
8.409.1 Detailed Description . . . 454
8.409.2 Field Documentation . . . 455
8.409.2.1 cellReselPriority . . . 455
8.409.2.2 cells_len . . . 455
8.409.2.3 GsmCellInfo . . . 455
8.409.2.4 nccPermitted . . . 455
8.409.2.5 threshGsmHigh . . . 455
8.409.2.6 threshGsmLow . . . 455
8.410 nas_LTEInfo Struct Reference . . . 455
8.410.1 Detailed Description . . . 455
8.410.2 Field Documentation . . . 457
8.410.2.1 band . . . 457
8.410.2.2 bandwidth . . . 457
8.410.2.3 emmConnState . . . 457
8.410.2.4 emmState . . . 457
8.410.2.5 emmSubState . . . 457
8.410.2.6 RXChan . . . 457

8.410.2.7 TXChan . . . 457
8.411 nas_LTEInfoInterfreq Struct Reference . . . 458
8.411.1 Detailed Description . . . 458
8.411.2 Field Documentation . . . 458
8.411.2.1 freqsLen . . . 458
8.411.2.2 InfoInterfreq . . . 458
8.411.2.3 ueInIdle . . . 458
8.412 nas_LTEInfoIntrafreq Struct Reference . . . 458
8.412.1 Detailed Description . . . 459
8.412.2 Field Documentation . . . 460
8.412.2.1 CellParams . . . 460
8.412.2.2 cellReselPriority . . . 460
8.412.2.3 cellsLen . . . 460
8.412.2.4 earfcn . . . 460
8.412.2.5 globalCellId . . . 460
8.412.2.6 plmn . . . 460
8.412.2.7 servingCellId . . . 460
8.412.2.8 sIntraSearch . . . 460
8.412.2.9 sNonIntraSearch . . . 460
8.412.2.10 tac . . . 460
8.412.2.11 threshServingLow . . . 460
8.412.2.12 ueInIdle . . . 460
8.413 nas_LTEInfoNeighboringGSM Struct Reference . . . 460
8.413.1 Detailed Description . . . 461
8.413.2 Field Documentation . . . 461
8.413.2.1 freqsLen . . . 461
8.413.2.2 LteGsmCellInfo . . . 461
8.413.2.3 ueInIdle . . . 461
8.414 nas_LTEInfoNeighboringWCDMA Struct Reference . . . 461
8.414.1 Detailed Description . . . 461
8.414.2 Field Documentation . . . 462
8.414.2.1 freqsLen . . . 462
8.414.2.2 LTEWCDMACellInfo . . . 462
8.414.2.3 ueInIdle . . . 462
8.415 nas_ltePCI Struct Reference . . . 462
8.415.1 Detailed Description . . . 462
8.415.2 Field Documentation . . . 462
8.415.2.1 earfcn . . . 462
8.415.2.2 pci . . . 462
8.415.2.3 status . . . 462

8.416 nas_lteRsrpinformation Struct Reference . . . 462
8.416.1 Detailed Description . . . 462
8.416.2 Field Documentation . . . 463
8.416.2.1 rsrplevel . . . 463
8.417 nas_LTERSRPThresh Struct Reference . . . 463
8.417.1 Detailed Description . . . 463
8.417.2 Field Documentation . . . 463
8.417.2.1 LTERSRPThreshListLen . . . 463
8.417.2.2 pLTERSRPThreshList . . . 463
8.418 nas_LTERSRQThresh Struct Reference . . . 463
8.418.1 Detailed Description . . . 463
8.418.2 Field Documentation . . . 464
8.418.2.1 LTERSRQThreshListLen . . . 464
8.418.2.2 pLTERSRQThreshList . . . 464
8.419 nas_LTERSSIThresh Struct Reference . . . 464
8.419.1 Detailed Description . . . 464
8.419.2 Field Documentation . . . 464
8.419.2.1 LTERSSIThreshListLen . . . 464
8.419.2.2 pLTERSSIThreshList . . . 464
8.420 nas_LTESigRptCfg Struct Reference . . . 464
8.420.1 Detailed Description . . . 465
8.420.2 Field Documentation . . . 465
8.420.2.1 avgPeriod . . . 465
8.420.2.2 rptRate . . . 465
8.421 nas_LTESigRptConfig Struct Reference . . . 465
8.421.1 Detailed Description . . . 465
8.421.2 Field Documentation . . . 466
8.421.2.1 avgPeriod . . . 466
8.421.2.2 rptRate . . . 466
8.422 nas_lteSnrinformation Struct Reference . . . 466
8.422.1 Detailed Description . . . 466
8.422.2 Field Documentation . . . 466
8.422.2.1 snrlevel . . . 467
8.423 nas_LTESNRThresh Struct Reference . . . 467
8.423.1 Detailed Description . . . 467
8.423.2 Field Documentation . . . 467
8.423.2.1 LTESNRThresListLen . . . 467
8.423.2.2 pLTESNRThresList . . . 467
8.424 nas_LTESNRThreshold Struct Reference . . . 467
8.424.1 Detailed Description . . . 467

8.424.2 Field Documentation . . . 468
8.424.2.1 LTESNRThreshListLen . . . 468
8.424.2.2 pLTESNRThreshList . . . 468
8.425 nas_LTESysInfo Struct Reference . . . 468
8.425.1 Detailed Description . . . 468
8.425.2 Field Documentation . . . 470
8.425.2.1 cellId . . . 470
8.425.2.2 cellIdValid . . . 470
8.425.2.3 lac . . . 470
8.425.2.4 lacValid . . . 470
8.425.2.5 MCC . . . 470
8.425.2.6 MNC . . . 470
8.425.2.7 networkIdValid . . . 470
8.425.2.8 regRejectInfoValid . . . 470
8.425.2.9 rejCause . . . 470
8.425.2.10 rejectSrvDomain . . . 470
8.425.2.11 sysInfoLTE . . . 470
8.425.2.12 tac . . . 470
8.425.2.13 tacValid . . . 470
8.426 nas_lteWcdmaCellInfo Struct Reference . . . 470
8.426.1 Detailed Description . . . 471
8.426.2 Field Documentation . . . 471
8.426.2.1 cellReselPriority . . . 471
8.426.2.2 cellsLen . . . 471
8.426.2.3 threshXhigh . . . 471
8.426.2.4 threshXlow . . . 471
8.426.2.5 uarfcn . . . 471
8.426.2.6 WCDMACellInfo . . . 471
8.427 nas_minBasedIMSI Struct Reference . . . 471
8.427.1 Detailed Description . . . 472
8.427.2 Field Documentation . . . 472
8.427.2.1 imsiM1112 . . . 472
8.427.2.2 imsiMS1 . . . 472
8.427.2.3 imsiMS2 . . . 472
8.427.2.4 mccM . . . 472
8.428 nas_MNRInfo Struct Reference . . . 472
8.428.1 Detailed Description . . . 472
8.428.2 Field Documentation . . . 473
8.428.2.1 mcc . . . 473
8.428.2.2 mnc . . . 473

8.428.2.3 rat . . . 473
8.429 nas_namName Struct Reference . . . 473
8.429.1 Detailed Description . . . 473
8.429.2 Field Documentation . . . 473
8.429.2.1 namName . . . 473
8.429.2.2 namNameLen . . . 473
8.430 nas_netSelectionPref Struct Reference . . . 473
8.430.1 Detailed Description . . . 474
8.430.2 Field Documentation . . . 474
8.430.2.1 mcc . . . 474
8.430.2.2 mnc . . . 474
8.430.2.3 netReg . . . 474
8.431 nas_NetworkStat1x Struct Reference . . . 474
8.431.1 Detailed Description . . . 474
8.431.2 Field Documentation . . . 476
8.431.2.1 ActSetCnt . . . 476
8.431.2.2 NeighborSetCnt . . . 476
8.431.2.3 pActPilotPNElements . . . 476
8.431.2.4 pNeighborSetPilotPN . . . 476
8.431.2.5 RX_EC_IO . . . 476
8.431.2.6 RX_PWR . . . 476
8.431.2.7 SO . . . 476
8.431.2.8 State . . . 476
8.431.2.9 TX_PWR . . . 476
8.432 nas_NetworkStatEVDO Struct Reference . . . 476
8.432.1 Detailed Description . . . 476
8.432.2 Field Documentation . . . 477
8.432.2.1 MACIndex . . . 478
8.432.2.2 PER . . . 478
8.432.2.3 PilotEnergy . . . 478
8.432.2.4 pSectorID . . . 478
8.432.2.5 RX_PWR . . . 478
8.432.2.6 SectorIDLen . . . 478
8.432.2.7 SNR . . . 478
8.432.2.8 State . . . 478
8.433 nas_nmrCellInfo Struct Reference . . . 478
8.433.1 Detailed Description . . . 478
8.433.2 Field Documentation . . . 479
8.433.2.1 nmrArfcn . . . 479
8.433.2.2 nmrBsic . . . 479

8.433.2.3 nmrCellID . . . 479
8.433.2.4 nmrLac . . . 479
8.433.2.5 nmrPlmn . . . 479
8.433.2.6 nmrRxLev . . . 479
8.434 nas_operatorNameString Struct Reference . . . 479
8.434.1 Detailed Description . . . 479
8.434.2 Field Documentation . . . 479
8.434.2.1 PLMNName . . . 479
8.435 nas_OperatorPLMNData Struct Reference . . . 479
8.435.1 Detailed Description . . . 480
8.435.2 Field Documentation . . . 480
8.435.2.1 lac1 . . . 480
8.435.2.2 lac2 . . . 480
8.435.2.3 mcc . . . 480
8.435.2.4 mnc . . . 480
8.435.2.5 PLMNRecID . . . 480
8.436 nas_operatorPLMNList Struct Reference . . . 480
8.436.1 Detailed Description . . . 480
8.436.2 Field Documentation . . . 481
8.436.2.1 numInstance . . . 481
8.436.2.2 PLMNData . . . 481
8.437 nas_PhyCaAggPcellInfo Struct Reference . . . 481
8.437.1 Detailed Description . . . 481
8.437.2 Field Documentation . . . 481
8.437.2.1 dl_bw_value . . . 481
8.437.2.2 freq . . . 481
8.437.2.3 iLTEbandValue . . . 482
8.437.2.4 pci . . . 482
8.437.2.5 TlvPresent . . . 482
8.438 nas_PhyCaAggScellDlBw Struct Reference . . . 482
8.438.1 Detailed Description . . . 482
8.438.2 Field Documentation . . . 482
8.438.2.1 dl_bw_value . . . 482
8.438.2.2 TlvPresent . . . 482
8.439 nas_PhyCaAggScellIndex Struct Reference . . . 482
8.439.1 Detailed Description . . . 482
8.439.2 Field Documentation . . . 483
8.439.2.1 scell_idx . . . 483
8.439.2.2 TlvPresent . . . 483
8.440 nas_PhyCaAggScellIndType Struct Reference . . . 483

8.440.1 Detailed Description . . . 483
8.440.2 Field Documentation . . . 483
8.440.2.1 freq . . . 483
8.440.2.2 pci . . . 483
8.440.2.3 scell_state . . . 483
8.440.2.4 TlvPresent . . . 483
8.441 nas_PhyCaAggScellInfo Struct Reference . . . 483
8.441.1 Detailed Description . . . 484
8.441.2 Field Documentation . . . 486
8.441.2.1 dl_bw_value . . . 486
8.441.2.2 freq . . . 486
8.441.2.3 iLTEbandValue . . . 486
8.441.2.4 pci . . . 486
8.441.2.5 scell_state . . . 486
8.441.2.6 TlvPresent . . . 486
8.442 nas_PilotSetData Struct Reference . . . 486
8.442.1 Detailed Description . . . 486
8.442.2 Field Documentation . . . 486
8.442.2.1 NumPilots . . . 486
8.442.2.2 pPilotSetInfo . . . 486
8.443 nas_PilotSetParams Struct Reference . . . 486
8.443.1 Detailed Description . . . 487
8.443.2 Field Documentation . . . 487
8.443.2.1 PilotPN . . . 487
8.443.2.2 PilotStrength . . . 487
8.443.2.3 PilotType . . . 487
8.444 nas_PLMNNetworkName Struct Reference . . . 487
8.444.1 Detailed Description . . . 487
8.444.2 Field Documentation . . . 487
8.444.2.1 numInstance . . . 487
8.444.2.2 PLMNNetName . . . 487
8.445 nas_PLMNNetworkNameData Struct Reference . . . 488
8.445.1 Detailed Description . . . 488
8.445.2 Field Documentation . . . 489
8.445.2.1 codingScheme . . . 489
8.445.2.2 countryInitials . . . 489
8.445.2.3 longName . . . 489
8.445.2.4 longNameLen . . . 489
8.445.2.5 longNameSpareBits . . . 489
8.445.2.6 shortName . . . 489

8.445.2.7 shortNameLen . . . 489
8.445.2.8 shortNameSpareBits . . . 489
8.446 nas_protocolSubtypeElement Struct Reference . . . 489
8.446.1 Detailed Description . . . 489
8.446.2 Field Documentation . . . 490
8.446.2.1 AccessMac . . . 490
8.446.2.2 AuthProt . . . 490
8.446.2.3 ControlMac . . . 490
8.446.2.4 EncryptProt . . . 490
8.446.2.5 ForwardMac . . . 490
8.446.2.6 IdleState . . . 490
8.446.2.7 KeyExchange . . . 490
8.446.2.8 MultDisc . . . 491
8.446.2.9 PhysicalLayer . . . 491
8.446.2.10 ReverseMac . . . 491
8.446.2.11 SecProt . . . 491
8.446.2.12 VirtStream . . . 491
8.447 nas_qaQmi3Gpp2TimeZone Struct Reference . . . 491
8.447.1 Detailed Description . . . 491
8.447.2 Field Documentation . . . 491
8.447.2.1 daylightSavings . . . 491
8.447.2.2 leapSeconds . . . 491
8.447.2.3 localTimeOffset . . . 491
8.448 nas_QmiNas3GppNetworkInfo Struct Reference . . . 491
8.448.1 Detailed Description . . . 492
8.448.2 Field Documentation . . . 492
8.448.2.1 Desription . . . 492
8.448.2.2 Forbidden . . . 492
8.448.2.3 InUse . . . 492
8.448.2.4 MCC . . . 493
8.448.2.5 MNC . . . 493
8.448.2.6 Preferred . . . 493
8.448.2.7 Roaming . . . 493
8.449 nas_QmiNas3GppNetworkRAT Struct Reference . . . 493
8.449.1 Detailed Description . . . 493
8.449.2 Field Documentation . . . 493
8.449.2.1 MCC . . . 493
8.449.2.2 MNC . . . 493
8.449.2.3 RAT . . . 493
8.450 nas_QmisNasPcsDigit Struct Reference . . . 493

8.450.1 Detailed Description . . . 494
8.450.2 Field Documentation . . . 494
8.450.2.1 includes_pcs_digit . . . 494
8.450.2.2 MCC . . . 494
8.450.2.3 MNC . . . 494
8.451 nas_RejectReasonTlv Struct Reference . . . 494
8.451.1 Detailed Description . . . 494
8.451.2 Field Documentation . . . 494
8.451.2.1 rejectCause . . . 494
8.451.2.2 serviceDomain . . . 494
8.451.2.3 TlvPresent . . . 494
8.452 nas_RFInfoTlv Struct Reference . . . 494
8.452.1 Detailed Description . . . 495
8.452.2 Field Documentation . . . 495
8.452.2.1 activeBandClass . . . 495
8.452.2.2 activeChannel . . . 495
8.452.2.3 radioInterface . . . 495
8.452.2.4 radioInterfaceSize . . . 495
8.452.2.5 TlvPresent . . . 495
8.453 nas_roamIndList Struct Reference . . . 495
8.453.1 Detailed Description . . . 495
8.453.2 Field Documentation . . . 496
8.453.2.1 numInstances . . . 496
8.453.2.2 radioInterface . . . 496
8.453.2.3 roamIndicator . . . 496
8.454 nas_RSRPThresh Struct Reference . . . 496
8.454.1 Detailed Description . . . 496
8.454.2 Field Documentation . . . 497
8.454.2.1 pRSRPThresList . . . 497
8.454.2.2 RSRPThresListLen . . . 497
8.455 nas_rsrqInformation Struct Reference . . . 497
8.455.1 Detailed Description . . . 497
8.455.2 Field Documentation . . . 497
8.455.2.1 radioIf . . . 497
8.455.2.2 rsrq . . . 497
8.456 nas_RSRQThresh Struct Reference . . . 497
8.456.1 Detailed Description . . . 497
8.456.2 Field Documentation . . . 498
8.456.2.1 pRSRQThresList . . . 498
8.456.2.2 RSRQThresListLen . . . 498

8.457nas_RSSIThresh Struct Reference . . . 498
8.457.1 Detailed Description . . . 498
8.457.2 Field Documentation . . . 498
8.457.2.1 pRSSIThresList . . . 498
8.457.2.2 RSSIThresListLen . . . 499
8.458nas_rxInfo Struct Reference . . . 499
8.458.1 Detailed Description . . . 499
8.458.2 Field Documentation . . . 499
8.458.2.1 ecio . . . 499
8.458.2.2 isRadioTuned . . . 499
8.458.2.3 phase . . . 499
8.458.2.4 rscp . . . 499
8.458.2.5 rsrp . . . 500
8.458.2.6 rxPower . . . 500
8.459nas_RxSigInfo Struct Reference . . . 500
8.459.1 Detailed Description . . . 500
8.459.2 Field Documentation . . . 500
8.459.2.1 isRadioTuned . . . 500
8.459.2.2 rsrp . . . 500
8.459.2.3 rxChainIndex . . . 501
8.459.2.4 rxPower . . . 501
8.460nas_rxSignalStrengthListElement Struct Reference . . . 501
8.460.1 Detailed Description . . . 501
8.460.2 Field Documentation . . . 501
8.460.2.1 radioIf . . . 501
8.460.2.2 rxSignalStrength . . . 501
8.461nas_SccRxInfo Struct Reference . . . 501
8.461.1 Detailed Description . . . 502
8.461.2 Field Documentation . . . 502
8.461.2.1 numInstances . . . 502
8.461.2.2 rsrq . . . 502
8.461.2.3 sigInfo . . . 502
8.461.2.4 snr . . . 502
8.461.2.5 TlvPresent . . . 502
8.462nas_serviceProviderName Struct Reference . . . 502
8.462.1 Detailed Description . . . 502
8.462.2 Field Documentation . . . 503
8.462.2.1 displayCondition . . . 503
8.462.2.2 spn . . . 503
8.462.2.3 spnLength . . . 503

8.463 nas_servSystem Struct Reference . . . 503
8.463.1 Detailed Description . . . 503
8.463.2 Field Documentation . . . 504
8.463.2.1 csAttachState . . . 504
8.463.2.2 numRadioInterfaces . . . 504
8.463.2.3 psAttachState . . . 504
8.463.2.4 radioInterface . . . 504
8.463.2.5 regState . . . 504
8.463.2.6 selNetwork . . . 504
8.464 nas_sidNid Struct Reference . . . 505
8.464.1 Detailed Description . . . 505
8.464.2 Field Documentation . . . 505
8.464.2.1 nid . . . 505
8.464.2.2 sid . . . 505
8.465 nas_SignalStrengthTlv Struct Reference . . . 505
8.465.1 Detailed Description . . . 505
8.465.2 Field Documentation . . . 505
8.465.2.1 radioInterface . . . 505
8.465.2.2 signalStrength . . . 505
8.465.2.3 TlvPresent . . . 505
8.466 nas_SLQSSignalStrengthsIndReq Struct Reference . . . 505
8.466.1 Detailed Description . . . 506
8.466.2 Field Documentation . . . 507
8.466.2.1 ecioDelta . . . 507
8.466.2.2 ecioThresholdList . . . 507
8.466.2.3 ecioThresholdListLen . . . 507
8.466.2.4 ioDelta . . . 507
8.466.2.5 lteRsrpDelta . . . 507
8.466.2.6 lteSnrDelta . . . 507
8.466.2.7 rsrqDelta . . . 507
8.466.2.8 rxSignalStrengthDelta . . . 507
8.466.2.9 sinrDelta . . . 507
8.466.2.10 sinrThresholdList . . . 507
8.466.2.11 sinrThresholdListLen . . . 507
8.467 nas_SLQSSignalStrengthsInformation Struct Reference . . . 507
8.467.1 Detailed Description . . . 508
8.467.2 Field Documentation . . . 508
8.467.2.1 ecioInfo . . . 508
8.467.2.2 errorRateInfo . . . 508
8.467.2.3 io . . . 508

8.467.2.4 lteRsrpinfo . . . 508
8.467.2.5 lteSnrinfo . . . 508
8.467.2.6 rsrqInfo . . . 508
8.467.2.7 rxSignalStrengthInfo . . . 508
8.467.2.8 sinr . . . 508
8.468 nas_SLQSSignalStrengthsTlv Struct Reference . . . 508
8.468.1 Detailed Description . . . 508
8.468.2 Field Documentation . . . 508
8.468.2.1 sSLQSSignalStrengthsInfo . . . 508
8.468.2.2 TlvPresent . . . 508
8.469 nas_SrvStatusInfo Struct Reference . . . 509
8.469.1 Detailed Description . . . 509
8.469.2 Field Documentation . . . 509
8.469.2.1 isPrefDataPath . . . 509
8.469.2.2 srvStatus . . . 509
8.470 nas_sysInfoCommon Struct Reference . . . 509
8.470.1 Detailed Description . . . 509
8.470.2 Field Documentation . . . 511
8.470.2.1 isSysForbidden . . . 511
8.470.2.2 isSysForbiddenValid . . . 511
8.470.2.3 roamStatus . . . 511
8.470.2.4 roamStatusValid . . . 511
8.470.2.5 srvCapability . . . 511
8.470.2.6 srvCapabilityValid . . . 511
8.470.2.7 srvDomain . . . 511
8.470.2.8 srvDomainValid . . . 511
8.471 nas_TDSCDMAECIOThresh Struct Reference . . . 512
8.471.1 Detailed Description . . . 512
8.471.2 Field Documentation . . . 512
8.471.2.1 pTDSCDMAECIOThreshList . . . 512
8.471.2.2 TDSCDMAECIOThreshListLen . . . 512
8.472 nas_TDSCDMARSCPThresh Struct Reference . . . 512
8.472.1 Detailed Description . . . 512
8.472.2 Field Documentation . . . 512
8.472.2.1 pTDSCDMARSCPThreshList . . . 512
8.472.2.2 TDSCDMARSCPThreshListLen . . . 513
8.473 nas_TDSCDMARSSIThresh Struct Reference . . . 513
8.473.1 Detailed Description . . . 513
8.473.2 Field Documentation . . . 513
8.473.2.1 pTDSCDMARSSIThreshList . . . 513

8.473.2.2 TDSCDMARSSIThreshListLen . . . 513
8.474 nas_TDSCDMASINRCONFThresh Struct Reference . . . 513
8.474.1 Detailed Description . . . 513
8.474.2 Field Documentation . . . 513
8.474.2.1 pTDSCDMASINRCONFThreshList . . . 513
8.474.2.2 TDSCDMASINRCONFThreshListLen . . . 514
8.475 nas_TDSCDMASINRThresh Struct Reference . . . 514
8.475.1 Detailed Description . . . 514
8.475.2 Field Documentation . . . 514
8.475.2.1 pTDSCDMASINRThreshList . . . 514
8.475.2.2 TDSCDMASINRThreshListLen . . . 514
8.476 nas_timeInfo Struct Reference . . . 514
8.476.1 Detailed Description . . . 514
8.476.2 Field Documentation . . . 516
8.476.2.1 day . . . 516
8.476.2.2 dayLtSavingAdj . . . 516
8.476.2.3 dayOfWeek . . . 516
8.476.2.4 hour . . . 516
8.476.2.5 minute . . . 516
8.476.2.6 month . . . 516
8.476.2.7 radioInterface . . . 516
8.476.2.8 second . . . 516
8.476.2.9 timeZone . . . 516
8.476.2.10 TlvPresent . . . 516
8.476.2.11 year . . . 516
8.477 nas_trueIMSI Struct Reference . . . 516
8.477.1 Detailed Description . . . 516
8.477.2 Field Documentation . . . 517
8.477.2.1 imsiT1112 . . . 517
8.477.2.2 imsiTaddrNum . . . 517
8.477.2.3 imsiTS1 . . . 517
8.477.2.4 imsiTS2 . . . 517
8.477.2.5 mccT . . . 517
8.478 nas_txInfo Struct Reference . . . 517
8.478.1 Detailed Description . . . 517
8.478.2 Field Documentation . . . 517
8.478.2.1 isInTraffic . . . 517
8.478.2.2 txPower . . . 517
8.479 nas_UMTSInfo Struct Reference . . . 517
8.479.1 Detailed Description . . . 518

8.479.2 Field Documentation . . . 518
8.479.2.1 cellID . . . 519
8.479.2.2 ecio . . . 519
8.479.2.3 geranInst . . . 519
8.479.2.4 GeranInstInfo . . . 519
8.479.2.5 lac . . . 519
8.479.2.6 plmn . . . 519
8.479.2.7 psc . . . 519
8.479.2.8 rscp . . . 519
8.479.2.9 uarfcn . . . 519
8.479.2.10umtsInst . . . 519
8.479.2.11UMTSInstInfo . . . 519
8.480nas_UMTSinstInfo Struct Reference . . . 519
8.480.1 Detailed Description . . . 519
8.480.2 Field Documentation . . . 519
8.480.2.1 umtsEcio . . . 519
8.480.2.2 umtsPsc . . . 520
8.480.2.3 umtsRscp . . . 520
8.480.2.4 umtsUarfcn . . . 520
8.481nas_umtsLTENbrCell Struct Reference . . . 520
8.481.1 Detailed Description . . . 520
8.481.2 Field Documentation . . . 520
8.481.2.1 cellIsTDD . . . 520
8.481.2.2 earfcn . . . 520
8.481.2.3 pci . . . 520
8.481.2.4 rsrp . . . 521
8.481.2.5 rsrq . . . 521
8.481.2.6 srxlev . . . 521
8.482nas_UniversalTime Struct Reference . . . 521
8.482.1 Detailed Description . . . 521
8.482.2 Field Documentation . . . 522
8.482.2.1 day . . . 522
8.482.2.2 dayOfWeek . . . 522
8.482.2.3 hour . . . 522
8.482.2.4 minute . . . 522
8.482.2.5 month . . . 522
8.482.2.6 second . . . 522
8.482.2.7 year . . . 522
8.483nas_wcdmaCellInfo Struct Reference . . . 522
8.483.1 Detailed Description . . . 522

8.483.2 Field Documentation . . . 522
8.483.2.1 cpich_ecno . . . 522
8.483.2.2 cpich_rscp . . . 523
8.483.2.3 psc . . . 523
8.483.2.4 srxlev . . . 523
8.484 nas_WCDMAECIOThresh Struct Reference . . . 523
8.484.1 Detailed Description . . . 523
8.484.2 Field Documentation . . . 523
8.484.2.1 pWCDMAECIOThreshList . . . 523
8.484.2.2 WCDMAECIOThreshListLen . . . 523
8.485 nas_WCDMAInfoLTENeighborCell Struct Reference . . . 523
8.485.1 Detailed Description . . . 523
8.485.2 Field Documentation . . . 524
8.485.2.1 UMTSLTENbrCell . . . 524
8.485.2.2 umtsLTENbrCellLen . . . 524
8.485.2.3 wcdmaRRCState . . . 524
8.486 nas_WCDMARSSIThresh Struct Reference . . . 524
8.486.1 Detailed Description . . . 524
8.486.2 Field Documentation . . . 525
8.486.2.1 pWCDMARSSIThreshList . . . 525
8.486.2.2 WCDMARSSIThreshListLen . . . 525
8.487 nas_WCDMASysInfo Struct Reference . . . 525
8.487.1 Detailed Description . . . 525
8.487.2 Field Documentation . . . 527
8.487.2.1 cellId . . . 527
8.487.2.2 cellIdValid . . . 527
8.487.2.3 hsCallStatus . . . 527
8.487.2.4 hsCallStatusValid . . . 528
8.487.2.5 hsInd . . . 528
8.487.2.6 hsIndValid . . . 528
8.487.2.7 lac . . . 528
8.487.2.8 lacValid . . . 528
8.487.2.9 MCC . . . 528
8.487.2.10 MNC . . . 528
8.487.2.11 networkIdValid . . . 528
8.487.2.12 psc . . . 528
8.487.2.13 pscValid . . . 528
8.487.2.14 regRejectInfoValid . . . 528
8.487.2.15 rejCause . . . 528
8.487.2.16 rejectSrvDomain . . . 528

8.487.2.17 sysInfoWCDMA . . . 528
8.488 nas_wcdmaUARFCN Struct Reference . . . 528
8.488.1 Detailed Description . . . 528
8.488.2 Field Documentation . . . 528
8.488.2.1 status . . . 528
8.488.2.2 uarfcn . . . 528
8.489 NASBandPreferenceTlv Struct Reference . . . 529
8.489.1 Detailed Description . . . 529
8.489.2 Field Documentation . . . 529
8.489.2.1 band_pref . . . 529
8.489.2.2 TlvPresent . . . 529
8.490 nasCellLocationInfoResp Struct Reference . . . 529
8.490.1 Detailed Description . . . 529
8.490.2 Field Documentation . . . 530
8.490.2.1 pCDMAInfo . . . 530
8.490.2.2 pGERANInfo . . . 530
8.490.2.3 pLTEInfoInterfreq . . . 530
8.490.2.4 pLTEInfoIntrafreq . . . 530
8.490.2.5 pLTEInfoNeighboringGSM . . . 530
8.490.2.6 pLTEInfoNeighboringWCDMA . . . 530
8.490.2.7 pUMTSCellID . . . 530
8.490.2.8 pUMTSInfo . . . 530
8.490.2.9 pWCDMAInfoLTENeighborCell . . . 530
8.491 NASEmergencyModeTlv Struct Reference . . . 530
8.491.1 Detailed Description . . . 530
8.491.2 Field Documentation . . . 531
8.491.2.1 EmerMode . . . 531
8.491.2.2 TlvPresent . . . 531
8.492 nasGet3GPP2SubscriptionInfoReq Struct Reference . . . 531
8.492.1 Detailed Description . . . 531
8.492.2 Field Documentation . . . 531
8.492.2.1 namID . . . 531
8.493 nasGet3GPP2SubscriptionInfoResp Struct Reference . . . 531
8.493.1 Detailed Description . . . 531
8.493.2 Field Documentation . . . 532
8.493.2.1 pCDMAChannel . . . 532
8.493.2.2 pDirNum . . . 532
8.493.2.3 pHomeSIDNID . . . 532
8.493.2.4 pMinBasedIMSI . . . 532
8.493.2.5 pNAMNameInfo . . . 532

8.493.2.6 pTrueIMSI . . . 532
8.494 nasGetHDRColorCodeResp Struct Reference . . . 532
8.494.1 Detailed Description . . . 532
8.494.2 Field Documentation . . . 532
8.494.2.1 pColorCode . . . 532
8.495 nasGetLTECphyCa Struct Reference . . . 532
8.495.1 Field Documentation . . . 533
8.495.1.1 sPhyCaAggPcellInfo . . . 533
8.495.1.2 sPhyCaAggScellDlBw . . . 533
8.495.1.3 sPhyCaAggScellIndex . . . 533
8.495.1.4 sPhyCaAggScellIndType . . . 533
8.495.1.5 sPhyCaAggScellInfo . . . 533
8.496 NasGetLTECphyCaInfo Struct Reference . . . 533
8.496.1 Field Documentation . . . 533
8.496.1.1 PhyCaAggPcellInfo . . . 533
8.496.1.2 PhyCaAggScellDlBw . . . 533
8.496.1.3 PhyCaAggScellIndex . . . 533
8.496.1.4 PhyCaAggScellIndType . . . 533
8.496.1.5 PhyCaAggScellInfo . . . 533
8.497 nasGetLTECphyCaResp Struct Reference . . . 533
8.497.1 Field Documentation . . . 534
8.497.1.1 pPhyCaAggPcellInfo . . . 534
8.497.1.2 pPhyCaAggScellDlBw . . . 534
8.497.1.3 pPhyCaAggScellIndex . . . 534
8.497.1.4 pPhyCaAggScellIndType . . . 534
8.497.1.5 pPhyCaAggScellInfo . . . 534
8.498 nasGetSigInfoResp Struct Reference . . . 534
8.498.1 Detailed Description . . . 534
8.498.2 Field Documentation . . . 535
8.498.2.1 pCDMASSInfo . . . 535
8.498.2.2 pGSMSSInfo . . . 535
8.498.2.3 pHDRSSInfo . . . 535
8.498.2.4 pLTESSInfo . . . 535
8.498.2.5 pTDSCDMASigInfoExt . . . 535
8.498.2.6 pTDSCDMASigInfoRscp . . . 535
8.498.2.7 pWCDMASSInfo . . . 535
8.499 nasGetSysInfoResp Struct Reference . . . 535
8.499.1 Detailed Description . . . 535
8.499.2 Field Documentation . . . 537
8.499.2.1 pAddCDMASysInfo . . . 537

8.499.2.2 pAddGSMSysInfo . . . 537
8.499.2.3 pAddHDRSysInfo . . . 537
8.499.2.4 pAddLTESysInfo . . . 537
8.499.2.5 pAddWCDMASysInfo . . . 537
8.499.2.6 pCDMASrvStatusInfo . . . 537
8.499.2.7 pCDMASysInfo . . . 537
8.499.2.8 pGSMCallBarringSysInfo . . . 537
8.499.2.9 pGSMCipherDomainSysInfo . . . 537
8.499.2.10 pGSMSrvStatusInfo . . . 537
8.499.2.11 pGSMSysInfo . . . 537
8.499.2.12 pHDRSrvStatusInfo . . . 537
8.499.2.13 pHDRSysInfo . . . 537
8.499.2.14 pLTESrvStatusInfo . . . 537
8.499.2.15 pLTESysInfo . . . 537
8.499.2.16 pLTEVoiceSupportSysInfo . . . 538
8.499.2.17 pWCDMACallBarringSysInfo . . . 538
8.499.2.18 pWCDMACipherDomainSysInfo . . . 538
8.499.2.19 pWCDMASrvStatusInfo . . . 538
8.499.2.20 pWCDMASysInfo . . . 538
8.500 nasGetTxRxInfoReq Struct Reference . . . 538
8.500.1 Detailed Description . . . 538
8.500.2 Field Documentation . . . 538
8.500.2.1 radio_if . . . 538
8.501 nasGetTxRxInfoResp Struct Reference . . . 538
8.501.1 Detailed Description . . . 538
8.501.2 Field Documentation . . . 539
8.501.2.1 pRXChain0Info . . . 539
8.501.2.2 pRXChain1Info . . . 539
8.501.2.3 pTXInfo . . . 539
8.502 NASGWAcqOrderPrefTlv Struct Reference . . . 539
8.502.1 Detailed Description . . . 539
8.502.2 Field Documentation . . . 539
8.502.2.1 GWAcqOrderPref . . . 539
8.502.2.2 TlvPresent . . . 539
8.503 nasIndicationRegisterReq Struct Reference . . . 539
8.503.1 Detailed Description . . . 540
8.503.2 Field Documentation . . . 542
8.503.2.1 pDDTMInd . . . 542
8.503.2.2 pDualStandByPrefInd . . . 542
8.503.2.3 pErrorRateInd . . . 542

8.503.2.4 pHDRNewUATIAssInd . . . 542
8.503.2.5 pHDRSessionCloseInd . . . 542
8.503.2.6 pLTECphyCa . . . 542
8.503.2.7 pManagedRoamingInd . . . 542
8.503.2.8 pNetworkRejectInd . . . 542
8.503.2.9 pNetworkTimeInd . . . 542
8.503.2.10 pServingSystemInd . . . 542
8.503.2.11 pSignalStrengthInd . . . 542
8.503.2.12 pSubscriptionInfoInd . . . 542
8.503.2.13 pSuppressSysInfoInd . . . 542
8.503.2.14 pSysInfoInd . . . 542
8.503.2.15 pSystemSelectionInd . . . 542
8.504 nasInitNetworkReg Struct Reference . . . 542
8.504.1 Detailed Description . . . 543
8.504.2 Field Documentation . . . 543
8.504.2.1 pChangeDuration . . . 543
8.504.2.2 pMncPcsDigitStatus . . . 543
8.504.2.3 pMNRInfo . . . 543
8.504.2.4 regAction . . . 543
8.505 NASLTEBandPreferenceTlv Struct Reference . . . 543
8.505.1 Detailed Description . . . 544
8.505.2 Field Documentation . . . 544
8.505.2.1 LTEBandPref . . . 544
8.505.2.2 TlvPresent . . . 544
8.506 NASLteNasReleaseInfoTlv Struct Reference . . . 544
8.506.1 Detailed Description . . . 544
8.506.2 Field Documentation . . . 544
8.506.2.1 nas_major . . . 544
8.506.2.2 nas_minor . . . 544
8.506.2.3 nas_release . . . 544
8.506.2.4 TlvPresent . . . 544
8.507 NASModePreferenceTlv Struct Reference . . . 545
8.507.1 Detailed Description . . . 545
8.507.2 Field Documentation . . . 545
8.507.2.1 ModePref . . . 545
8.507.2.2 TlvPresent . . . 545
8.508 NASNetSelPreferenceTlv Struct Reference . . . 545
8.508.1 Detailed Description . . . 545
8.508.2 Field Documentation . . . 545
8.508.2.1 NetSelPref . . . 545

8.508.2.2 TlvPresent . . . 545
8.509 nasNetworkReject Struct Reference . . . 545
8.509.1 Detailed Description . . . 546
8.509.2 Field Documentation . . . 548
8.509.2.1 pCsgId . . . 548
8.509.2.2 pPlmnId . . . 548
8.509.2.3 radioIf . . . 548
8.509.2.4 rejectCause . . . 548
8.509.2.5 serviceDomain . . . 548
8.510 nasNetworkTime Struct Reference . . . 548
8.510.1 Detailed Description . . . 548
8.510.2 Field Documentation . . . 549
8.510.2.1 pDayltSavAdj . . . 549
8.510.2.2 pRadioInterface . . . 549
8.510.2.3 pTimeZone . . . 549
8.510.2.4 universalTime . . . 549
8.511 nasOperatorNameResp Struct Reference . . . 549
8.511.1 Detailed Description . . . 549
8.511.2 Field Documentation . . . 550
8.511.2.1 pNITZInformation . . . 550
8.511.2.2 pOperatorNameString . . . 550
8.511.2.3 pOperatorPLMNList . . . 550
8.511.2.4 pPLMNNetworkName . . . 550
8.511.2.5 pSrvcProviderName . . . 550
8.512 NASOTAMessageTlv Struct Reference . . . 550
8.512.1 Detailed Description . . . 550
8.512.2 Field Documentation . . . 551
8.512.2.1 data_buf . . . 551
8.512.2.2 data_len . . . 551
8.512.2.3 message_type . . . 551
8.512.2.4 TlvPresent . . . 551
8.513 NASPhyCaAggPcellInfo Struct Reference . . . 551
8.513.1 Detailed Description . . . 551
8.513.2 Field Documentation . . . 551
8.513.2.1 dl_bw_value . . . 551
8.513.2.2 freq . . . 551
8.513.2.3 iLTEbandValue . . . 551
8.513.2.4 pci . . . 552
8.513.2.5 TlvPresent . . . 552
8.514 NASPhyCaAggScellDlBw Struct Reference . . . 552

8.514.1 Detailed Description . . . 552
8.514.2 Field Documentation . . . 552
8.514.2.1 dl_bw_value . . . 552
8.514.2.2 TlvPresent . . . 552
8.515NASPhyCaAggScellIndex Struct Reference . . . 552
8.515.1 Detailed Description . . . 552
8.515.2 Field Documentation . . . 552
8.515.2.1 scell_idx . . . 553
8.515.2.2 TlvPresent . . . 553
8.516NASPhyCaAggScellIndType Struct Reference . . . 553
8.516.1 Detailed Description . . . 553
8.516.2 Field Documentation . . . 553
8.516.2.1 freq . . . 553
8.516.2.2 pci . . . 553
8.516.2.3 scell_state . . . 553
8.516.2.4 TlvPresent . . . 553
8.517NASPhyCaAggScellInfo Struct Reference . . . 553
8.517.1 Detailed Description . . . 554
8.517.2 Field Documentation . . . 554
8.517.2.1 dl_bw_value . . . 554
8.517.2.2 freq . . . 554
8.517.2.3 iLTEbandValue . . . 554
8.517.2.4 pci . . . 554
8.517.2.5 scell_state . . . 554
8.517.2.6 TlvPresent . . . 554
8.518nasPLMNNameReq Struct Reference . . . 554
8.518.1 Detailed Description . . . 555
8.518.2 Field Documentation . . . 555
8.518.2.1 mcc . . . 555
8.518.2.2 mnc . . . 555
8.518.2.3 pMncPcsStatus . . . 555
8.519nasPLMNNameResp Struct Reference . . . 555
8.519.1 Detailed Description . . . 556
8.519.2 Field Documentation . . . 557
8.519.2.1 longName . . . 557
8.519.2.2 longNameCI . . . 557
8.519.2.3 longNameEn . . . 557
8.519.2.4 longNameLen . . . 557
8.519.2.5 longNameSB . . . 557
8.519.2.6 shortName . . . 557

8.519.2.7 shortNameCI . . . 557
8.519.2.8 shortNameEn . . . 557
8.519.2.9 shortNameLen . . . 557
8.519.2.10 shortNameSB . . . 557
8.519.2.11 spn . . . 557
8.519.2.12 spnEncoding . . . 558
8.519.2.13 spnLength . . . 558
8.520 NASPRLPreferenceTlv Struct Reference . . . 558
8.520.1 Detailed Description . . . 558
8.520.2 Field Documentation . . . 558
8.520.2.1 PRLPref . . . 558
8.520.2.2 TlvPresent . . . 558
8.521 NASQmiCbkNasSwiOTAMessageInd Struct Reference . . . 558
8.521.1 Detailed Description . . . 558
8.521.2 Field Documentation . . . 558
8.521.2.1 nasRelInfoTlv . . . 559
8.521.2.2 otaMsgTlv . . . 559
8.521.2.3 timeTlv . . . 559
8.522 NASQmiCbkNasSystemSelPrefInd Struct Reference . . . 559
8.522.1 Detailed Description . . . 559
8.522.2 Field Documentation . . . 560
8.522.2.1 BPTlv . . . 560
8.522.2.2 EMTlv . . . 560
8.522.2.3 GWAOPTlv . . . 560
8.522.2.4 LBPTlv . . . 560
8.522.2.5 MPTlv . . . 560
8.522.2.6 NSPTlv . . . 560
8.522.2.7 PRLPTlv . . . 560
8.522.2.8 RPTlv . . . 560
8.522.2.9 SDPTlv . . . 560
8.523 NASRoamPreferenceTlv Struct Reference . . . 560
8.523.1 Detailed Description . . . 560
8.523.2 Field Documentation . . . 560
8.523.2.1 RoamPref . . . 560
8.523.2.2 TlvPresent . . . 560
8.524 NASServDomainPrefTlv Struct Reference . . . 560
8.524.1 Detailed Description . . . 561
8.524.2 Field Documentation . . . 561
8.524.2.1 SrvDomainPref . . . 561
8.524.2.2 TlvPresent . . . 561

8.525 NASServingSystemInfo Struct Reference . . . 561
8.525.1 Detailed Description . . . 561
8.525.2 Field Documentation . . . 562
8.525.2.1 csAttachState . . . 562
8.525.2.2 hdrPersonality . . . 562
8.525.2.3 psAttachState . . . 562
8.525.2.4 radioInterfaceList . . . 562
8.525.2.5 radioInterfaceNo . . . 562
8.525.2.6 registrationState . . . 562
8.525.2.7 selectedNetwork . . . 562
8.526 nasSigInfo Struct Reference . . . 562
8.526.1 Detailed Description . . . 563
8.526.2 Field Documentation . . . 563
8.526.2.1 pCDMASigInfo . . . 563
8.526.2.2 pGSMSigInfo . . . 563
8.526.2.3 pHDRSigInfo . . . 563
8.526.2.4 pLTESigInfo . . . 563
8.526.2.5 pRscp . . . 563
8.526.2.6 pTDSCDMASigInfoExt . . . 563
8.526.2.7 pWCDMASigInfo . . . 563
8.527 nasSwiGetChannelLockResp Struct Reference . . . 563
8.527.1 Detailed Description . . . 564
8.527.2 Field Documentation . . . 564
8.527.2.1 pLteEARFCN . . . 564
8.527.2.2 pLtePCI . . . 564
8.527.2.3 pWcdmaUARFCN . . . 564
8.528 NasSwiIndReg Struct Reference . . . 564
8.528.1 Detailed Description . . . 564
8.528.2 Field Documentation . . . 565
8.528.2.1 gsmUmtsDl . . . 565
8.528.2.2 gsmUmtsUl . . . 565
8.528.2.3 lteEmmDl . . . 565
8.528.2.4 lteEmmUl . . . 565
8.528.2.5 lteEsmDl . . . 565
8.528.2.6 lteEsmUl . . . 565
8.528.2.7 pRankIndicatorInd . . . 565
8.528.2.8 pTimer . . . 565
8.529 nasSwiSetChannelLockReq Struct Reference . . . 565
8.529.1 Detailed Description . . . 565
8.529.2 Field Documentation . . . 566

8.529.2.1 pLteEARFCN . . . 566
8.529.2.2 pLtePCI . . . 566
8.529.2.3 pWcdmaUARFCN . . . 566
8.530 nasSysInfo Struct Reference . . . 566
8.530.1 Detailed Description . . . 566
8.530.2 Field Documentation . . . 568
8.530.2.1 pAddCDMASysInfo . . . 568
8.530.2.2 pAddGSMSysInfo . . . 568
8.530.2.3 pAddHDRSysInfo . . . 568
8.530.2.4 pAddLTESysInfo . . . 568
8.530.2.5 pAddWCDMASysInfo . . . 568
8.530.2.6 pCDMASrvStatusInfo . . . 568
8.530.2.7 pCDMASysInfo . . . 568
8.530.2.8 pGSMCallBarringSysInfo . . . 568
8.530.2.9 pGSMCipherDomainSysInfo . . . 568
8.530.2.10 pGSMSrvStatusInfo . . . 568
8.530.2.11 pGSMSysInfo . . . 568
8.530.2.12 pHDRSrvStatusInfo . . . 569
8.530.2.13 pHDRSysInfo . . . 569
8.530.2.14 pLTESrvStatusInfo . . . 569
8.530.2.15 pLTESysInfo . . . 569
8.530.2.16 pLTEVoiceSupportSysInfo . . . 569
8.530.2.17 pSysInfoNoChange . . . 569
8.530.2.18 pWCDMACallBarringSysInfo . . . 569
8.530.2.19 pWCDMACipherDomainSysInfo . . . 569
8.530.2.20 pWCDMASrvStatusInfo . . . 569
8.530.2.21 pWCDMASysInfo . . . 569
8.531 NASTimeInfoTlv Struct Reference . . . 569
8.531.1 Detailed Description . . . 569
8.531.2 Field Documentation . . . 569
8.531.2.1 time . . . 569
8.531.2.2 TlvPresent . . . 569
8.532 nasTimers Struct Reference . . . 569
8.532.1 Detailed Description . . . 570
8.532.2 Field Documentation . . . 570
8.532.2.1 t3396_apn . . . 570
8.532.2.2 t3396_plmn_id . . . 570
8.532.2.3 t3396_val . . . 570
8.533 netSelectionPref Struct Reference . . . 570
8.533.1 Detailed Description . . . 570

8.533.2 Field Documentation . . . 570
8.533.2.1 mcc . . . 571
8.533.2.2 mnc . . . 571
8.533.2.3 netReg . . . 571
8.534 NetStats Struct Reference . . . 571
8.534.1 Detailed Description . . . 571
8.534.2 Field Documentation . . . 571
8.534.2.1 rx_bytes . . . 571
8.534.2.2 rx_errors . . . 572
8.534.2.3 rx_overflows . . . 572
8.534.2.4 rx_packets . . . 572
8.534.2.5 tx_bytes . . . 572
8.534.2.6 tx_errors . . . 572
8.534.2.7 tx_overflows . . . 572
8.534.2.8 tx_packets . . . 572
8.535 NetworkDebugResp Struct Reference . . . 572
8.535.1 Detailed Description . . . 572
8.535.2 Field Documentation . . . 572
8.535.2.1 pDataStatusDetail . . . 572
8.535.2.2 pDeviceConfigDetail . . . 572
8.535.2.3 pNetworkStat1x . . . 573
8.535.2.4 pNetworkStatEVDO . . . 573
8.535.2.5 pObjectVer . . . 573
8.536 NetworkStat1x Struct Reference . . . 573
8.536.1 Detailed Description . . . 573
8.536.2 Field Documentation . . . 574
8.536.2.1 ActSetCnt . . . 574
8.536.2.2 NeighborSetCnt . . . 574
8.536.2.3 pActPilotPNElements . . . 574
8.536.2.4 pNeighborSetPilotPN . . . 575
8.536.2.5 RX_EC_IO . . . 575
8.536.2.6 RX_PWR . . . 575
8.536.2.7 SO . . . 575
8.536.2.8 State . . . 575
8.536.2.9 TX_PWR . . . 575
8.537 NetworkStatEVDO Struct Reference . . . 575
8.537.1 Detailed Description . . . 575
8.537.2 Field Documentation . . . 576
8.537.2.1 MACIndex . . . 576
8.537.2.2 PER . . . 576

8.537.2.3 PilotEnergy . . . 576
8.537.2.4 pSectorID . . . 576
8.537.2.5 RX_PWR . . . 576
8.537.2.6 SectorIDLen . . . 576
8.537.2.7 SNR . . . 576
8.537.2.8 State . . . 576
8.538 newMTMessageTlv Struct Reference . . . 576
8.538.1 Detailed Description . . . 577
8.538.2 Field Documentation . . . 577
8.538.2.1 MTMessageInfo . . . 577
8.538.2.2 TlvPresent . . . 577
8.539 newPwdData Struct Reference . . . 577
8.539.1 Detailed Description . . . 577
8.539.2 Field Documentation . . . 577
8.539.2.1 newPwd . . . 577
8.539.2.2 newPwdAgain . . . 577
8.540 nmrCellInfo Struct Reference . . . 577
8.540.1 Detailed Description . . . 578
8.540.2 Field Documentation . . . 578
8.540.2.1 nmrArfcn . . . 578
8.540.2.2 nmrBsic . . . 578
8.540.2.3 nmrCellID . . . 579
8.540.2.4 nmrLac . . . 579
8.540.2.5 nmrPlmn . . . 579
8.540.2.6 nmrRxLev . . . 579
8.541 NSSAudioCtrl Struct Reference . . . 579
8.541.1 Detailed Description . . . 579
8.541.2 Field Documentation . . . 579
8.541.2.1 downLink . . . 579
8.541.2.2 upLink . . . 579
8.542 NWProfile Struct Reference . . . 579
8.542.1 Detailed Description . . . 579
8.542.2 Field Documentation . . . 580
8.542.2.1 pProfSz . . . 580
8.542.2.2 pProfValues . . . 580
8.542.2.3 tech . . . 580
8.543 omaDmConfigTlv Struct Reference . . . 580
8.543.1 Detailed Description . . . 580
8.543.2 Field Documentation . . . 580
8.543.2.1 alertmsg . . . 580

8.543.2.2 alertmsglength . . . 580
8.543.2.3 state . . . 580
8.543.2.4 userInputReq . . . 580
8.543.2.5 userInputTimeout . . . 581
8.544 omaDmConfigTlvExt Struct Reference . . . 581
8.544.1 Detailed Description . . . 581
8.544.2 Field Documentation . . . 583
8.544.2.1 alertmsg . . . 583
8.544.2.2 alertmsglength . . . 583
8.544.2.3 state . . . 583
8.544.2.4 userInputReq . . . 583
8.544.2.5 userInputTimeout . . . 583
8.545 omaDmFotaTlv Struct Reference . . . 583
8.545.1 Detailed Description . . . 583
8.545.2 Field Documentation . . . 584
8.545.2.1 description . . . 584
8.545.2.2 descriptionlength . . . 584
8.545.2.3 fwdloadsize . . . 585
8.545.2.4 fwloadComplete . . . 585
8.545.2.5 namelength . . . 585
8.545.2.6 package_name . . . 585
8.545.2.7 sessionType . . . 585
8.545.2.8 severity . . . 585
8.545.2.9 state . . . 585
8.545.2.10 updateCompleteStatus . . . 585
8.545.2.11 userInputReq . . . 585
8.545.2.12 userInputTimeout . . . 585
8.545.2.13 version . . . 585
8.545.2.14 versionlength . . . 585
8.546 omaDmFotaTlvExt Struct Reference . . . 585
8.546.1 Detailed Description . . . 585
8.546.2 Field Documentation . . . 587
8.546.2.1 description . . . 587
8.546.2.2 descriptionlength . . . 587
8.546.2.3 fumoResultCode . . . 587
8.546.2.4 namelength . . . 587
8.546.2.5 package_name . . . 587
8.546.2.6 packageSize . . . 587
8.546.2.7 receivedBytes . . . 587
8.546.2.8 reserved . . . 587

8.546.2.9 state . . . 587
8.546.2.10 userInputTimeout . . . 587
8.546.2.11 version . . . 587
8.546.2.12 versionlength . . . 587
8.547 omaDmNotificationsTlv Struct Reference . . . 587
8.547.1 Detailed Description . . . 587
8.547.2 Field Documentation . . . 588
8.547.2.1 notification . . . 588
8.547.2.2 sessionStatus . . . 588
8.548 operatorNameString Struct Reference . . . 588
8.548.1 Detailed Description . . . 588
8.548.2 Field Documentation . . . 588
8.548.2.1 PLMNName . . . 589
8.549 OperatorPLMNData Struct Reference . . . 589
8.549.1 Detailed Description . . . 589
8.549.2 Field Documentation . . . 589
8.549.2.1 lac1 . . . 589
8.549.2.2 lac2 . . . 589
8.549.2.3 mcc . . . 589
8.549.2.4 mnc . . . 589
8.549.2.5 PLMNRecID . . . 589
8.550 operatorPLMNList Struct Reference . . . 589
8.550.1 Detailed Description . . . 590
8.550.2 Field Documentation . . . 590
8.550.2.1 numInstance . . . 590
8.550.2.2 PLMNData . . . 590
8.551 pack_dms_ActivateAutomatic_t Struct Reference . . . 590
8.551.1 Detailed Description . . . 590
8.551.2 Field Documentation . . . 590
8.551.2.1 actCode . . . 590
8.552 pack_dms_GetCustFeaturesV2_t Struct Reference . . . 590
8.552.1 Detailed Description . . . 591
8.552.2 Field Documentation . . . 591
8.552.2.1 cust_id . . . 591
8.552.2.2 list_type . . . 591
8.552.2.3 Tlvresult . . . 591
8.553 pack_dms_ResetToFactoryDefaults_t Struct Reference . . . 591
8.553.1 Detailed Description . . . 591
8.553.2 Field Documentation . . . 591
8.553.2.1 spc . . . 591

8.554 pack_dms_SetActivationStatusCallback_t Struct Reference . . . 591
8.554.1 Detailed Description . . . 592
8.554.2 Field Documentation . . . 592
8.554.2.1 activationState . . . 592
8.555 pack_dms_SetCrashAction_t Struct Reference . . . 592
8.555.1 Detailed Description . . . 592
8.555.2 Field Documentation . . . 592
8.555.2.1 crashAction . . . 592
8.556 pack_dms_SetCustFeature_t Struct Reference . . . 592
8.556.1 Detailed Description . . . 593
8.556.2 Field Documentation . . . 594
8.556.2.1 DHCPRelayEnabled . . . 594
8.556.2.2 DisableIMSI . . . 594
8.556.2.3 GpsEnable . . . 594
8.556.2.4 GPSLPM . . . 594
8.556.2.5 GPSSel . . . 594
8.556.2.6 IPFamSupport . . . 594
8.556.2.7 IsVoiceEnabled . . . 594
8.556.2.8 RMAutoConnect . . . 594
8.556.2.9 SMSSupport . . . 594
8.557 pack_dms_SetCustFeaturesV2_t Struct Reference . . . 594
8.557.1 Detailed Description . . . 595
8.557.2 Field Documentation . . . 595
8.557.2.1 cust_id . . . 595
8.557.2.2 cust_value . . . 595
8.557.2.3 Tlvresult . . . 595
8.557.2.4 value_length . . . 595
8.558 pack_dms_SetEventReport_t Struct Reference . . . 595
8.558.1 Detailed Description . . . 595
8.558.2 Field Documentation . . . 595
8.558.2.1 mode . . . 595
8.559 pack_dms_SetPower_t Struct Reference . . . 596
8.559.1 Detailed Description . . . 596
8.559.2 Field Documentation . . . 596
8.559.2.1 mode . . . 596
8.559.2.2 Tlvresult . . . 596
8.560 pack_dms_SetUSBComp_t Struct Reference . . . 596
8.560.1 Detailed Description . . . 596
8.560.2 Field Documentation . . . 597
8.560.2.1 Tlvresult . . . 597

8.560.2.2 USBComp . . . 597
8.561 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 597
8.561.1 Detailed Description . . . 597
8.561.2 Field Documentation . . . 598
8.561.2.1 resetInfoInd . . . 598
8.562 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 598
8.562.1 Detailed Description . . . 598
8.562.2 Field Documentation . . . 598
8.562.2.1 clear . . . 598
8.563 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 598
8.563.1 Detailed Description . . . 598
8.563.2 Field Documentation . . . 598
8.563.2.1 pDestSMSContent . . . 598
8.563.2.2 pDestSMSNum . . . 599
8.564 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 599
8.564.1 Detailed Description . . . 599
8.564.2 Field Documentation . . . 599
8.564.2.1 manString . . . 599
8.564.2.2 modelString . . . 599
8.564.2.3 plasmaIDString . . . 599
8.564.2.4 swVerString . . . 599
8.565 pack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 599
8.565.1 Detailed Description . . . 600
8.565.2 Field Documentation . . . 600
8.565.2.1 nameString . . . 600
8.565.2.2 versionString . . . 600
8.566 pack_dms_UIMChangePIN_t Struct Reference . . . 600
8.566.1 Detailed Description . . . 600
8.566.2 Field Documentation . . . 600
8.566.2.1 id . . . 600
8.566.2.2 newValue . . . 600
8.566.2.3 oldValue . . . 600
8.567 pack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 601
8.567.1 Detailed Description . . . 601
8.567.2 Field Documentation . . . 601
8.567.2.1 facility . . . 601
8.568 pack_dms_UIMGetICCID_t Struct Reference . . . 601
8.568.1 Detailed Description . . . 601
8.568.2 Field Documentation . . . 601
8.568.2.1 Tlvresult . . . 601

8.569pack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 601
8.569.1 Detailed Description . . . 602
8.569.2 Field Documentation . . . 602
8.569.2.1 facility . . . 602
8.569.2.2 facilityCk . . . 602
8.569.2.3 facilityState . . . 602
8.570pack_dms_UIMSetPINProtection_t Struct Reference . . . 602
8.570.1 Detailed Description . . . 602
8.570.2 Field Documentation . . . 603
8.570.2.1 bEnable . . . 603
8.570.2.2 id . . . 603
8.570.2.3 value . . . 603
8.571pack_dms_UIMUnblockControlKey_t Struct Reference . . . 603
8.571.1 Detailed Description . . . 603
8.571.2 Field Documentation . . . 603
8.571.2.1 facility . . . 603
8.571.2.2 facilityCk . . . 603
8.572pack_dms_UIMUnblockPIN_t Struct Reference . . . 603
8.572.1 Detailed Description . . . 604
8.572.2 Field Documentation . . . 604
8.572.2.1 id . . . 604
8.572.2.2 newPin . . . 604
8.572.2.3 pukValue . . . 604
8.573pack_dms_UIMVerifyPIN_t Struct Reference . . . 604
8.573.1 Detailed Description . . . 604
8.573.2 Field Documentation . . . 604
8.573.2.1 id . . . 604
8.573.2.2 value . . . 604
8.574pack_fms_GetImagesPreference_t Struct Reference . . . 605
8.574.1 Detailed Description . . . 605
8.574.2 Field Documentation . . . 605
8.574.2.1 TlvResult . . . 605
8.575pack_fms_GetStoredImages_t Struct Reference . . . 605
8.575.1 Detailed Description . . . 605
8.575.2 Field Documentation . . . 605
8.575.2.1 TlvResult . . . 605
8.576pack_fms_SetImagesPreference_t Struct Reference . . . 605
8.576.1 Detailed Description . . . 606
8.576.2 Field Documentation . . . 606
8.576.2.1 bForceDownload . . . 606

8.576.2.2 imageListSize . . . 606
8.576.2.3 modemindex . . . 606
8.576.2.4 pImageList . . . 606
8.576.2.5 TlvResult . . . 606
8.577 pack_loc_Delete_Assist_Data_t Struct Reference . . . 606
8.577.1 Detailed Description . . . 606
8.577.2 Field Documentation . . . 607
8.577.2.1 pBdsSVInfo . . . 607
8.577.2.2 pCellDb . . . 607
8.577.2.3 pClkInfo . . . 607
8.577.2.4 pGnssData . . . 607
8.577.2.5 pSVInfo . . . 607
8.577.2.6 TlvResult . . . 607
8.578 pack_loc_EventRegister_t Struct Reference . . . 607
8.578.1 Detailed Description . . . 607
8.578.2 Field Documentation . . . 609
8.578.2.1 eventRegister . . . 609
8.578.2.2 TlvResult . . . 609
8.579 pack_loc_SetExtPowerState_t Struct Reference . . . 609
8.579.1 Detailed Description . . . 609
8.579.2 Field Documentation . . . 609
8.579.2.1 extPowerState . . . 609
8.579.2.2 TlvResult . . . 609
8.580 pack_loc_SetOperationMode_t Struct Reference . . . 609
8.580.1 Detailed Description . . . 610
8.580.2 Field Documentation . . . 610
8.580.2.1 mode . . . 610
8.580.2.2 TlvResult . . . 610
8.581 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 610
8.581.1 Detailed Description . . . 610
8.581.2 Field Documentation . . . 610
8.581.2.1 TlvResult . . . 610
8.581.2.2 xid . . . 611
8.582 pack_loc_SLQSLOCGetServer_t Struct Reference . . . 611
8.582.1 Detailed Description . . . 611
8.582.2 Field Documentation . . . 611
8.582.2.1 has_serverAddrTypeMask . . . 611
8.582.2.2 serverAddrTypeMask . . . 611
8.582.2.3 serverType . . . 611
8.583 pack_loc_SLQSLOCInjectPosition_t Struct Reference . . . 611

8.583.1 Detailed Description . . . 612
8.583.2 Field Documentation . . . 615
8.583.2.1 altitudeSrcInfo . . . 615
8.583.2.2 altitudeWrtEllipsoid . . . 615
8.583.2.3 altitudeWrtMeanSeaLevel . . . 615
8.583.2.4 has_altitudeSrcInfo . . . 615
8.583.2.5 has_altitudeWrtEllipsoid . . . 615
8.583.2.6 has_altitudeWrtMeanSeaLevel . . . 615
8.583.2.7 has_horConfidence . . . 615
8.583.2.8 has_horReliability . . . 615
8.583.2.9 has_horUncCircular . . . 615
8.583.2.10 has_latitude . . . 615
8.583.2.11 has_longitude . . . 615
8.583.2.12 has_positionSrc . . . 615
8.583.2.13 has_rawHorConfidence . . . 615
8.583.2.14 has_rawHorUncCircular . . . 615
8.583.2.15 has_timestampAge . . . 615
8.583.2.16 has_timestampUtc . . . 615
8.583.2.17 has_vertConfidence . . . 615
8.583.2.18 has_vertRelicability . . . 616
8.583.2.19 has_vertUnc . . . 616
8.583.2.20 horConfidence . . . 616
8.583.2.21 horReliability . . . 616
8.583.2.22 horUncCircular . . . 616
8.583.2.23 latitude . . . 616
8.583.2.24 longitude . . . 616
8.583.2.25 positionSrc . . . 616
8.583.2.26 rawHorConfidence . . . 616
8.583.2.27 rawHorUncCircular . . . 616
8.583.2.28 timestampAge . . . 616
8.583.2.29 timestampUtc . . . 616
8.583.2.30 vertConfidence . . . 616
8.583.2.31 vertReliability . . . 616
8.583.2.32 vertUnc . . . 616
8.584 pack_loc_SLQSLOCInjectSensorData_t Struct Reference . . . 616
8.584.1 Detailed Description . . . 616
8.584.2 Field Documentation . . . 617
8.584.2.1 acceleroData . . . 617
8.584.2.2 acceleroTimeSrc . . . 617
8.584.2.3 accelTemp . . . 617

8.584.2.4 gyroData . . . 617
8.584.2.5 gyroTemp . . . 618
8.584.2.6 gyroTimeSrc . . . 618
8.584.2.7 has_acceleroTimeSrc . . . 618
8.584.2.8 has_accelTemp . . . 618
8.584.2.9 has_accleroData . . . 618
8.584.2.10 has_gyroData . . . 618
8.584.2.11 has_gyroTemp . . . 618
8.584.2.12 has_gyroTimeSrc . . . 618
8.584.2.13 has_opaqueId . . . 618
8.584.2.14 opaqueId . . . 618
8.585 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference . . . 618
8.585.1 Detailed Description . . . 618
8.585.2 Field Documentation . . . 618
8.585.2.1 timeMsec . . . 618
8.585.2.2 timeUncMsec . . . 618
8.586 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference . . . 618
8.586.1 Detailed Description . . . 619
8.586.2 Field Documentation . . . 619
8.586.2.1 confidence . . . 619
8.586.2.2 has_confidence . . . 619
8.586.2.3 state . . . 619
8.587 pack_loc_Start_t Struct Reference . . . 619
8.587.1 Detailed Description . . . 619
8.587.2 Field Documentation . . . 620
8.587.2.1 pApplicationInfo . . . 620
8.587.2.2 pConfigAltitudeAssumed . . . 621
8.587.2.3 pHorizontalAccuracyLvl . . . 621
8.587.2.4 pIntermediateReportState . . . 621
8.587.2.5 pMinIntervalTime . . . 621
8.587.2.6 pRecurrenceType . . . 621
8.587.2.7 SessionId . . . 621
8.587.2.8 Tlvresult . . . 621
8.588 pack_loc_Stop_t Struct Reference . . . 621
8.588.1 Detailed Description . . . 621
8.588.2 Field Documentation . . . 621
8.588.2.1 SessionId . . . 621
8.588.2.2 Tlvresult . . . 621
8.589 pack_nas_InitiateDomainAttach_t Struct Reference . . . 621
8.589.1 Detailed Description . . . 621

8.589.2 Field Documentation . . . 622
8.589.2.1 action . . . 622
8.590 pack_nas_SetACCOLC_t Struct Reference . . . 622
8.590.1 Detailed Description . . . 622
8.590.2 Field Documentation . . . 622
8.590.2.1 accolc . . . 622
8.590.2.2 spc . . . 622
8.591 pack_nas_SetCDMANetworkParameters_t Struct Reference . . . 622
8.591.1 Detailed Description . . . 622
8.591.2 Field Documentation . . . 623
8.591.2.1 pApplication . . . 623
8.591.2.2 pBroadcast . . . 623
8.591.2.3 pCustomSCP . . . 624
8.591.2.4 pForceRev0 . . . 624
8.591.2.5 pProtocol . . . 624
8.591.2.6 pRoaming . . . 624
8.591.2.7 pSPC . . . 624
8.592 pack_nas_SetNetworkPreference_t Struct Reference . . . 624
8.592.1 Detailed Description . . . 624
8.592.2 Field Documentation . . . 624
8.592.2.1 Duration . . . 624
8.592.2.2 TechnologyPref . . . 624
8.592.2.3 Tlvresult . . . 625
8.593 pack_nas_SLQSConfigSigInfo_t Struct Reference . . . 625
8.593.1 Detailed Description . . . 625
8.593.2 Field Documentation . . . 626
8.593.2.1 pECIOThresh . . . 626
8.593.2.2 pHDRSINRThresh . . . 626
8.593.2.3 pIOThresh . . . 626
8.593.2.4 pLTESigRptCfg . . . 626
8.593.2.5 pLTESNRThresh . . . 626
8.593.2.6 pRSRPThresh . . . 626
8.593.2.7 pRSRQThresh . . . 626
8.593.2.8 pRSSIThresh . . . 626
8.593.2.9 pTDSCDMASINRCONFThresh . . . 626
8.594 pack_nas_SLQSGetPLMNName_t Struct Reference . . . 626
8.594.1 Detailed Description . . . 626
8.594.2 Field Documentation . . . 627
8.594.2.1 mcc . . . 627
8.594.2.2 mnc . . . 627

8.594.2.3 pMncPcsStatus . . . 627
8.595 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference . . . 627
8.595.1 Detailed Description . . . 627
8.595.2 Field Documentation . . . 627
8.595.2.1 pChangeDuration . . . 628
8.595.2.2 pMncPcsDigitStatus . . . 628
8.595.2.3 pMNRInfo . . . 628
8.595.2.4 regAction . . . 628
8.596 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . 628
8.596.1 Detailed Description . . . 629
8.596.2 Field Documentation . . . 631
8.596.2.1 pCDMAECIODelta . . . 631
8.596.2.2 pCDMAECIOThresh . . . 631
8.596.2.3 pCDMARSSIDelta . . . 631
8.596.2.4 pCDMARSSIThresh . . . 631
8.596.2.5 pGSMRSSIDelta . . . 631
8.596.2.6 pGSMRSSIThresh . . . 631
8.596.2.7 pHDRECIODelta . . . 631
8.596.2.8 pHDRECIOThresh . . . 631
8.596.2.9 pHDRIODelta . . . 631
8.596.2.10 pHDRIOThresh . . . 631
8.596.2.11 pHDRRSSIDelta . . . 631
8.596.2.12 pHDRRSSIThresh . . . 631
8.596.2.13 pHDRSINRDelta . . . 631
8.596.2.14 pHDRSINRThresh . . . 631
8.596.2.15 pLTERSRPDelta . . . 632
8.596.2.16 pLTERSRPThresh . . . 632
8.596.2.17 pLTERSRQDelta . . . 632
8.596.2.18 pLTERSRQThresh . . . 632
8.596.2.19 pLTERSSIDelta . . . 632
8.596.2.20 pLTERSSIThresh . . . 632
8.596.2.21 pLTESigRptConfig . . . 632
8.596.2.22 pLTESNRDelta . . . 632
8.596.2.23 pLTESNRThresh . . . 632
8.596.2.24 pTDSCDMAECIODelta . . . 632
8.596.2.25 pTDSCDMAECIOThresh . . . 632
8.596.2.26 pTDSCDMARSCPDelta . . . 632
8.596.2.27 pTDSCDMARSCPThresh . . . 632
8.596.2.28 pTDSCDMARSSIDelta . . . 632
8.596.2.29 pTDSCDMARSSIThresh . . . 632

8.596.2.30 pTDSCDMASINRDelta . . . 632
8.596.2.31 pTDSCDMASINRThresh . . . 632
8.596.2.32 pWCDMAECIODelta . . . 632
8.596.2.33 pWCDMAECIOThresh . . . 632
8.596.2.34 pWCDMARSSIDelta . . . 632
8.596.2.35 pWCDMARSSIThresh . . . 632
8.597 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 632
8.597.1 Detailed Description . . . 632
8.597.2 Field Documentation . . . 633
8.597.2.1 namID . . . 633
8.598 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 633
8.598.1 Detailed Description . . . 633
8.598.2 Field Documentation . . . 633
8.598.2.1 radio_if . . . 633
8.599 pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference . . . 633
8.599.1 Detailed Description . . . 634
8.599.2 Field Documentation . . . 635
8.599.2.1 pDDTMInd . . . 635
8.599.2.2 pDualStandByPrefInd . . . 636
8.599.2.3 pErrorRateInd . . . 636
8.599.2.4 pHDRNewUATIAssInd . . . 636
8.599.2.5 pHDRSessionCloseInd . . . 636
8.599.2.6 pLTECphyCa . . . 636
8.599.2.7 pManagedRoamingInd . . . 636
8.599.2.8 pNetworkRejectInd . . . 636
8.599.2.9 pNetworkTimeInd . . . 636
8.599.2.10 pServingSystemInd . . . 636
8.599.2.11 pSignalStrengthInd . . . 636
8.599.2.12 pSubscriptionInfoInd . . . 636
8.599.2.13 pSuppressSysInfoInd . . . 636
8.599.2.14 pSysInfoInd . . . 636
8.599.2.15 pSystemSelectionInd . . . 636
8.600 pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference . . . 636
8.600.1 Detailed Description . . . 636
8.600.2 Field Documentation . . . 637
8.600.2.1 gsmUmtsDl . . . 637
8.600.2.2 gsmUmtsUl . . . 637
8.600.2.3 lteEmmDl . . . 637
8.600.2.4 lteEmmUl . . . 637
8.600.2.5 lteEsmDl . . . 637

8.600.2.6 lteEsmUl . . . 637
8.600.2.7 pRankIndicatorInd . . . 637
8.600.2.8 pTimer . . . 637
8.601 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference . . . 637
8.601.1 Detailed Description . . . 638
8.601.2 Field Documentation . . . 638
8.601.2.1 pLteEARFCN . . . 638
8.601.2.2 pLtePCI . . . 638
8.601.2.3 pWcdmaUARFCN . . . 638
8.602 pack_nas_SLQSSetBandPreference_t Struct Reference . . . 638
8.602.1 Detailed Description . . . 638
8.602.2 Field Documentation . . . 639
8.602.2.1 bandPref . . . 640
8.603 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference . . . 640
8.603.1 Detailed Description . . . 640
8.603.2 Field Documentation . . . 640
8.603.2.1 bEnable . . . 640
8.603.2.2 pSigIndReq . . . 640
8.604 pack_nas_SLQSSetSysSelectionPref_t Struct Reference . . . 640
8.604.1 Detailed Description . . . 640
8.604.2 Field Documentation . . . 645
8.604.2.1 pAcqOrderPref . . . 645
8.604.2.2 pBandPref . . . 645
8.604.2.3 pChgDuration . . . 645
8.604.2.4 pCSGID . . . 645
8.604.2.5 pEmerMode . . . 645
8.604.2.6 pGWAcqOrderPref . . . 645
8.604.2.7 pLTEBandPref . . . 645
8.604.2.8 pMNCIncPCSDigStat . . . 645
8.604.2.9 pModePref . . . 645
8.604.2.10 pNetSelPref . . . 645
8.604.2.11 pPRLPref . . . 645
8.604.2.12 pRAT . . . 645
8.604.2.13 pRoamPref . . . 645
8.604.2.14 pSrvDomainPref . . . 645
8.604.2.15 pSrvRegRestriction . . . 646
8.604.2.16 pTdscdmaBandPref . . . 646
8.605 pack_nas_SLQSSwiPSDetach_t Struct Reference . . . 646
8.605.1 Detailed Description . . . 646
8.605.2 Field Documentation . . . 646

8.605.2.1 pDetachAction . . . 646
8.606 pack_pds_PDSInjectTimeReference_t Struct Reference . . . 646
8.606.1 Detailed Description . . . 646
8.606.2 Field Documentation . . . 646
8.606.2.1 systemDiscontinuities . . . 646
8.606.2.2 systemTime . . . 646
8.607 pack_pds_ResetPDSData_t Struct Reference . . . 647
8.607.1 Detailed Description . . . 647
8.607.2 Field Documentation . . . 647
8.607.2.1 pCellDataMask . . . 647
8.607.2.2 pGPSDataMask . . . 647
8.608 pack_pds_SetEventReportCallback_t Struct Reference . . . 647
8.608.1 Detailed Description . . . 648
8.608.2 Field Documentation . . . 648
8.608.2.1 posDataNmea . . . 648
8.608.2.2 rptPosData . . . 648
8.609 pack_pds_SetPDSDefaults_t Struct Reference . . . 648
8.609.1 Detailed Description . . . 648
8.609.2 Field Documentation . . . 649
8.609.2.1 accuracy . . . 649
8.609.2.2 interval . . . 649
8.609.2.3 operation . . . 649
8.609.2.4 timeout . . . 649
8.610 pack_pds_SetPDSState_t Struct Reference . . . 649
8.610.1 Detailed Description . . . 649
8.610.2 Field Documentation . . . 649
8.610.2.1 enable . . . 649
8.611 pack_pds_SetPortAutomaticTracking_t Struct Reference . . . 649
8.611.1 Detailed Description . . . 649
8.611.2 Field Documentation . . . 650
8.611.2.1 bAuto . . . 650
8.612 pack_pds_SetServiceAutomaticTracking_t Struct Reference . . . 650
8.612.1 Detailed Description . . . 650
8.612.2 Field Documentation . . . 650
8.612.2.1 bAuto . . . 650
8.613 pack_pds_SetXTRAAutomaticDownload_t Struct Reference . . . 650
8.613.1 Detailed Description . . . 650
8.613.2 Field Documentation . . . 651
8.613.2.1 bEnabled . . . 651
8.613.2.2 interval . . . 651

8.614 pack_pds_SetXTRANetwork_t Struct Reference . . . 651
8.614.1 Detailed Description . . . 651
8.614.2 Field Documentation . . . 651
8.614.2.1 preference . . . 651
8.615 pack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 651
8.615.1 Detailed Description . . . 651
8.615.2 Field Documentation . . . 652
8.615.2.1 pNetworkMode . . . 652
8.616 pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference . . . 652
8.616.1 Detailed Description . . . 652
8.616.2 Field Documentation . . . 652
8.616.2.1 forceFlag . . . 652
8.616.2.2 timeBase . . . 652
8.616.2.3 timeMsec . . . 652
8.616.2.4 timeUncMsec . . . 652
8.617 pack_pds_SLQSPDSInjectPositionData_t Struct Reference . . . 652
8.617.1 Detailed Description . . . 653
8.617.2 Field Documentation . . . 654
8.617.2.1 pAltitudeWrtEllipsoid . . . 654
8.617.2.2 pAltitudeWrtSealevel . . . 654
8.617.2.3 pHorizontalConfidence . . . 654
8.617.2.4 pHorizontalUncCircular . . . 654
8.617.2.5 pLatitude . . . 654
8.617.2.6 pLongitude . . . 654
8.617.2.7 pPositionSource . . . 654
8.617.2.8 pTimeStamp . . . 654
8.617.2.9 pTimeType . . . 654
8.617.2.10 pVerticalConfidence . . . 654
8.617.2.11 pVerticalUnc . . . 654
8.618 pack_pds_SLQSSetAGPSConfig_t Struct Reference . . . 654
8.618.1 Detailed Description . . . 655
8.618.2 Field Documentation . . . 655
8.618.2.1 pNetworkMode . . . 655
8.618.2.2 pServerAddress . . . 655
8.618.2.3 pServerPort . . . 655
8.618.2.4 pServerURL . . . 655
8.618.2.5 pServerURLLength . . . 655
8.619 pack_pds_SLQSSetPositionMethodState_t Struct Reference . . . 655
8.619.1 Detailed Description . . . 655
8.619.2 Field Documentation . . . 656

8.619.2.1 pWifiState . . . 656
8.619.2.2 pXtraDataState . . . 656
8.619.2.3 pXtraTimeState . . . 656
8.620 pack_pds_StartPDSTrackingSessionExt_t Struct Reference . . . 656
8.620.1 Detailed Description . . . 656
8.620.2 Field Documentation . . . 657
8.620.2.1 fixAccuracy . . . 657
8.620.2.2 fixCount . . . 657
8.620.2.3 fixInterval . . . 657
8.620.2.4 fixTimeout . . . 657
8.620.2.5 sessionControl . . . 657
8.620.2.6 sessionOperation . . . 657
8.620.2.7 sessionServerOption . . . 657
8.620.2.8 sessionType . . . 657
8.621 pack_qmi_t Struct Reference . . . 657
8.621.1 Detailed Description . . . 658
8.621.2 Field Documentation . . . 658
8.621.2.1 msgid . . . 658
8.621.2.2 svc . . . 658
8.621.2.3 timeout . . . 658
8.621.2.4 xid . . . 658
8.622 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 658
8.622.1 Detailed Description . . . 658
8.622.2 Field Documentation . . . 658
8.622.2.1 apnId . . . 658
8.623 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 658
8.623.1 Detailed Description . . . 658
8.623.2 Field Documentation . . . 659
8.623.2.1 apnId . . . 659
8.624 pack_qos_SLQSSetQosEventCallback_t Struct Reference . . . 659
8.624.1 Detailed Description . . . 659
8.624.2 Field Documentation . . . 659
8.624.2.1 enable . . . 659
8.625 pack_rms_SetSMSWake_t Struct Reference . . . 659
8.625.1 Detailed Description . . . 659
8.625.2 Field Documentation . . . 659
8.625.2.1 enabled . . . 660
8.625.2.2 wake_mask . . . 660
8.626 pack_sar_SLQSSetRfSarState_t Struct Reference . . . 660
8.626.1 Detailed Description . . . 660

8.626.2 Field Documentation . . . 660
8.626.2.1 RfSarState . . . 660
8.627 pack_sms_SaveSMS_t Struct Reference . . . 660
8.627.1 Detailed Description . . . 660
8.627.2 Field Documentation . . . 661
8.627.2.1 messageFormat . . . 661
8.627.2.2 messageSize . . . 661
8.627.2.3 pMessage . . . 661
8.627.2.4 storageType . . . 661
8.628 pack_sms_SendSMS_t Struct Reference . . . 661
8.628.1 Detailed Description . . . 661
8.628.2 Field Documentation . . . 662
8.628.2.1 messageFormat . . . 662
8.628.2.2 messageSize . . . 662
8.628.2.3 pLinktimer . . . 662
8.628.2.4 pMessage . . . 662
8.629 pack_sms_SetNewSMSCallback_t Struct Reference . . . 662
8.629.1 Detailed Description . . . 662
8.629.2 Field Documentation . . . 662
8.629.2.1 status . . . 662
8.630 pack_sms_SetSMSCAddress_t Struct Reference . . . 662
8.630.1 Detailed Description . . . 662
8.630.2 Field Documentation . . . 662
8.630.2.1 pSMSCAddress . . . 662
8.630.2.2 pSMSCType . . . 662
8.631 pack_sms_SLQSDeleteSMS_t Struct Reference . . . 663
8.631.1 Detailed Description . . . 663
8.631.2 Field Documentation . . . 663
8.631.2.1 pMessageIndex . . . 663
8.631.2.2 pMessageMode . . . 663
8.631.2.3 pMessageTag . . . 663
8.631.2.4 storageType . . . 663
8.632 pack_sms_SLQSGetSMS_t Struct Reference . . . 663
8.632.1 Detailed Description . . . 664
8.632.2 Field Documentation . . . 664
8.632.2.1 messageIndex . . . 664
8.632.2.2 pMessageMode . . . 664
8.632.2.3 storageType . . . 664
8.633 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 664
8.633.1 Detailed Description . . . 664

8.633.2 Field Documentation . . . 664
8.633.2.1 mode . . . 664
8.634 pack_sms_SLQSGetSMSList_t Struct Reference . . . 664
8.634.1 Detailed Description . . . 665
8.634.2 Field Documentation . . . 665
8.634.2.1 pMessageMode . . . 665
8.634.2.2 pRequestedTag . . . 665
8.634.2.3 storageType . . . 665
8.635 pack_sms_SLQSModifySMSStatus_t Struct Reference . . . 665
8.635.1 Detailed Description . . . 665
8.635.2 Field Documentation . . . 666
8.635.2.1 messageIndex . . . 666
8.635.2.2 messageTag . . . 666
8.635.2.3 pMessageMode . . . 666
8.635.2.4 storageType . . . 666
8.636 pack_sms_SLQSSendAsyncSMS_t Struct Reference . . . 666
8.636.1 Detailed Description . . . 666
8.636.2 Field Documentation . . . 666
8.636.2.1 pSendSmsParams . . . 666
8.637 pack_sms_SLQSSetIndicationRegister_t Struct Reference . . . 666
8.637.1 Detailed Description . . . 666
8.637.2 Field Documentation . . . 667
8.637.2.1 pSetIndicationRegReq . . . 667
8.638 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . 667
8.638.1 Detailed Description . . . 667
8.638.2 Field Documentation . . . 667
8.638.2.1 broadcastActivate . . . 667
8.638.2.2 mode . . . 667
8.639 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference . . . 667
8.639.1 Detailed Description . . . 667
8.639.2 Field Documentation . . . 668
8.639.2.1 mode . . . 668
8.639.2.2 pBroadcastConfig . . . 668
8.639.2.3 pCDMABroadcastConfig . . . 668
8.640 pack_sms_SLQSSetSmsStorage_t Struct Reference . . . 668
8.640.1 Detailed Description . . . 668
8.640.2 Field Documentation . . . 668
8.640.2.1 smsStorage . . . 668
8.641 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 668
8.641.1 Detailed Description . . . 668

8.641.2 Field Documentation . . . 669
8.641.2.1 pMaxStorageSizeReq . . . 669
8.642 pack_sms_SLQSSmsSetRoutes_t Struct Reference . . . 669
8.642.1 Detailed Description . . . 669
8.642.2 Field Documentation . . . 669
8.642.2.1 pSetRoutesReq . . . 669
8.643 pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . 669
8.643.1 Detailed Description . . . 669
8.643.2 Field Documentation . . . 670
8.643.2.1 fix_rate . . . 670
8.643.2.2 fix_type . . . 670
8.643.2.3 function . . . 671
8.643.2.4 max_dist . . . 671
8.643.2.5 max_time . . . 671
8.643.2.6 set_fix_rate . . . 671
8.643.2.7 set_fix_type . . . 671
8.643.2.8 set_function . . . 671
8.643.2.9 set_max_dist . . . 671
8.643.2.10 set_max_time . . . 671
8.644 pack_swioma_SLQSOMADMCancelSession_t Struct Reference . . . 671
8.644.1 Detailed Description . . . 671
8.644.2 Field Documentation . . . 671
8.644.2.1 sessionType . . . 671
8.645 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 671
8.645.1 Detailed Description . . . 671
8.645.2 Field Documentation . . . 672
8.645.2.1 SessionType . . . 672
8.646 pack_swioma_SLQSOMADMSendSelection_t Struct Reference . . . 672
8.646.1 Detailed Description . . . 672
8.646.2 Field Documentation . . . 672
8.646.2.1 pDeferTime . . . 672
8.646.2.2 pRejectReason . . . 672
8.646.2.3 selection . . . 672
8.647 pack_swioma_SLQSOMADMSetSettings_t Struct Reference . . . 672
8.647.1 Detailed Description . . . 673
8.647.2 Field Documentation . . . 673
8.647.2.1 FOTAdownload . . . 673
8.647.2.2 FOTAUpdate . . . 673
8.647.2.3 pAutosdm . . . 673
8.647.2.4 pFwAutoCheck . . . 673

8.648 pack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 673
8.648.1 Detailed Description . . . 674
8.648.2 Field Documentation . . . 674
8.648.2.1 sessionType . . . 674
8.649 pack_uim_ChangePin_t Struct Reference . . . 674
8.649.1 Detailed Description . . . 674
8.649.2 Field Documentation . . . 675
8.649.2.1 changePIN . . . 675
8.649.2.2 EncryptedPIN1 . . . 675
8.649.2.3 pIndicationToken . . . 675
8.649.2.4 pKeyReferenceID . . . 675
8.649.2.5 sessionInfo . . . 675
8.649.2.6 Tlvresult . . . 675
8.650 pack_uim_ReadTransparent_t Struct Reference . . . 675
8.650.1 Detailed Description . . . 675
8.650.2 Field Documentation . . . 676
8.650.2.1 fileIndex . . . 676
8.650.2.2 pEncryptData . . . 676
8.650.2.3 pIndicationToken . . . 676
8.650.2.4 readTransparent . . . 676
8.650.2.5 sessionInfo . . . 676
8.650.2.6 Tlvresult . . . 676
8.651 pack_uim_SetPinProtection_t Struct Reference . . . 676
8.651.1 Detailed Description . . . 676
8.651.2 Field Documentation . . . 677
8.651.2.1 EncryptedPIN1 . . . 677
8.651.2.2 pIndicationToken . . . 677
8.651.2.3 pinProtection . . . 677
8.651.2.4 pKeyReferenceID . . . 677
8.651.2.5 sessionInfo . . . 677
8.651.2.6 Tlvresult . . . 677
8.652 pack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 677
8.652.1 Detailed Description . . . 677
8.652.2 Field Documentation . . . 678
8.652.2.1 authData . . . 678
8.652.2.2 pIndicationToken . . . 678
8.652.2.3 sessionInfo . . . 678
8.653 pack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 678
8.653.1 Detailed Description . . . 678
8.653.2 Field Documentation . . . 678

8.653.2.1 depersonilisationInfo . . . 678
8.654 pack_uim_SLQSUIMEventRegister_t Struct Reference . . . 678
8.654.1 Detailed Description . . . 678
8.654.2 Field Documentation . . . 678
8.654.2.1 eventMask . . . 678
8.655 pack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 678
8.655.1 Detailed Description . . . 679
8.655.2 Field Documentation . . . 679
8.655.2.1 pConfigurationMask . . . 679
8.656 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 679
8.656.1 Detailed Description . . . 679
8.656.2 Field Documentation . . . 680
8.656.2.1 fileIndex . . . 680
8.656.2.2 pIndicationToken . . . 680
8.656.2.3 sessionInfo . . . 680
8.657 pack_uim_SLQSUIMPowerDown_t Struct Reference . . . 680
8.657.1 Detailed Description . . . 680
8.657.2 Field Documentation . . . 680
8.657.2.1 slot . . . 680
8.658 pack_uim_SLQSUIMPowerUp_t Struct Reference . . . 680
8.658.1 Detailed Description . . . 680
8.658.2 Field Documentation . . . 681
8.658.2.1 pIgnoreHotSwapSwitch . . . 681
8.658.2.2 slot . . . 681
8.659 pack_uim_SLQSUIMRefreshComplete_t Struct Reference . . . 681
8.659.1 Detailed Description . . . 681
8.659.2 Field Documentation . . . 681
8.659.2.1 refreshComplete . . . 681
8.659.2.2 sessionInfo . . . 681
8.660 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 681
8.660.1 Detailed Description . . . 681
8.660.2 Field Documentation . . . 682
8.660.2.1 sessionInfo . . . 682
8.661 pack_uim_SLQSUIMRefreshOK_t Struct Reference . . . 682
8.661.1 Detailed Description . . . 682
8.661.2 Field Documentation . . . 682
8.661.2.1 OKtoRefresh . . . 682
8.661.2.2 sessionInfo . . . 682
8.662 pack_uim_SLQSUIMRefreshRegister_t Struct Reference . . . 682
8.662.1 Detailed Description . . . 682

8.662.2 Field Documentation . . . 683
8.662.2.1 regRefresh . . . 683
8.662.2.2 sessionInfo . . . 683
8.663 pack_uim_SLQSUIMSwitchSlot_t Struct Reference . . . 683
8.663.1 Detailed Description . . . 683
8.663.2 Field Documentation . . . 683
8.663.2.1 bLogicalSlot . . . 683
8.663.2.2 ulPhysicalSlot . . . 683
8.664 pack_uim_UnblockPin_t Struct Reference . . . 684
8.664.1 Detailed Description . . . 684
8.664.2 Field Documentation . . . 684
8.664.2.1 EncryptedPIN1 . . . 684
8.664.2.2 pIndicationToken . . . 684
8.664.2.3 pinProtection . . . 684
8.664.2.4 pKeyReferenceID . . . 684
8.664.2.5 sessionInfo . . . 684
8.664.2.6 Tlvresult . . . 685
8.665 pack_uim_VerifyPin_t Struct Reference . . . 685
8.665.1 Detailed Description . . . 685
8.665.2 Field Documentation . . . 685
8.665.2.1 pEncryptedPIN1 . . . 685
8.665.2.2 pIndicationToken . . . 685
8.665.2.3 pKeyReferenceID . . . 686
8.665.2.4 sessionInfo . . . 686
8.665.2.5 Tlvresult . . . 686
8.665.2.6 verifyPIN . . . 686
8.666 pack_voice_AnswerUSSD_t Struct Reference . . . 686
8.666.1 Detailed Description . . . 686
8.666.2 Field Documentation . . . 686
8.666.2.1 pInfo . . . 686
8.667 pack_voice_OriginateUSSD_t Struct Reference . . . 686
8.667.1 Detailed Description . . . 686
8.667.2 Field Documentation . . . 686
8.667.2.1 pInfo . . . 686
8.668 pack_voice_SLQSOriginateUSSD_t Struct Reference . . . 686
8.668.1 Detailed Description . . . 687
8.668.2 Field Documentation . . . 687
8.668.2.1 ussData . . . 687
8.668.2.2 ussDCS . . . 687
8.668.2.3 ussLen . . . 687

8.669 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference . . . 687
8.669.1 Detailed Description . . . 687
8.669.2 Field Documentation . . . 687
8.669.2.1 lineValue . . . 687
8.670 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference . . . 687
8.670.1 Detailed Description . . . 688
8.670.2 Field Documentation . . . 688
8.670.2.1 switchOption . . . 688
8.671 pack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 688
8.671.1 Detailed Description . . . 688
8.671.2 Field Documentation . . . 688
8.671.2.1 pCallId . . . 688
8.672 pack_voice_SLQSVoiceBindSubscription_t Struct Reference . . . 688
8.672.1 Detailed Description . . . 688
8.672.2 Field Documentation . . . 689
8.672.2.1 subsType . . . 689
8.673 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 689
8.673.1 Detailed Description . . . 689
8.673.2 Field Documentation . . . 689
8.673.2.1 BurstDTMFInfo . . . 689
8.673.2.2 pBurstDTMFLengths . . . 689
8.674 pack_voice_SLQSVoiceDialCall_t Struct Reference . . . 689
8.674.1 Detailed Description . . . 690
8.674.2 Field Documentation . . . 691
8.674.2.1 callNumber . . . 691
8.674.2.2 pCallPartySubAdd . . . 691
8.674.2.3 pCallType . . . 691
8.674.2.4 pCLIRType . . . 691
8.674.2.5 pCUGInfo . . . 691
8.674.2.6 pEmergencyCategory . . . 691
8.674.2.7 pSvcType . . . 691
8.674.2.8 pUUSInFo . . . 691
8.675 pack_voice_SLQSVoiceEndCall_t Struct Reference . . . 691
8.675.1 Detailed Description . . . 691
8.675.2 Field Documentation . . . 691
8.675.2.1 pCallId . . . 691
8.676 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 691
8.676.1 Detailed Description . . . 692
8.676.2 Field Documentation . . . 692
8.676.2.1 pSvcClass . . . 692

8.676.2.2 reason . . . 692
8.677 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 692
8.677.1 Detailed Description . . . 692
8.677.2 Field Documentation . . . 693
8.677.2.1 pSvcClass . . . 693
8.677.2.2 Reason . . . 693
8.678 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 693
8.678.1 Detailed Description . . . 693
8.678.2 Field Documentation . . . 693
8.678.2.1 callID . . . 693
8.679 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 693
8.679.1 Detailed Description . . . 694
8.679.2 Field Documentation . . . 694
8.679.2.1 pSvcClass . . . 694
8.680 pack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 694
8.680.1 Detailed Description . . . 694
8.680.2 Field Documentation . . . 695
8.680.2.1 pAirTimer . . . 695
8.680.2.2 pAMRStatus . . . 695
8.680.2.3 pAutoAnswer . . . 695
8.680.2.4 pNamID . . . 695
8.680.2.5 pPrefVoicePrivacy . . . 695
8.680.2.6 pPrefVoiceSO . . . 695
8.680.2.7 pRoamTimer . . . 695
8.680.2.8 pTTYMode . . . 695
8.680.2.9 pVoiceDomainPref . . . 695
8.681 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference . . . 696
8.681.1 Detailed Description . . . 696
8.681.2 Field Documentation . . . 696
8.681.2.1 pRegDTMFEvents . . . 696
8.681.2.2 pRegVoicePrivacyEvents . . . 696
8.681.2.3 pSuppsNotifEvents . . . 696
8.682 pack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 696
8.682.1 Detailed Description . . . 697
8.682.2 Field Documentation . . . 697
8.682.2.1 pCallID . . . 697
8.682.2.2 SUPSType . . . 697
8.683 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference . . . 697
8.683.1 Detailed Description . . . 697
8.683.2 Field Documentation . . . 698

8.683.2.1 USSInformation . . . 698
8.684 pack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 698
8.684.1 Detailed Description . . . 698
8.684.2 Field Documentation . . . 698
8.684.2.1 pCallID . . . 698
8.684.2.2 pFlashPayLd . . . 698
8.684.2.3 pFlashType . . . 698
8.685 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 698
8.685.1 Detailed Description . . . 699
8.685.2 Field Documentation . . . 699
8.685.2.1 newPasswd . . . 699
8.685.2.2 newPasswdAgain . . . 699
8.685.2.3 oldPasswd . . . 699
8.685.2.4 Reason . . . 699
8.686 pack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 699
8.686.1 Detailed Description . . . 700
8.686.2 Field Documentation . . . 701
8.686.2.1 pAirTimerConfig . . . 701
8.686.2.2 pAutoAnswer . . . 701
8.686.2.3 pPrefVoiceDomain . . . 701
8.686.2.4 pPrefVoiceSO . . . 701
8.686.2.5 pRoamTimerConfig . . . 701
8.686.2.6 pTTYMode . . . 701
8.687 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference . . . 701
8.687.1 Detailed Description . . . 701
8.687.2 Field Documentation . . . 701
8.687.2.1 privacyPref . . . 701
8.688 pack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 701
8.688.1 Detailed Description . . . 701
8.688.2 Field Documentation . . . 703
8.688.2.1 pCallBarringPasswd . . . 703
8.688.2.2 pCallForwardingNumber . . . 703
8.688.2.3 pCallFwdTypeAndPlan . . . 703
8.688.2.4 pServiceClass . . . 703
8.688.2.5 pTimerVal . . . 703
8.688.2.6 reason . . . 703
8.688.2.7 voiceSvc . . . 703
8.689 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 703
8.689.1 Detailed Description . . . 703
8.689.2 Field Documentation . . . 704

8.689.2.1 DTMFdigit . . . 704
8.689.2.2 pCallID . . . 704
8.690 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 704
8.690.1 Detailed Description . . . 704
8.690.2 Field Documentation . . . 704
8.690.2.1 callID . . . 704
8.691 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference . . . 704
8.691.1 Detailed Description . . . 704
8.691.2 Field Documentation . . . 705
8.691.2.1 pEnableNotification . . . 705
8.692 pack_wds_GetDefaultProfile_t Struct Reference . . . 705
8.692.1 Detailed Description . . . 705
8.692.2 Field Documentation . . . 705
8.692.2.1 profiletype . . . 705
8.693 pack_wds_GetDefaultProfileNum_t Struct Reference . . . 705
8.693.1 Detailed Description . . . 705
8.693.2 Field Documentation . . . 705
8.693.2.1 family . . . 705
8.693.2.2 type . . . 705
8.694 pack_wds_GetDormancyState_t Struct Reference . . . 705
8.694.1 Detailed Description . . . 705
8.695 pack_wds_GetLastMobileIPError_t Struct Reference . . . 706
8.695.1 Detailed Description . . . 706
8.696 pack_wds_GetMobileIP_t Struct Reference . . . 706
8.696.1 Detailed Description . . . 706
8.697 pack_wds_GetMobileIPProfile_t Struct Reference . . . 706
8.697.1 Detailed Description . . . 706
8.697.2 Field Documentation . . . 706
8.697.2.1 index . . . 706
8.698 pack_wds_GetPacketStatistics_t Struct Reference . . . 706
8.698.1 Detailed Description . . . 707
8.698.2 Field Documentation . . . 707
8.698.2.1 pStatMask . . . 707
8.699 pack_wds_GetPacketStatus_t Struct Reference . . . 707
8.699.1 Detailed Description . . . 707
8.699.2 Field Documentation . . . 707
8.699.2.1 statmask . . . 707
8.700 pack_wds_GetSessionDuration_t Struct Reference . . . 707
8.700.1 Detailed Description . . . 707
8.701 pack_wds_RMSetTransferStatistics_t Struct Reference . . . 707

8.701.1 Detailed Description . . . 708
8.701.2 Field Documentation . . . 708
8.701.2.1 RmTrasnferStaticsReq . . . 708
8.702 pack_wds_SetAutoconnect_t Struct Reference . . . 708
8.702.1 Detailed Description . . . 708
8.702.2 Field Documentation . . . 708
8.702.2.1 acroamsetting . . . 708
8.702.2.2 acsetting . . . 708
8.703 pack_wds_SetDefaultProfile_t Struct Reference . . . 708
8.703.1 Detailed Description . . . 709
8.703.2 Field Documentation . . . 710
8.703.2.1 authentication . . . 710
8.703.2.2 ipAddress . . . 710
8.703.2.3 pApnname . . . 710
8.703.2.4 pdpType . . . 710
8.703.2.5 pName . . . 710
8.703.2.6 pPassword . . . 710
8.703.2.7 primaryDNS . . . 710
8.703.2.8 profileType . . . 710
8.703.2.9 pUsername . . . 710
8.703.2.10 secondaryDNS . . . 710
8.704 pack_wds_SetDefaultProfileNum_t Struct Reference . . . 710
8.704.1 Detailed Description . . . 710
8.704.2 Field Documentation . . . 711
8.704.2.1 family . . . 711
8.704.2.2 index . . . 711
8.704.2.3 type . . . 711
8.705 pack_wds_SetMobileIP_t Struct Reference . . . 711
8.705.1 Detailed Description . . . 711
8.705.2 Field Documentation . . . 711
8.705.2.1 mode . . . 711
8.706 pack_wds_SetMobileIPParameters_t Struct Reference . . . 711
8.706.1 Detailed Description . . . 711
8.706.2 Field Documentation . . . 712
8.706.2.1 pHA2002bis . . . 712
8.706.2.2 pHAAuthenticator . . . 712
8.706.2.3 pMode . . . 712
8.706.2.4 pReRegPeriod . . . 712
8.706.2.5 pReRegTraffic . . . 712
8.706.2.6 pRetryInterval . . . 712

8.706.2.7 pRetryLimit . . . 712
8.706.2.8 pSPC . . . 712
8.707 pack_wds_SetMobileIPProfile_t Struct Reference . . . 712
8.707.1 Detailed Description . . . 713
8.707.2 Field Documentation . . . 714
8.707.2.1 index . . . 714
8.707.2.2 pAAASPI . . . 714
8.707.2.3 pAddress . . . 714
8.707.2.4 pEnabled . . . 714
8.707.2.5 pHASPI . . . 714
8.707.2.6 pMNAAA . . . 714
8.707.2.7 pMNHA . . . 714
8.707.2.8 pNAI . . . 714
8.707.2.9 pPrimaryHA . . . 714
8.707.2.10 pRevTunneling . . . 714
8.707.2.11 pSecondaryHA . . . 714
8.707.2.12 spc . . . 714
8.708 pack_wds_SLQSCreateProfile_t Struct Reference . . . 714
8.708.1 Detailed Description . . . 714
8.708.2 Field Documentation . . . 715
8.708.2.1 pCurProfile . . . 715
8.708.2.2 pProfileId . . . 715
8.708.2.3 pProfileType . . . 715
8.709 pack_wds_SLQSDeleteProfile_t Struct Reference . . . 715
8.709.1 Detailed Description . . . 715
8.709.2 Field Documentation . . . 715
8.709.2.1 profileIndex . . . 715
8.709.2.2 profileType . . . 715
8.710 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 715
8.710.1 Detailed Description . . . 715
8.711 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 716
8.711.1 Detailed Description . . . 716
8.712 pack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 716
8.712.1 Detailed Description . . . 716
8.712.2 Field Documentation . . . 717
8.712.2.1 Mask . . . 717
8.712.2.2 pReportChannelRate . . . 717
8.712.2.3 pReportConnStatus . . . 717
8.712.2.4 pReportDataBearerTech . . . 717
8.712.2.5 pReportDormStatus . . . 717

8.712.2.6 pTransferStatInd . . . 717
8.713pack_wds_SLQSGetProfileSettings_t Struct Reference . . . 717
8.713.1 Detailed Description . . . 717
8.713.2 Field Documentation . . . 718
8.713.2.1 ProfileId . . . 718
8.713.2.2 ProfileType . . . 718
8.714pack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 718
8.714.1 Detailed Description . . . 718
8.714.2 Field Documentation . . . 719
8.714.2.1 pReqSettings . . . 719
8.715pack_wds_SLQSModifyProfile_t Struct Reference . . . 719
8.715.1 Detailed Description . . . 719
8.715.2 Field Documentation . . . 720
8.715.2.1 curProfile . . . 720
8.715.2.2 pProfileId . . . 720
8.715.2.3 pProfileType . . . 720
8.716pack_wds_SLQSSet3GPPConfigItem_t Struct Reference . . . 720
8.716.1 Detailed Description . . . 720
8.716.2 Field Documentation . . . 721
8.716.2.1 LTEAttachProfileListLen . . . 721
8.716.2.2 p3gppRelease . . . 721
8.716.2.3 pDefaultPDNEnabled . . . 721
8.716.2.4 pLTEAttachProfile . . . 721
8.716.2.5 pLTEAttachProfileList . . . 721
8.716.2.6 pProfileList . . . 722
8.717pack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 722
8.717.1 Detailed Description . . . 722
8.717.2 Field Documentation . . . 722
8.717.2.1 IPFamilyPreference . . . 722
8.718pack_wds_SLQSSetWdsEventCallback_t Struct Reference . . . 722
8.718.1 Detailed Description . . . 722
8.718.2 Field Documentation . . . 722
8.718.2.1 currentDataBearer . . . 723
8.718.2.2 dataBearer . . . 723
8.718.2.3 dataBearerTechExt . . . 723
8.718.2.4 dataSystemStatus . . . 723
8.718.2.5 dormancyStatus . . . 723
8.718.2.6 interval . . . 723
8.718.2.7 mobileIP . . . 723
8.718.2.8 transferStats . . . 723

8.719 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 723
8.719.1 Detailed Description . . . 723
8.719.2 Field Documentation . . . 723
8.719.2.1 pProfileId . . . 723
8.720 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference . . . 723
8.720.1 Detailed Description . . . 723
8.720.2 Field Documentation . . . 724
8.720.2.1 pHwConfig . . . 724
8.720.2.2 pProfileId . . . 724
8.720.2.3 pRequestOptionList . . . 724
8.721 pack_wds_SLQSSSetLoopback_t Struct Reference . . . 724
8.721.1 Detailed Description . . . 724
8.721.2 Field Documentation . . . 724
8.721.2.1 loopbackMode . . . 724
8.721.2.2 loopbackMultiplier . . . 724
8.722 pack_wds_SLQSStartDataSession_t Struct Reference . . . 724
8.722.1 Detailed Description . . . 724
8.722.2 Field Documentation . . . 725
8.722.2.1 pAuth . . . 725
8.722.2.2 pPass . . . 725
8.722.2.3 pprofileid3gpp . . . 725
8.722.2.4 pprofileid3gpp2 . . . 725
8.722.2.5 pTech . . . 725
8.722.2.6 pUser . . . 725
8.723 pack_wds_SLQSStopDataSession_t Struct Reference . . . 725
8.723.1 Detailed Description . . . 726
8.723.2 Field Documentation . . . 726
8.723.2.1 psid . . . 726
8.724 pack_wds_SLQSWdsSetEventReport_t Struct Reference . . . 726
8.724.1 Detailed Description . . . 726
8.724.2 Field Documentation . . . 727
8.724.2.1 pCurrChannelRateInd . . . 727
8.724.2.2 pCurrDataBearerTechInd . . . 727
8.724.2.3 pCurrPrefDataSysInd . . . 727
8.724.2.4 pDataBearerTechInd . . . 727
8.724.2.5 pDataCallStatusChangeInd . . . 727
8.724.2.6 pDataSystemStatusChangeInd . . . 727
8.724.2.7 pDormancyStatusInd . . . 727
8.724.2.8 pEVDOPageMonPerChangeInd . . . 727
8.724.2.9 pMIPStatusInd . . . 727

8.724.2.10 pTransferStatInd . . . 727
8.725 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 728
8.725.1 Detailed Description . . . 728
8.725.2 Field Documentation . . . 728
8.725.2.1 contextId . . . 728
8.725.2.2 contextType . . . 728
8.726 PackCreateProfileOut Struct Reference . . . 728
8.726.1 Detailed Description . . . 728
8.726.2 Field Documentation . . . 729
8.726.2.1 ExtErrorCode . . . 729
8.726.2.2 ProfileIndex . . . 729
8.726.2.3 ProfileType . . . 729
8.727 packgetDyingGaspCfg Struct Reference . . . 729
8.727.1 Detailed Description . . . 729
8.727.2 Field Documentation . . . 729
8.727.2.1 pDestSMSContent . . . 729
8.727.2.2 pDestSMSNum . . . 729
8.728 packgetDyingGaspStatistics Struct Reference . . . 729
8.728.1 Detailed Description . . . 729
8.728.2 Field Documentation . . . 730
8.728.2.1 pSMSAttemptedFlag . . . 730
8.728.2.2 pTimeStamp . . . 730
8.729 PCMparams Struct Reference . . . 730
8.729.1 Detailed Description . . . 730
8.729.2 Field Documentation . . . 730
8.729.2.1 iFaceTab . . . 730
8.729.2.2 iFaceTabLen . . . 730
8.730 PCSCFFQDNAddress Struct Reference . . . 730
8.730.1 Detailed Description . . . 730
8.730.2 Field Documentation . . . 731
8.730.2.1 fqdnAddr . . . 731
8.730.2.2 fqdnLen . . . 731
8.731 PCSCFFQDNAddressList Struct Reference . . . 731
8.731.1 Detailed Description . . . 731
8.731.2 Field Documentation . . . 731
8.731.2.1 numInstances . . . 731
8.731.2.2 pcsfFQDNAddress . . . 731
8.732 PCSCFIPv4ServerAddressList Struct Reference . . . 731
8.732.1 Detailed Description . . . 731
8.732.2 Field Documentation . . . 732

8.732.2.1 numInstances . . . 732
8.732.2.2 pscsfIPv4Addr . . . 732
8.733 PDSPositionData Struct Reference . . . 732
8.733.1 Detailed Description . . . 732
8.733.2 Field Documentation . . . 733
8.733.2.1 pAltitudeWrtEllipsoid . . . 733
8.733.2.2 pAltitudeWrtSealevel . . . 733
8.733.2.3 pHorizontalConfidence . . . 733
8.733.2.4 pHorizontalUncCircular . . . 734
8.733.2.5 pLatitude . . . 734
8.733.2.6 pLongitude . . . 734
8.733.2.7 pPositionSource . . . 734
8.733.2.8 pTimeStamp . . . 734
8.733.2.9 pTimeType . . . 734
8.733.2.10 pVerticalConfidence . . . 734
8.733.2.11 pVerticalUnc . . . 734
8.734 PDSPosMethodStateReq Struct Reference . . . 734
8.734.1 Detailed Description . . . 734
8.734.2 Field Documentation . . . 734
8.734.2.1 pWifiState . . . 734
8.734.2.2 pXtraDataState . . . 735
8.734.2.3 pXtraTimeState . . . 735
8.735 peerNumberInfo Struct Reference . . . 735
8.735.1 Detailed Description . . . 735
8.735.2 Field Documentation . . . 736
8.735.2.1 callID . . . 736
8.735.2.2 number . . . 736
8.735.2.3 numLen . . . 736
8.735.2.4 numPI . . . 736
8.735.2.5 numPlan . . . 736
8.735.2.6 numSI . . . 736
8.735.2.7 numType . . . 736
8.736 personalizationStatus Struct Reference . . . 736
8.736.1 Detailed Description . . . 736
8.736.2 Field Documentation . . . 737
8.736.2.1 feature . . . 737
8.736.2.2 numFeatures . . . 737
8.736.2.3 unblockLeft . . . 737
8.736.2.4 verifyLeft . . . 737
8.737 PhyCaAggPcellInfo Struct Reference . . . 737

8.737.1 Detailed Description . . . 737
8.737.2 Field Documentation . . . 738
8.737.2.1 dl_bw_value . . . 738
8.737.2.2 freq . . . 738
8.737.2.3 iLTEbandValue . . . 738
8.737.2.4 pci . . . 738
8.737.2.5 TlvPresent . . . 738
8.738 PhyCaAggScellDlBw Struct Reference . . . 738
8.738.1 Detailed Description . . . 738
8.738.2 Field Documentation . . . 738
8.738.2.1 dl_bw_value . . . 738
8.738.2.2 TlvPresent . . . 738
8.739 PhyCaAggScellIndex Struct Reference . . . 739
8.739.1 Detailed Description . . . 739
8.739.2 Field Documentation . . . 739
8.739.2.1 scell_idx . . . 739
8.739.2.2 TlvPresent . . . 739
8.740 PhyCaAggScellIndType Struct Reference . . . 739
8.740.1 Detailed Description . . . 739
8.740.2 Field Documentation . . . 740
8.740.2.1 freq . . . 740
8.740.2.2 pci . . . 740
8.740.2.3 scell_state . . . 740
8.740.2.4 TlvPresent . . . 740
8.741 PhyCaAggScellInfo Struct Reference . . . 740
8.741.1 Detailed Description . . . 740
8.741.2 Field Documentation . . . 742
8.741.2.1 dl_bw_value . . . 742
8.741.2.2 freq . . . 742
8.741.2.3 iLTEbandValue . . . 742
8.741.2.4 pci . . . 742
8.741.2.5 scell_state . . . 742
8.741.2.6 TlvPresent . . . 742
8.742 PilotSetData Struct Reference . . . 742
8.742.1 Detailed Description . . . 742
8.742.2 Field Documentation . . . 742
8.742.2.1 NumPilots . . . 742
8.742.2.2 pPilotSetInfo . . . 742
8.743 PilotSetParams Struct Reference . . . 743
8.743.1 Detailed Description . . . 743

8.743.2 Field Documentation . . . 743
8.743.2.1 PilotPN . . . 743
8.743.2.2 PilotStrength . . . 743
8.743.2.3 PilotType . . . 743
8.744 pktErrRate Struct Reference . . . 743
8.744.1 Detailed Description . . . 743
8.744.2 Field Documentation . . . 743
8.744.2.1 exponent . . . 743
8.744.2.2 multiplier . . . 744
8.745 PlmnID Struct Reference . . . 744
8.745.1 Detailed Description . . . 744
8.745.2 Field Documentation . . . 744
8.745.2.1 mcc . . . 744
8.745.2.2 mnc . . . 744
8.745.2.3 pcsDigit . . . 744
8.745.2.4 TlvPresent . . . 744
8.746 PLMNNetworkName Struct Reference . . . 744
8.746.1 Detailed Description . . . 745
8.746.2 Field Documentation . . . 745
8.746.2.1 numInstance . . . 745
8.746.2.2 PLMNNetName . . . 745
8.747 PLMNNetworkNameData Struct Reference . . . 745
8.747.1 Detailed Description . . . 745
8.747.2 Field Documentation . . . 746
8.747.2.1 codingScheme . . . 746
8.747.2.2 countryInitials . . . 746
8.747.2.3 longName . . . 746
8.747.2.4 longNameLen . . . 746
8.747.2.5 longNameSpareBits . . . 746
8.747.2.6 shortName . . . 746
8.747.2.7 shortNameLen . . . 747
8.747.2.8 shortNameSpareBits . . . 747
8.748 Port Struct Reference . . . 747
8.748.1 Detailed Description . . . 747
8.748.2 Field Documentation . . . 747
8.748.2.1 port . . . 747
8.748.2.2 range . . . 747
8.749 precisionDilution_s Struct Reference . . . 747
8.749.1 Detailed Description . . . 747
8.749.2 Field Documentation . . . 748

8.749.2.1 HDOP . . . 748
8.749.2.2 PDOP . . . 748
8.749.2.3 VDOP . . . 748
8.750 PrefImageList Struct Reference . . . 748
8.750.1 Detailed Description . . . 748
8.750.2 Field Documentation . . . 748
8.750.2.1 listEntries . . . 748
8.750.2.2 listSize . . . 748
8.751 prefVoiceSO Struct Reference . . . 748
8.751.1 Detailed Description . . . 749
8.751.2 Field Documentation . . . 750
8.751.2.1 evrcCapability . . . 750
8.751.2.2 homeOrigVoiceSO . . . 750
8.751.2.3 homePageVoiceSO . . . 750
8.751.2.4 namID . . . 750
8.751.2.5 roamOrigVoiceSO . . . 750
8.752 Profile3GPP Struct Reference . . . 750
8.752.1 Detailed Description . . . 751
8.752.2 Field Documentation . . . 755
8.752.2.1 pAddrAllocPref . . . 755
8.752.2.2 pAPNClass . . . 755
8.752.2.3 pAPNDisabledFlag . . . 755
8.752.2.4 pAPNName . . . 755
8.752.2.5 pAPNnameSize . . . 755
8.752.2.6 pAuthenticationPref . . . 755
8.752.2.7 pGPRSMinimumQoS . . . 755
8.752.2.8 pGPRSRequestedQos . . . 755
8.752.2.9 pImCnFlag . . . 755
8.752.2.10 pIPv4AddrPref . . . 755
8.752.2.11 pIPv6AddPref . . . 755
8.752.2.12 pPassword . . . 755
8.752.2.13 pPasswordSize . . . 755
8.752.2.14 pPcscfAddrUsingDhcp . . . 755
8.752.2.15 pPcscfAddrUsingPCO . . . 755
8.752.2.16 pPDNInactivTimeout . . . 755
8.752.2.17 pPdpAccessConFlag . . . 755
8.752.2.18 pPdpContext . . . 755
8.752.2.19 pPdpDataCompType . . . 755
8.752.2.20 pPdpHdrCompType . . . 755
8.752.2.21 pPDPtype . . . 755

8.752.2.22 pPriDNSIPv4AddPref . . . 755
8.752.2.23 pPriDNSIPv6addpref . . . 755
8.752.2.24 pPrimaryID . . . 756
8.752.2.25 pProfilename . . . 756
8.752.2.26 pProfilenameSize . . . 756
8.752.2.27 pQosClassID . . . 756
8.752.2.28 pSecDNSIPv4AddPref . . . 756
8.752.2.29 pSecDNSIPv6addpref . . . 756
8.752.2.30 pSecondaryFlag . . . 756
8.752.2.31 pTFTID1Params . . . 756
8.752.2.32 pTFTID2Params . . . 756
8.752.2.33 pUMTSMinQoS . . . 756
8.752.2.34 pUMTSMinQosSigInd . . . 756
8.752.2.35 pUMTSReqQoS . . . 756
8.752.2.36 pUMTSReqQoSSigInd . . . 756
8.752.2.37 pUsername . . . 756
8.752.2.38 pUsernameSize . . . 756
8.753 Profile3GPP2 Struct Reference . . . 756
8.753.1 Detailed Description . . . 757
8.753.2 Field Documentation . . . 760
8.753.2.1 pAllowLinger . . . 760
8.753.2.2 pAPNClass3GPP2 . . . 760
8.753.2.3 pAPNEnabled3GPP2 . . . 760
8.753.2.4 pApnString . . . 760
8.753.2.5 pApnStringSize . . . 760
8.753.2.6 pAppPriority . . . 760
8.753.2.7 pAppType . . . 760
8.753.2.8 pAuthPassword . . . 760
8.753.2.9 pAuthPasswordSize . . . 760
8.753.2.10 pAuthProtocol . . . 761
8.753.2.11 pAuthRetryCount . . . 761
8.753.2.12 pAuthTimeout . . . 761
8.753.2.13 pDataMode . . . 761
8.753.2.14 pDataRate . . . 761
8.753.2.15 pIpcpAckTimeout . . . 761
8.753.2.16 pIpcpCreqRetryCount . . . 761
8.753.2.17 pIsPcscfAddressNedded . . . 761
8.753.2.18 pLcpAckTimeout . . . 761
8.753.2.19 pLcpCreqRetryCount . . . 761
8.753.2.20 pNegoDnsSrvrPref . . . 761

8.753.2.21 pPDNInactivTimeout3GPP2 . . . 761
8.753.2.22 pPdnType . . . 761
8.753.2.23 pPppSessCloseTimer1x . . . 761
8.753.2.24 pPppSessCloseTimerDO . . . 761
8.753.2.25 pPrimaryV4DnsAddress . . . 761
8.753.2.26 pPriV6DnsAddress . . . 761
8.753.2.27 pRATType . . . 761
8.753.2.28 pSecondaryV4DnsAddress . . . 761
8.753.2.29 pSecV6DnsAddress . . . 761
8.753.2.30 pUserId . . . 761
8.753.2.31 pUserIdSize . . . 761
8.754 ProfileIdentifier Struct Reference . . . 761
8.754.1 Detailed Description . . . 761
8.754.2 Field Documentation . . . 762
8.754.2.1 profileIndex . . . 762
8.754.2.2 profileType . . . 762
8.755 protocolSubtypeElement Struct Reference . . . 762
8.755.1 Detailed Description . . . 762
8.755.2 Field Documentation . . . 763
8.755.2.1 AccessMac . . . 763
8.755.2.2 AuthProt . . . 763
8.755.2.3 ControlMac . . . 763
8.755.2.4 EncryptProt . . . 763
8.755.2.5 ForwardMac . . . 763
8.755.2.6 IdleState . . . 763
8.755.2.7 KeyExchange . . . 763
8.755.2.8 MultDisc . . . 763
8.755.2.9 PhysicalLayer . . . 763
8.755.2.10 ReverseMac . . . 763
8.755.2.11 SecProt . . . 763
8.755.2.12 VirtStream . . . 763
8.756 PSDetachReq Struct Reference . . . 763
8.756.1 Detailed Description . . . 764
8.756.2 Field Documentation . . . 764
8.756.2.1 pDetachAction . . . 764
8.757 qaQmi3Gpp2TimeZone Struct Reference . . . 764
8.757.1 Detailed Description . . . 764
8.757.2 Field Documentation . . . 764
8.757.2.1 daylightSavings . . . 764
8.757.2.2 leapSeconds . . . 764

8.757.2.3 localTimeOffset . . . 765
8.758 qaQmiInterfaceInfo Struct Reference . . . 765
8.758.1 Detailed Description . . . 765
8.758.2 Field Documentation . . . 765
8.758.2.1 qaQmiinstanceid . . . 765
8.758.2.2 qaQmisvctype . . . 765
8.758.2.3 v4sessionId . . . 765
8.758.2.4 v6sessionId . . . 765
8.759 qaQmiServingSystemParam Struct Reference . . . 765
8.759.1 Detailed Description . . . 766
8.759.2 Field Documentation . . . 768
8.759.2.1 BasestationID . . . 768
8.759.2.2 BasestationLatitude . . . 768
8.759.2.3 BasestationLongitude . . . 768
8.759.2.4 CallBarStatus . . . 768
8.759.2.5 CDMA_P_Rev . . . 768
8.759.2.6 CDMASystemInfoExt . . . 768
8.759.2.7 CellID . . . 768
8.759.2.8 concSvcInfo . . . 768
8.759.2.9 CurrentPLMN . . . 769
8.759.2.10 DataSrvCapabilities . . . 769
8.759.2.11 defaultRoamInd . . . 769
8.759.2.12 DetailedSvcInfo . . . 769
8.759.2.13 DTMInd . . . 769
8.759.2.14 Gpp2TimeZone . . . 769
8.759.2.15 GppNetworkDSTAdjustment . . . 769
8.759.2.16 GppTimeZone . . . 769
8.759.2.17 hdrPersonality . . . 769
8.759.2.18 Lac . . . 769
8.759.2.19 NetworkID . . . 769
8.759.2.20 PRLInd . . . 769
8.759.2.21 roamIndicatorVal . . . 769
8.759.2.22 RoamingIndicatorList . . . 769
8.759.2.23 ServingSystem . . . 769
8.759.2.24 SystemID . . . 769
8.759.2.25 trackAreaCode . . . 769
8.760 QmiCbkCatEventStatusReportInd Struct Reference . . . 769
8.760.1 Field Documentation . . . 769
8.760.1.1 CCETlv . . . 769
8.760.1.2 event_Index . . . 769

8.761 QmiCbkLocBestAvailPosInd Struct Reference . . . 769
8.761.1 Detailed Description . . . 770
8.761.2 Field Documentation . . . 774
8.761.2.1 pAltitudeWrtEllipsoid . . . 774
8.761.2.2 pAltitudeWrtMeanSeaLevel . . . 774
8.761.2.3 pGpsTime . . . 774
8.761.2.4 pHeading . . . 774
8.761.2.5 pHeadingUnc . . . 774
8.761.2.6 pHorCirConf . . . 774
8.761.2.7 pHorEllpConf . . . 774
8.761.2.8 pHorReliability . . . 774
8.761.2.9 pHorUncCircular . . . 774
8.761.2.10 pHorUncEllipseOrientAzimuth . . . 774
8.761.2.11 pHorUncEllipseSemiMajor . . . 774
8.761.2.12 pHorUncEllipseSemiMinor . . . 775
8.761.2.13 pLatitude . . . 775
8.761.2.14 pLongitude . . . 775
8.761.2.15 pMagneticDeviation . . . 775
8.761.2.16 pPrecisionDilution . . . 775
8.761.2.17 pSensorDataUsage . . . 775
8.761.2.18 pSpeedHorizontal . . . 775
8.761.2.19 pSpeedUnc . . . 775
8.761.2.20 pSpeedVertical . . . 775
8.761.2.21 pSpeedVerticalUnc . . . 775
8.761.2.22 pSvUsedforFix . . . 775
8.761.2.23 pTechnologyMask . . . 775
8.761.2.24 pTimeSrc . . . 775
8.761.2.25 pTimestampUtc . . . 775
8.761.2.26 pTimeUnc . . . 775
8.761.2.27 pVertConfidence . . . 775
8.761.2.28 pVertReliability . . . 775
8.761.2.29 pVertUnc . . . 775
8.761.2.30 pXid . . . 775
8.761.2.31 status . . . 775
8.762 QmiCbkLocCradleMountInd Struct Reference . . . 775
8.762.1 Detailed Description . . . 775
8.762.2 Field Documentation . . . 776
8.762.2.1 cradleMountConfigStatus . . . 776
8.763 QmiCbkLocEngineStateInd Struct Reference . . . 776
8.763.1 Detailed Description . . . 776

8.763.2 Field Documentation . . . 776
8.763.2.1 engineState . . . 776
8.764 QmiCbkLocEventTimeSyncInd Struct Reference . . . 776
8.764.1 Detailed Description . . . 777
8.764.2 Field Documentation . . . 777
8.764.2.1 timeSyncRefCounter . . . 777
8.765 QmiCbkLocGetServerInd Struct Reference . . . 777
8.765.1 Detailed Description . . . 777
8.765.2 Field Documentation . . . 778
8.765.2.1 pIPv4AddrInfo . . . 778
8.765.2.2 pIPv6AddrInfo . . . 778
8.765.2.3 pURL . . . 778
8.765.2.4 serverStatus . . . 778
8.765.2.5 serverType . . . 778
8.766 QmiCbkLocInjectPositionInd Struct Reference . . . 778
8.766.1 Detailed Description . . . 778
8.766.2 Field Documentation . . . 779
8.766.2.1 status . . . 779
8.767 QmiCbkLocInjectSensorDataInd Struct Reference . . . 779
8.767.1 Detailed Description . . . 779
8.767.2 Field Documentation . . . 780
8.767.2.1 injectSensorDataStatus . . . 780
8.767.2.2 pAccelSamplesAccepted . . . 780
8.767.2.3 pAccelTempSamplesAccepted . . . 780
8.767.2.4 pGyroSamplesAccepted . . . 780
8.767.2.5 pGyroTempSamplesAccepted . . . 780
8.767.2.6 pOpaqueIdentifier . . . 780
8.768 QmiCbkLocInjectTimeInd Struct Reference . . . 780
8.768.1 Detailed Description . . . 781
8.768.2 Field Documentation . . . 781
8.768.2.1 injectTimeSyncStatus . . . 781
8.769 QmiCbkLocInjectUTCTimeInd Struct Reference . . . 781
8.769.1 Detailed Description . . . 781
8.769.2 Field Documentation . . . 782
8.769.2.1 status . . . 782
8.770 QmiCbkLocPositionReportInd Struct Reference . . . 782
8.770.1 Detailed Description . . . 782
8.770.2 Field Documentation . . . 786
8.770.2.1 pAltitudeAssumed . . . 786
8.770.2.2 pAltitudeWrtEllipsoid . . . 786

8.770.2.3 pAltitudeWrtMeanSeaLevel . . . 786
8.770.2.4 pFixId . . . 786
8.770.2.5 pGpsTime . . . 786
8.770.2.6 pHeading . . . 787
8.770.2.7 pHeadingUnc . . . 787
8.770.2.8 pHorConfidence . . . 787
8.770.2.9 pHorReliability . . . 787
8.770.2.10 pHorUncCircular . . . 787
8.770.2.11 pHorUncEllipseOrientAzimuth . . . 787
8.770.2.12 pHorUncEllipseSemiMajor . . . 787
8.770.2.13 pHorUncEllipseSemiMinor . . . 787
8.770.2.14 pLatitude . . . 787
8.770.2.15 pLeapSeconds . . . 787
8.770.2.16 pLongitude . . . 787
8.770.2.17 pMagneticDeviation . . . 787
8.770.2.18 pPrecisionDilution . . . 787
8.770.2.19 pSensorDataUsage . . . 787
8.770.2.20 pSpeedHorizontal . . . 787
8.770.2.21 pSpeedUnc . . . 787
8.770.2.22 pSpeedVertical . . . 787
8.770.2.23 pSvUsedforFix . . . 787
8.770.2.24 pTechnologyMask . . . 787
8.770.2.25 pTimeSrc . . . 787
8.770.2.26 pTimestampUtc . . . 787
8.770.2.27 pTimeUnc . . . 787
8.770.2.28 pVertConfidence . . . 787
8.770.2.29 pVertReliability . . . 787
8.770.2.30 pVertUnc . . . 787
8.770.2.31 sessionId . . . 787
8.770.2.32 sessionStatus . . . 787
8.771 QmiCbkLocSensorStreamingInd Struct Reference . . . 787
8.771.1 Detailed Description . . . 788
8.771.2 Field Documentation . . . 788
8.771.2.1 pAccelAcceptReady . . . 788
8.771.2.2 pAccelTempAcceptReady . . . 788
8.771.2.3 pGyroAcceptReady . . . 788
8.771.2.4 pGyroTempAcceptReady . . . 788
8.772 QmiCbkLocSetExtPowerConfigInd Struct Reference . . . 788
8.772.1 Detailed Description . . . 788
8.772.2 Field Documentation . . . 789

8.772.2.1 status . . . 789
8.773 QmiCbkNasLTECphyCaInfo Struct Reference . . . 789
8.773.1 Detailed Description . . . 789
8.773.2 Field Documentation . . . 790
8.773.2.1 sPhyCaAggPcellInfo . . . 790
8.773.2.2 sPhyCaAggScellDlBw . . . 790
8.773.2.3 sPhyCaAggScellIndex . . . 790
8.773.2.4 sPhyCaAggScellIndType . . . 790
8.773.2.5 sPhyCaAggScellInfo . . . 790
8.774 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference . . . 790
8.774.1 Field Documentation . . . 790
8.774.1.1 SITlv . . . 790
8.775 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference . . . 790
8.775.1 Field Documentation . . . 790
8.775.1.1 SITlv . . . 790
8.776 QmiCbkTmdMitiLvlRptInd Struct Reference . . . 790
8.776.1 Detailed Description . . . 790
8.776.2 Field Documentation . . . 791
8.776.2.1 currentMitigationLvl . . . 791
8.776.2.2 MitigationDevInfo . . . 791
8.777 QmiCbkWdsStatisticsIndState Struct Reference . . . 791
8.777.1 Detailed Description . . . 791
8.777.2 Field Documentation . . . 791
8.777.2.1 RxDropConutTlv . . . 791
8.777.2.2 RxOkByteCountTlv . . . 791
8.777.2.3 RxOkConutTlv . . . 791
8.777.2.4 TxDropConutTlv . . . 791
8.777.2.5 TxOkByteCountTlv . . . 791
8.777.2.6 TxOkConutTlv . . . 792
8.778 qmifwinfo_s Struct Reference . . . 792
8.778.1 Detailed Description . . . 792
8.778.2 Field Documentation . . . 792
8.778.2.1 dev . . . 792
8.778.2.2 g . . . 792
8.778.2.3 s . . . 792
8.779 QmiNas3GppNetworkInfo Struct Reference . . . 792
8.779.1 Detailed Description . . . 793
8.779.2 Field Documentation . . . 793
8.779.2.1 pDesription . . . 794
8.779.2.2 pForbidden . . . 794

8.779.2.3 pInUse . . . 794
8.779.2.4 pMCC . . . 794
8.779.2.5 pMNC . . . 794
8.779.2.6 pPreferred . . . 794
8.779.2.7 pRoaming . . . 794
8.780 QmiNasGetRFBandInfoResp Struct Reference . . . 794
8.780.1 Field Documentation . . . 794
8.780.1.1 pInstancesSize . . . 794
8.780.1.2 pRFBandInfoElements . . . 794
8.780.1.3 results . . . 794
8.781 QmiNasPerformNetworkScanResp Struct Reference . . . 794
8.781.1 Field Documentation . . . 794
8.781.1.1 pInstances . . . 794
8.781.1.2 pInstanceSize . . . 794
8.781.1.3 results . . . 794
8.782 qmiSmsMessageList Struct Reference . . . 794
8.782.1 Detailed Description . . . 795
8.782.2 Field Documentation . . . 795
8.782.2.1 messageIndex . . . 795
8.782.2.2 messageTag . . . 795
8.783 qmiWDSDataBearerTechnology Struct Reference . . . 795
8.783.1 Detailed Description . . . 795
8.783.2 Field Documentation . . . 796
8.783.2.1 currentNetwork . . . 796
8.783.2.2 ratMask . . . 796
8.783.2.3 soMask . . . 796
8.784 QmiWdsIpAddressInfo Struct Reference . . . 796
8.784.1 Detailed Description . . . 796
8.784.2 Field Documentation . . . 797
8.784.2.1 pIPAddressV4 . . . 797
8.784.2.2 pIPAddressV6 . . . 797
8.784.2.3 pIPv6prefixlen . . . 797
8.785 qmiWdsRunTimeSettings Struct Reference . . . 797
8.785.1 Detailed Description . . . 797
8.785.2 Field Documentation . . . 799
8.785.2.1 pAPNName . . . 799
8.785.2.2 pAuthentication . . . 800
8.785.2.3 pDomainList . . . 800
8.785.2.4 pGPRSGrantedQoS . . . 800
8.785.2.5 pGWAddressV4 . . . 800

8.785.2.6 pIMCNflag . . . 800
8.785.2.7 pIPAddressV4 . . . 800
8.785.2.8 pIPFamilyPreference . . . 800
8.785.2.9 pIPV6AddrInfo . . . 800
8.785.2.10 pIPV6GWAddrInfo . . . 800
8.785.2.11 pMtu . . . 800
8.785.2.12 pPCSCFAddrPCO . . . 800
8.785.2.13 pPCSCFFQDNAddrList . . . 800
8.785.2.14 pPDPType . . . 800
8.785.2.15 pPrimaryDNSV4 . . . 800
8.785.2.16 pPrimaryDNSV6 . . . 800
8.785.2.17 pProfileID . . . 800
8.785.2.18 pProfileName . . . 800
8.785.2.19 pSecondaryDNSV4 . . . 800
8.785.2.20 pSecondaryDNSV6 . . . 800
8.785.2.21 pServerAddrList . . . 800
8.785.2.22 pSubnetMaskV4 . . . 800
8.785.2.23 pTechnology . . . 800
8.785.2.24 pUMTSGrantedQoS . . . 800
8.785.2.25 pUsername . . . 800
8.786 QosClassID Struct Reference . . . 800
8.786.1 Detailed Description . . . 801
8.786.2 Field Documentation . . . 801
8.786.2.1 gDlBitRate . . . 801
8.786.2.2 gUlBitRate . . . 801
8.786.2.3 maxDlBitRate . . . 801
8.786.2.4 maxUlBitRate . . . 801
8.786.2.5 QCI . . . 801
8.787 QosEventInfo Struct Reference . . . 801
8.787.1 Detailed Description . . . 801
8.787.2 Field Documentation . . . 802
8.787.2.1 pDataBearer . . . 802
8.787.2.2 pPacketsCountRX . . . 802
8.787.2.3 pPacketsCountTX . . . 802
8.787.2.4 pTotalBytesRX . . . 802
8.787.2.5 pTotalBytesTX . . . 803
8.788 QosFlowInfo Struct Reference . . . 803
8.788.1 Detailed Description . . . 803
8.788.2 Field Documentation . . . 803
8.788.2.1 pBearerID . . . 803

8.788.2.2 pQFlowState . . . 803
8.788.2.3 pRxQFilter . . . 803
8.788.2.4 pRxQFlowGranted . . . 803
8.788.2.5 pTxQFilter . . . 803
8.788.2.6 pTxQFlowGranted . . . 804
8.789 QosFlowInfoState Struct Reference . . . 804
8.789.1 Detailed Description . . . 804
8.789.2 Field Documentation . . . 804
8.789.2.1 id . . . 804
8.789.2.2 isNewFlow . . . 804
8.789.2.3 state . . . 804
8.790 QosMap Struct Reference . . . 804
8.790.1 Detailed Description . . . 804
8.790.2 Field Documentation . . . 805
8.790.2.1 dscp . . . 805
8.790.2.2 qos_id . . . 805
8.790.2.3 state . . . 805
8.791 RankIndicatorInd Struct Reference . . . 805
8.791.1 Field Documentation . . . 805
8.791.1.1 Count1 . . . 805
8.791.1.2 Count2 . . . 805
8.792 readResult Struct Reference . . . 805
8.792.1 Detailed Description . . . 805
8.792.2 Field Documentation . . . 806
8.792.2.1 content . . . 806
8.792.2.2 contentLen . . . 806
8.793 readTransparentInfo Struct Reference . . . 806
8.793.1 Detailed Description . . . 806
8.793.2 Field Documentation . . . 806
8.793.2.1 length . . . 806
8.793.2.2 offset . . . 806
8.794 redirNumInfo Struct Reference . . . 806
8.794.1 Detailed Description . . . 806
8.794.2 Field Documentation . . . 808
8.794.2.1 number . . . 808
8.794.2.2 numLen . . . 808
8.794.2.3 numPlan . . . 808
8.794.2.4 numType . . . 808
8.794.2.5 PI . . . 808
8.794.2.6 reason . . . 808

8.794.2.7 SI . . . 808
8.795 registerRefresh Struct Reference . . . 808
8.795.1 Detailed Description . . . 808
8.795.2 Field Documentation . . . 809
8.795.2.1 arrfileInfo . . . 809
8.795.2.2 numFiles . . . 809
8.795.2.3 registerFlag . . . 809
8.795.2.4 voteForInit . . . 809
8.796 remainingRetries Struct Reference . . . 809
8.796.1 Detailed Description . . . 809
8.796.2 Field Documentation . . . 809
8.796.2.1 unblockLeft . . . 809
8.796.2.2 verifyLeft . . . 809
8.797 remotePartyName Struct Reference . . . 809
8.797.1 Detailed Description . . . 810
8.797.2 Field Documentation . . . 810
8.797.2.1 callerName . . . 810
8.797.2.2 codingScheme . . . 810
8.797.2.3 nameLen . . . 810
8.797.2.4 namePI . . . 810
8.798 remotePartyNum Struct Reference . . . 810
8.798.1 Detailed Description . . . 810
8.798.2 Field Documentation . . . 811
8.798.2.1 numLen . . . 811
8.798.2.2 presentationInd . . . 811
8.798.2.3 remPartyNumber . . . 811
8.799 ReqFieldsList Struct Reference . . . 811
8.799.1 Detailed Description . . . 811
8.799.2 Field Documentation . . . 811
8.799.2.1 requestFields . . . 811
8.799.2.2 requestFieldsLen . . . 812
8.800 RespFieldsList Struct Reference . . . 812
8.800.1 Detailed Description . . . 812
8.800.2 Field Documentation . . . 812
8.800.2.1 responseFields . . . 812
8.800.2.2 responseFieldsLen . . . 812
8.801 RFBandInfoElements Struct Reference . . . 812
8.801.1 Detailed Description . . . 812
8.801.2 Field Documentation . . . 813
8.801.2.1 activeBandClass . . . 813

8.801.2.2 activeBandClass . . . 813
8.801.2.3 activeChannel . . . 813
8.801.2.4 activeChannel . . . 813
8.801.2.5 radioInterface . . . 813
8.801.2.6 radioInterface . . . 813
8.802 rmTrasnferStaticsReq Struct Reference . . . 813
8.802.1 Detailed Description . . . 813
8.802.2 Field Documentation . . . 814
8.802.2.1 bResetStatistics . . . 814
8.802.2.2 ulMask . . . 814
8.803 roamIndList Struct Reference . . . 814
8.803.1 Detailed Description . . . 814
8.803.2 Field Documentation . . . 815
8.803.2.1 numInstances . . . 815
8.803.2.2 radioInterface . . . 815
8.803.2.3 roamIndicator . . . 815
8.804 RoamingInfo Struct Reference . . . 815
8.804.1 Field Documentation . . . 815
8.804.1.1 roaming_ind . . . 815
8.804.1.2 TlvPresent . . . 815
8.805 roamTimer Struct Reference . . . 815
8.805.1 Detailed Description . . . 815
8.805.2 Field Documentation . . . 816
8.805.2.1 namID . . . 816
8.805.2.2 roamTimerValue . . . 816
8.806 RSRPThresh Struct Reference . . . 816
8.806.1 Detailed Description . . . 816
8.806.2 Field Documentation . . . 816
8.806.2.1 pRSRPThresList . . . 816
8.806.2.2 RSRPThresListLen . . . 816
8.807 rsrqInformation Struct Reference . . . 816
8.807.1 Detailed Description . . . 816
8.807.2 Field Documentation . . . 817
8.807.2.1 radioIf . . . 817
8.807.2.2 rsrq . . . 817
8.808 RSRQThresh Struct Reference . . . 817
8.808.1 Detailed Description . . . 817
8.808.2 Field Documentation . . . 817
8.808.2.1 pRSRQThresList . . . 817
8.808.2.2 RSRQThresListLen . . . 817

8.809 RSSIThresh Struct Reference . . . 817
8.809.1 Detailed Description . . . 818
8.809.2 Field Documentation . . . 818
8.809.2.1 pRSSIThresList . . . 818
8.809.2.2 RSSIThresListLen . . . 818
8.810 RXAGCList Struct Reference . . . 818
8.810.1 Detailed Description . . . 818
8.810.2 Field Documentation . . . 819
8.810.2.1 pRXAIG . . . 819
8.810.2.2 pRXComprSlope . . . 819
8.810.2.3 pRXComprThres . . . 819
8.810.2.4 pRXExpSlope . . . 819
8.810.2.5 pRXExpThres . . . 819
8.810.2.6 pRXStaticGain . . . 819
8.811 RXAVCList Struct Reference . . . 819
8.811.1 Detailed Description . . . 819
8.811.2 Field Documentation . . . 819
8.811.2.1 pAVRXAVCHeadroom . . . 820
8.811.2.2 pAVRXAVCSens . . . 820
8.812 rxInfo Struct Reference . . . 820
8.812.1 Detailed Description . . . 820
8.812.2 Field Documentation . . . 820
8.812.2.1 ecio . . . 820
8.812.2.2 isRadioTuned . . . 820
8.812.2.3 phase . . . 821
8.812.2.4 rscp . . . 821
8.812.2.5 rsrp . . . 821
8.812.2.6 rxPower . . . 821
8.813 RXPCMIIRFltr Struct Reference . . . 821
8.813.1 Detailed Description . . . 821
8.813.2 Field Documentation . . . 822
8.813.2.1 pFlag . . . 822
8.813.2.2 pStage0Val . . . 822
8.813.2.3 pStage1Val . . . 822
8.813.2.4 pStage2Val . . . 822
8.813.2.5 pStage3Val . . . 822
8.813.2.6 pStage4Val . . . 822
8.813.2.7 pStageCnt . . . 822
8.814 RxSigInfo Struct Reference . . . 822
8.814.1 Detailed Description . . . 823

8.814.2 Field Documentation . . . 823
8.814.2.1 isRadioTuned . . . 823
8.814.2.2 rsrp . . . 823
8.814.2.3 rxChainIndex . . . 823
8.814.2.4 rxPower . . . 823
8.815 rxSignalStrengthListElement Struct Reference . . . 823
8.815.1 Detailed Description . . . 823
8.815.2 Field Documentation . . . 824
8.815.2.1 radioIf . . . 824
8.815.2.2 rxSignalStrength . . . 824
8.816 sApnExtraParams Struct Reference . . . 824
8.816.1 Detailed Description . . . 824
8.816.2 Field Documentation . . . 825
8.816.2.1 ambr_dl . . . 825
8.816.2.2 ambr_dl_ext . . . 825
8.816.2.3 ambr_dl_ext2 . . . 825
8.816.2.4 ambr_ul . . . 825
8.816.2.5 ambr_ul_ext . . . 825
8.816.2.6 ambr_ul_ext2 . . . 825
8.816.2.7 apnId . . . 825
8.817 satelliteInfo Struct Reference . . . 825
8.817.1 Detailed Description . . . 826
8.817.2 Field Documentation . . . 827
8.817.2.1 azimuth . . . 827
8.817.2.2 elevation . . . 827
8.817.2.3 gnssSvId . . . 827
8.817.2.4 healthStatus . . . 827
8.817.2.5 snr . . . 827
8.817.2.6 svInfoMask . . . 827
8.817.2.7 svListLen . . . 827
8.817.2.8 svStatus . . . 827
8.817.2.9 system . . . 827
8.817.2.10 validMask . . . 827
8.818 SccRxInfo Struct Reference . . . 828
8.818.1 Detailed Description . . . 828
8.818.2 Field Documentation . . . 828
8.818.2.1 numInstances . . . 828
8.818.2.2 rsrq . . . 828
8.818.2.3 sigInfo . . . 828
8.818.2.4 snr . . . 828

8.818.2.5 TlvPresent . . . 828
8.819 sensorData Struct Reference . . . 828
8.819.1 Detailed Description . . . 829
8.819.2 Field Documentation . . . 830
8.819.2.1 flags . . . 830
8.819.2.2 sensorDataLen . . . 830
8.819.2.3 timeOfFirstSample . . . 830
8.819.2.4 timeOffset . . . 830
8.819.2.5 xAxis . . . 830
8.819.2.6 yAxis . . . 830
8.819.2.7 zAxis . . . 830
8.820 sensorData_t Struct Reference . . . 830
8.820.1 Detailed Description . . . 830
8.820.2 Field Documentation . . . 831
8.820.2.1 flags . . . 831
8.820.2.2 sensorDataLen . . . 831
8.820.2.3 timeOfFirstSample . . . 831
8.820.2.4 timeOffset . . . 831
8.820.2.5 xAxis . . . 831
8.820.2.6 yAxis . . . 831
8.820.2.7 zAxis . . . 831
8.821 sensorDataUsage_s Struct Reference . . . 832
8.821.1 Detailed Description . . . 832
8.821.2 Field Documentation . . . 832
8.821.2.1 aidingIndicatorMask . . . 832
8.821.2.2 usageMask . . . 832
8.822 serialNumbersInfo Struct Reference . . . 832
8.822.1 Detailed Description . . . 833
8.822.2 Field Documentation . . . 833
8.822.2.1 esnSize . . . 833
8.822.2.2 imeiSize . . . 833
8.822.2.3 imeiSvnSize . . . 833
8.822.2.4 meidSize . . . 833
8.822.2.5 pESNString . . . 833
8.822.2.6 pIMEIString . . . 833
8.822.2.7 pImeiSvnString . . . 833
8.822.2.8 pMEIDString . . . 833
8.823 serviceProviderName Struct Reference . . . 834
8.823.1 Detailed Description . . . 834
8.823.2 Field Documentation . . . 834

8.823.2.1 displayCondition . . . 834
8.823.2.2 spn . . . 834
8.823.2.3 spnLength . . . 834
8.824 ServingSystemInfo Struct Reference . . . 834
8.824.1 Detailed Description . . . 834
8.824.2 Field Documentation . . . 835
8.824.2.1 csAttachState . . . 835
8.824.2.2 hdrPersonality . . . 835
8.824.2.3 psAttachState . . . 835
8.824.2.4 radioInterfaceList . . . 836
8.824.2.5 radioInterfaceNo . . . 836
8.824.2.6 registrationState . . . 836
8.824.2.7 selectedNetwork . . . 836
8.825 servSystem Struct Reference . . . 836
8.825.1 Detailed Description . . . 836
8.825.2 Field Documentation . . . 837
8.825.2.1 csAttachState . . . 837
8.825.2.2 numRadioInterfaces . . . 837
8.825.2.3 psAttachState . . . 837
8.825.2.4 radioInterface . . . 837
8.825.2.5 regState . . . 837
8.825.2.6 selNetwork . . . 837
8.826 sessionInfo Union Reference . . . 837
8.826.1 Detailed Description . . . 837
8.826.2 Field Documentation . . . 838
8.826.2.1 omaDmConfig . . . 838
8.826.2.2 omaDmFota . . . 838
8.826.2.3 omaDmNotifications . . . 838
8.827 sessionInfoExt Union Reference . . . 838
8.827.1 Detailed Description . . . 838
8.827.2 Field Documentation . . . 838
8.827.2.1 omaDmConfig . . . 838
8.827.2.2 omaDmFota . . . 838
8.828 sessionInfoTlv Struct Reference . . . 838
8.828.1 Detailed Description . . . 838
8.828.2 Field Documentation . . . 838
8.828.2.1 sessionInfo . . . 838
8.828.2.2 sessionType . . . 838
8.828.2.3 TlvPresent . . . 838
8.829 sessionInfoTlvExt Struct Reference . . . 839

8.829.1 Detailed Description . . . 839
8.829.2 Field Documentation . . . 839
8.829.2.1 sessionInfo . . . 839
8.829.2.2 sessionType . . . 839
8.829.2.3 TlvPresent . . . 839
8.830 SetAudioPathConfigReq Struct Reference . . . 839
8.830.1 Detailed Description . . . 839
8.830.2 Field Documentation . . . 840
8.830.2.1 pCodecSTGain . . . 840
8.830.2.2 pDTMFTXGain . . . 841
8.830.2.3 pECMode . . . 841
8.830.2.4 pNSEnable . . . 841
8.830.2.5 Profile . . . 841
8.830.2.6 pRXAGCList . . . 841
8.830.2.7 pRXAVCAGCSwitch . . . 841
8.830.2.8 pRXAVCList . . . 841
8.830.2.9 pRXPCMIIRFltr . . . 841
8.830.2.10 pTXAGCList . . . 841
8.830.2.11 pTXAVCSwitch . . . 841
8.830.2.12 pTXGain . . . 841
8.830.2.13 pTXPCMIIRFltr . . . 841
8.831 SetAudioProfileReq Struct Reference . . . 841
8.831.1 Detailed Description . . . 841
8.831.2 Field Documentation . . . 842
8.831.2.1 EarMute . . . 842
8.831.2.2 Generator . . . 842
8.831.2.3 MicMute . . . 842
8.831.2.4 Profile . . . 842
8.831.2.5 Volume . . . 842
8.832 SetAudioVolTLBConfigReq Struct Reference . . . 842
8.832.1 Detailed Description . . . 842
8.832.2 Field Documentation . . . 843
8.832.2.1 Generator . . . 843
8.832.2.2 Item . . . 843
8.832.2.3 Profile . . . 843
8.832.2.4 Volume . . . 843
8.832.2.5 VolValue . . . 843
8.833 SetAudioVolTLBConfigResp Struct Reference . . . 843
8.833.1 Detailed Description . . . 843
8.833.2 Field Documentation . . . 843

8.833.2.1 ResCode . . . 843
8.834 setCustomSettingV2 Struct Reference . . . 844
8.834.1 Detailed Description . . . 844
8.834.2 Field Documentation . . . 844
8.834.2.1 cust_id . . . 844
8.834.2.2 cust_value . . . 844
8.834.2.3 value_length . . . 844
8.835 setDyingGaspCfg Struct Reference . . . 844
8.835.1 Detailed Description . . . 844
8.835.2 Field Documentation . . . 845
8.835.2.1 pDestSMSContent . . . 845
8.835.2.2 pDestSMSNum . . . 845
8.836 SetIMSSMSConfigReq Struct Reference . . . 845
8.836.1 Detailed Description . . . 845
8.836.2 Field Documentation . . . 845
8.836.2.1 pPhoneCtxtURI . . . 845
8.836.2.2 pPhoneCtxtURILen . . . 845
8.836.2.3 pSMSFormat . . . 845
8.836.2.4 pSMSOverIPNwInd . . . 845
8.837 SetIMSSMSConfigResp Struct Reference . . . 845
8.837.1 Detailed Description . . . 846
8.837.2 Field Documentation . . . 846
8.837.2.1 pSettingResp . . . 846
8.838 SetIMSUserConfigReq Struct Reference . . . 846
8.838.1 Detailed Description . . . 846
8.838.2 Field Documentation . . . 846
8.838.2.1 pIMSDomain . . . 846
8.838.2.2 pIMSDomainLen . . . 846
8.839 SetIMSUserConfigResp Struct Reference . . . 846
8.839.1 Detailed Description . . . 847
8.839.2 Field Documentation . . . 847
8.839.2.1 pSettingResp . . . 847
8.840 SetIMSVoIPConfigReq Struct Reference . . . 847
8.840.1 Detailed Description . . . 847
8.840.2 Field Documentation . . . 849
8.840.2.1 pAmrMode . . . 849
8.840.2.2 pAmrOctetAligned . . . 849
8.840.2.3 pAmrWbEnable . . . 849
8.840.2.4 pAmrWBMode . . . 849
8.840.2.5 pAmrWBOctetAligned . . . 849

8.840.2.6 pMinSessionExpiryTimer . . . 849
8.840.2.7 pRingBackTimer . . . 849
8.840.2.8 pRingingTimer . . . 849
8.840.2.9 pRTPRTCPInactTimer . . . 849
8.840.2.10 pScrAmrEnable . . . 849
8.840.2.11 pScrAmrWbEnable . . . 849
8.840.2.12 pSessionExpiryTimer . . . 849
8.841 SetIMSVoIPConfigResp Struct Reference . . . 849
8.841.1 Detailed Description . . . 849
8.841.2 Field Documentation . . . 850
8.841.2.1 pSettingResp . . . 850
8.842 SetM2MAudioAVCFGReq Struct Reference . . . 850
8.842.1 Detailed Description . . . 850
8.842.2 Field Documentation . . . 850
8.842.2.1 Device . . . 850
8.842.2.2 PIFACEId . . . 850
8.842.2.3 pPCMParams . . . 850
8.842.2.4 Profile . . . 850
8.843 SetM2MAudioLPBKReq Struct Reference . . . 851
8.843.1 Detailed Description . . . 851
8.843.2 Field Documentation . . . 851
8.843.2.1 Enable . . . 851
8.844 SetM2MAudioProfileReq Struct Reference . . . 851
8.844.1 Detailed Description . . . 851
8.844.2 Field Documentation . . . 852
8.844.2.1 pCwtMute . . . 852
8.844.2.2 pEarMute . . . 852
8.844.2.3 pGenerator . . . 852
8.844.2.4 pMicMute . . . 852
8.844.2.5 Profile . . . 852
8.844.2.6 pVolume . . . 852
8.845 SetM2MAudioVolumeReq Struct Reference . . . 852
8.845.1 Detailed Description . . . 852
8.845.2 Field Documentation . . . 853
8.845.2.1 Generator . . . 853
8.845.2.2 Level . . . 853
8.845.2.3 Profile . . . 853
8.846 SetM2MAVMuteReq Struct Reference . . . 853
8.846.1 Detailed Description . . . 853
8.846.2 Field Documentation . . . 853

8.846.2.1 EarMute . . . 853
8.846.2.2 MicMute . . . 853
8.846.2.3 pCwtMute . . . 854
8.846.2.4 Profile . . . 854
8.847 SetM2MSpkrGainReq Struct Reference . . . 854
8.847.1 Detailed Description . . . 854
8.847.2 Field Documentation . . . 854
8.847.2.1 Profile . . . 854
8.847.2.2 Value . . . 854
8.848 setPINProtection Struct Reference . . . 854
8.848.1 Detailed Description . . . 854
8.848.2 Field Documentation . . . 855
8.848.2.1 pinID . . . 855
8.848.2.2 pinLength . . . 855
8.848.2.3 pinOperation . . . 855
8.848.2.4 pinValue . . . 855
8.849 SetRegMgrConfigReq Struct Reference . . . 855
8.849.1 Detailed Description . . . 855
8.849.2 Field Documentation . . . 856
8.849.2.1 pCSCFPortName . . . 856
8.849.2.2 pCSCFPortNameLen . . . 856
8.849.2.3 pIMSTestMode . . . 856
8.849.2.4 pPriCSCFPort . . . 856
8.850 SetRegMgrConfigResp Struct Reference . . . 856
8.850.1 Detailed Description . . . 856
8.850.2 Field Documentation . . . 856
8.850.2.1 pSettingResp . . . 856
8.851 setSignalStrengthInfo Struct Reference . . . 856
8.851.1 Detailed Description . . . 857
8.851.2 Field Documentation . . . 860
8.851.2.1 pCDMAECIODelta . . . 860
8.851.2.2 pCDMAECIOThresh . . . 860
8.851.2.3 pCDMARSSIDelta . . . 860
8.851.2.4 pCDMARSSIThresh . . . 860
8.851.2.5 pGSMRSSIDelta . . . 860
8.851.2.6 pGSMRSSIThresh . . . 860
8.851.2.7 pHDRECIODelta . . . 860
8.851.2.8 pHDRECIOThresh . . . 860
8.851.2.9 pHDRIODelta . . . 860
8.851.2.10 pHDRIOThresh . . . 860

8.851.2.11 pHDRRSSIDelta . . . 860
8.851.2.12 pHDRRSSIThresh . . . 860
8.851.2.13 pHDRSINRDelta . . . 860
8.851.2.14 pHDRSINRThresh . . . 860
8.851.2.15 pLTERSRPDelta . . . 860
8.851.2.16 pLTERSRPThresh . . . 860
8.851.2.17 pLTERSRQDelta . . . 860
8.851.2.18 pLTERSRQThresh . . . 861
8.851.2.19 pLTERSSIDelta . . . 861
8.851.2.20 pLTERSSIThresh . . . 861
8.851.2.21 pLTESigRptConfig . . . 861
8.851.2.22 pLTESNRDelta . . . 861
8.851.2.23 pLTESNRThresh . . . 861
8.851.2.24 pTDSCDMAECIODelta . . . 861
8.851.2.25 pTDSCDMAECIOThresh . . . 861
8.851.2.26 pTDSCDMARSCPDelta . . . 861
8.851.2.27 pTDSCDMARSCPThresh . . . 861
8.851.2.28 pTDSCDMARSSIDelta . . . 861
8.851.2.29 pTDSCDMARSSIThresh . . . 861
8.851.2.30 pTDSCDMASINRDelta . . . 861
8.851.2.31 pTDSCDMASINRThresh . . . 861
8.851.2.32 pWCDMAECIODelta . . . 861
8.851.2.33 pWCDMAECIOThresh . . . 861
8.851.2.34 pWCDMARSSIDelta . . . 861
8.851.2.35 pWCDMARSSIThresh . . . 861
8.852 SetSIPConfigReq Struct Reference . . . 861
8.852.1 Detailed Description . . . 861
8.852.2 Field Documentation . . . 862
8.852.2.1 pSigCompEnabled . . . 862
8.852.2.2 pSIPLocalPort . . . 862
8.852.2.3 pSubscribeTimer . . . 862
8.852.2.4 pTimerSIPReg . . . 862
8.852.2.5 pTimerT1 . . . 862
8.852.2.6 pTimerT2 . . . 862
8.852.2.7 pTimerTf . . . 862
8.853 SetSIPConfigResp Struct Reference . . . 862
8.853.1 Detailed Description . . . 862
8.853.2 Field Documentation . . . 863
8.853.2.1 pSettingResp . . . 863
8.854 sGetDeviceSeriesResult Struct Reference . . . 863

8.854.1 Detailed Description . . . 863
8.854.2 Field Documentation . . . 863
8.854.2.1 eDevice . . . 863
8.854.2.2 uResult . . . 863
8.855 sidNid Struct Reference . . . 863
8.855.1 Detailed Description . . . 863
8.855.2 Field Documentation . . . 864
8.855.2.1 nid . . . 864
8.855.2.2 sid . . . 864
8.856 sigInfo Struct Reference . . . 864
8.856.1 Detailed Description . . . 864
8.856.2 Field Documentation . . . 865
8.856.2.1 pECIOThresh . . . 865
8.856.2.2 pHDRSINRThresh . . . 865
8.856.2.3 pIOThresh . . . 865
8.856.2.4 pLTESigRptCfg . . . 865
8.856.2.5 pLTESNRThresh . . . 865
8.856.2.6 pRSRPThresh . . . 865
8.856.2.7 pRSRQThresh . . . 865
8.856.2.8 pRSSIThresh . . . 865
8.856.2.9 pTDSCDMASINRCONFThresh . . . 865
8.857 signalInfo Struct Reference . . . 865
8.857.1 Detailed Description . . . 865
8.857.2 Field Documentation . . . 866
8.857.2.1 alertPitch . . . 866
8.857.2.2 signal . . . 866
8.857.2.3 signalType . . . 866
8.858 SignalStrengthDataType Struct Reference . . . 866
8.858.1 Field Documentation . . . 866
8.858.1.1 thresholds . . . 866
8.858.1.2 thresholdsSize . . . 866
8.859 slot_t Struct Reference . . . 866
8.859.1 Detailed Description . . . 866
8.859.2 Field Documentation . . . 867
8.859.2.1 bICCID . . . 867
8.859.2.2 bICCIDLength . . . 867
8.859.2.3 bLogicalSlot . . . 867
8.859.2.4 uPhyCardStatus . . . 867
8.859.2.5 uPhySlotStatus . . . 867
8.860 slotInf Struct Reference . . . 867

8.860.1 Detailed Description . . . 867
8.860.2 Field Documentation . . . 868
8.860.2.1 AppStatus . . . 868
8.860.2.2 cardState . . . 868
8.860.2.3 errorState . . . 868
8.860.2.4 numApp . . . 868
8.860.2.5 upinRetries . . . 869
8.860.2.6 upinState . . . 869
8.860.2.7 upukRetries . . . 869
8.861 slotInfo Struct Reference . . . 869
8.861.1 Detailed Description . . . 869
8.861.2 Field Documentation . . . 870
8.861.2.1 AppStatus . . . 870
8.861.2.2 cardState . . . 870
8.861.2.3 errorState . . . 870
8.861.2.4 numApp . . . 870
8.861.2.5 upinRetries . . . 870
8.861.2.6 upinState . . . 870
8.861.2.7 upukRetries . . . 870
8.862 slots_t Struct Reference . . . 870
8.862.1 Field Documentation . . . 870
8.862.1.1 uimSlotStatus . . . 870
8.863 slqsautoconnect Struct Reference . . . 870
8.863.1 Detailed Description . . . 871
8.863.2 Field Documentation . . . 871
8.863.2.1 acroamsetting . . . 871
8.863.2.2 acsetting . . . 871
8.863.2.3 action . . . 871
8.864 SLQSDeleteProfileParams Struct Reference . . . 871
8.864.1 Detailed Description . . . 871
8.864.2 Field Documentation . . . 872
8.864.2.1 profileIndex . . . 872
8.864.2.2 profileType . . . 872
8.865 slqsfwinfo_s Struct Reference . . . 872
8.865.1 Detailed Description . . . 872
8.865.2 Field Documentation . . . 873
8.865.2.1 appversion_str . . . 873
8.865.2.2 bootversion_str . . . 873
8.865.2.3 carrier_str . . . 873
8.865.2.4 cur_carr_name . . . 873

8.865.2.5 cur_carr_rev . . . 873
8.865.2.6 modelid_str . . . 873
8.865.2.7 packageid_str . . . 873
8.865.2.8 priversion_str . . . 873
8.865.2.9 sku_str . . . 873
8.866 SlqsNas3GppNetworkInfo Struct Reference . . . 873
8.866.1 Detailed Description . . . 873
8.866.2 Field Documentation . . . 874
8.866.2.1 Description . . . 874
8.866.2.2 Forbidden . . . 874
8.866.2.3 InUse . . . 874
8.866.2.4 MCC . . . 874
8.866.2.5 MNC . . . 874
8.866.2.6 Preferred . . . 874
8.866.2.7 Roaming . . . 874
8.867 SlqsNasPcsDigit Struct Reference . . . 874
8.867.1 Detailed Description . . . 875
8.867.2 Field Documentation . . . 875
8.867.2.1 includes_pcs_digit . . . 875
8.867.2.2 MCC . . . 875
8.867.2.3 MNC . . . 875
8.868 slqssendasyncsmsparams_s Struct Reference . . . 875
8.868.1 Detailed Description . . . 875
8.868.2 Field Documentation . . . 877
8.868.2.1 messageFormat . . . 877
8.868.2.2 messageSize . . . 877
8.868.2.3 pFollowOnDC . . . 877
8.868.2.4 pForceOnDC . . . 877
8.868.2.5 pLinktimer . . . 877
8.868.2.6 pMessage . . . 877
8.868.2.7 pRetryMessage . . . 877
8.868.2.8 pRetryMessageId . . . 877
8.868.2.9 pServiceOption . . . 877
8.868.2.10 pSmsOnIms . . . 877
8.868.2.11 pUserData . . . 877
8.869 slqssendsmsparams_s Struct Reference . . . 877
8.869.1 Detailed Description . . . 877
8.869.2 Field Documentation . . . 878
8.869.2.1 messageFailureCode . . . 878
8.869.2.2 messageFormat . . . 878

8.869.2.3 messageID . . . 878
8.869.2.4 messageSize . . . 878
8.869.2.5 pLinktimer . . . 878
8.869.2.6 pMessage . . . 878
8.869.2.7 pSmsOnIms . . . 878
8.870 slqsSessionStateInfo Struct Reference . . . 878
8.870.1 Detailed Description . . . 879
8.870.2 Field Documentation . . . 879
8.870.2.1 pQmiInterfaceInfo . . . 879
8.870.2.2 reconfiguration_required . . . 879
8.870.2.3 sessionEndReason . . . 879
8.870.2.4 state . . . 879
8.871 slqsSignalStrengthInfo Struct Reference . . . 879
8.871.1 Detailed Description . . . 880
8.871.2 Field Documentation . . . 881
8.871.2.1 ecioList . . . 881
8.871.2.2 ecioListLen . . . 881
8.871.2.3 errorRateList . . . 881
8.871.2.4 errorRateListLen . . . 881
8.871.2.5 Io . . . 881
8.871.2.6 ltersrp . . . 881
8.871.2.7 ltesnr . . . 881
8.871.2.8 rsrqInfo . . . 881
8.871.2.9 rxSignalStrengthList . . . 881
8.871.2.10 rxSignalStrengthListLen . . . 881
8.871.2.11 signalStrengthReqMask . . . 882
8.871.2.12 sinr . . . 882
8.872 SLQSSignalStrengthsIndReq Struct Reference . . . 882
8.872.1 Detailed Description . . . 882
8.872.2 Field Documentation . . . 883
8.872.2.1 ecioDelta . . . 883
8.872.2.2 ecioThresholdList . . . 883
8.872.2.3 ecioThresholdListLen . . . 883
8.872.2.4 ioDelta . . . 883
8.872.2.5 lteRsrpDelta . . . 883
8.872.2.6 lteSnrDelta . . . 883
8.872.2.7 rsrqDelta . . . 883
8.872.2.8 rxSignalStrengthDelta . . . 883
8.872.2.9 sinrDelta . . . 883
8.872.2.10 sinrThresholdList . . . 883

8.872.2.11 sinrThresholdListLen . . . 883
8.873 SLQSSignalStrengthsInformation Struct Reference . . . 884
8.873.1 Detailed Description . . . 884
8.873.2 Field Documentation . . . 884
8.873.2.1 ecioInfo . . . 885
8.873.2.2 errorRateInfo . . . 885
8.873.2.3 io . . . 885
8.873.2.4 lteRsrpinfo . . . 885
8.873.2.5 lteSnrinfo . . . 885
8.873.2.6 rsrqInfo . . . 885
8.873.2.7 rxSignalStrengthInfo . . . 885
8.873.2.8 sinr . . . 885
8.874 slqsWdsEventInfo Struct Reference . . . 885
8.874.1 Detailed Description . . . 885
8.874.2 Field Documentation . . . 886
8.874.2.1 pDataBearer . . . 886
8.874.2.2 pDormancyStatus . . . 886
8.874.2.3 pPacketsCountRX . . . 886
8.874.2.4 pPacketsCountTX . . . 886
8.874.2.5 pQmiInterfaceInfo . . . 887
8.874.2.6 pTotalBytesRX . . . 887
8.874.2.7 pTotalBytesTX . . . 887
8.875 sms_BroadcastConfig Struct Reference . . . 887
8.875.1 Detailed Description . . . 887
8.875.2 Field Documentation . . . 887
8.875.2.1 fromServiceId . . . 887
8.875.2.2 selected . . . 887
8.875.2.3 toServiceId . . . 887
8.876 sms_CDMABroadcastConfig Struct Reference . . . 887
8.876.1 Detailed Description . . . 887
8.876.2 Field Documentation . . . 888
8.876.2.1 language . . . 888
8.876.2.2 selected . . . 888
8.876.2.3 serviceCategory . . . 888
8.877 sms_getIndicationReg Struct Reference . . . 888
8.877.1 Detailed Description . . . 888
8.877.2 Field Documentation . . . 889
8.877.2.1 pRegCallStatInfoEvt . . . 889
8.877.2.2 pRegTransLayerInfoEvt . . . 889
8.877.2.3 pRegTransNWRegInfoEvt . . . 889

8.878 sms_getMsgWaitingInfo Struct Reference . . . 889
8.878.1 Detailed Description . . . 889
8.878.2 Field Documentation . . . 889
8.878.2.1 msgWaitInfo . . . 889
8.878.2.2 numInstances . . . 889
8.879 sms_getTransLayerInfo Struct Reference . . . 889
8.879.1 Detailed Description . . . 890
8.879.2 Field Documentation . . . 890
8.879.2.1 pRegInd . . . 890
8.879.2.2 pTransLayerInfo . . . 890
8.880 sms_getTransNWRegInfo Struct Reference . . . 890
8.880.1 Detailed Description . . . 890
8.880.2 Field Documentation . . . 890
8.880.2.1 pRegStatus . . . 891
8.881 sms_maxStorageSizeReq Struct Reference . . . 891
8.881.1 Detailed Description . . . 891
8.881.2 Field Documentation . . . 891
8.881.2.1 pMessageMode . . . 891
8.881.2.2 storageType . . . 891
8.882 sms_maxStorageSizeResp Struct Reference . . . 891
8.882.1 Detailed Description . . . 891
8.882.2 Field Documentation . . . 892
8.882.2.1 freeSlots . . . 892
8.882.2.2 maxStorageSize . . . 892
8.883 sms_messageWaitingInfoContent Struct Reference . . . 892
8.883.1 Detailed Description . . . 892
8.883.2 Field Documentation . . . 892
8.883.2.1 activeInd . . . 892
8.883.2.2 msgCount . . . 892
8.883.2.3 msgType . . . 892
8.884 sms_msgProtocolResp Struct Reference . . . 893
8.884.1 Detailed Description . . . 893
8.884.2 Field Documentation . . . 893
8.884.2.1 msgProtocol . . . 893
8.885 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . 893
8.885.1 Detailed Description . . . 893
8.885.2 Field Documentation . . . 894
8.885.2.1 activated_ind . . . 894
8.885.2.2 CDMABroadcastConfig . . . 894
8.885.2.3 num_instances . . . 894

8.886 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference . . . 894
8.886.1 Detailed Description . . . 894
8.886.2 Field Documentation . . . 894
8.886.2.1 activated_ind . . . 894
8.886.2.2 broadcastConfig . . . 894
8.886.2.3 num_instances . . . 894
8.887 sms_routeEntry Struct Reference . . . 894
8.887.1 Detailed Description . . . 895
8.887.2 Field Documentation . . . 895
8.887.2.1 messageClass . . . 895
8.887.2.2 messageType . . . 895
8.887.2.3 receiptAction . . . 895
8.887.2.4 routeStorage . . . 895
8.888 sms_sendAsyncsmsParams Struct Reference . . . 895
8.888.1 Detailed Description . . . 896
8.888.2 Field Documentation . . . 897
8.888.2.1 messageFormat . . . 897
8.888.2.2 messageSize . . . 897
8.888.2.3 pFollowOnDC . . . 897
8.888.2.4 pForceOnDC . . . 897
8.888.2.5 pLinktimer . . . 897
8.888.2.6 pMessage . . . 897
8.888.2.7 pRetryMessage . . . 897
8.888.2.8 pRetryMessageId . . . 897
8.888.2.9 pServiceOption . . . 897
8.888.2.10 pSmsOnIms . . . 897
8.888.2.11 pUserData . . . 897
8.889 sms_setIndicationReg Struct Reference . . . 898
8.889.1 Detailed Description . . . 898
8.889.2 Field Documentation . . . 898
8.889.2.1 pRegCallStatInfoEvt . . . 898
8.889.2.2 pRegTransLayerInfoEvt . . . 898
8.889.2.3 pRegTransNWRegInfoEvt . . . 898
8.890 sms_setRoutesReq Struct Reference . . . 898
8.890.1 Detailed Description . . . 899
8.890.2 Field Documentation . . . 899
8.890.2.1 numOfRoutes . . . 899
8.890.2.2 pTransferStatusReport . . . 899
8.890.2.3 routeList . . . 899
8.891 sms_transLayerInfo Struct Reference . . . 899

8.891.1 Detailed Description . . . 899
8.891.2 Field Documentation . . . 899
8.891.2.1 TransCap . . . 900
8.891.2.2 TransType . . . 900
8.892 SMSAsyncRawSend_s Struct Reference . . . 900
8.892.1 Detailed Description . . . 900
8.892.2 Field Documentation . . . 901
8.892.2.1 alphaIDLen . . . 901
8.892.2.2 causeCode . . . 901
8.892.2.3 errorClass . . . 901
8.892.2.4 messageID . . . 901
8.892.2.5 msgDelFailureCause . . . 901
8.892.2.6 msgDelFailureType . . . 901
8.892.2.7 pAlphaID . . . 901
8.892.2.8 RPCause . . . 901
8.892.2.9 sendStatus . . . 901
8.892.2.10 TPCause . . . 901
8.892.2.11 userData . . . 901
8.893 SMSCAddress Struct Reference . . . 901
8.893.1 Detailed Description . . . 902
8.893.2 Field Documentation . . . 902
8.893.2.1 data . . . 902
8.893.2.2 length . . . 902
8.894 sMSCAddress Struct Reference . . . 902
8.894.1 Detailed Description . . . 902
8.894.2 Field Documentation . . . 902
8.894.2.1 data . . . 902
8.894.2.2 length . . . 902
8.895 sMSCAddressTlv Struct Reference . . . 902
8.895.1 Detailed Description . . . 902
8.895.2 Field Documentation . . . 903
8.895.2.1 SMSCInfo . . . 903
8.895.2.2 TlvPresent . . . 903
8.896 sMSEtwsMessage Struct Reference . . . 903
8.896.1 Detailed Description . . . 903
8.896.2 Field Documentation . . . 903
8.896.2.1 data . . . 903
8.896.2.2 length . . . 903
8.896.2.3 notificationType . . . 903
8.897 SMSEtwsMessage Struct Reference . . . 903

8.897.1 Detailed Description . . . 904
8.897.2 Field Documentation . . . 904
8.897.2.1 data . . . 904
8.897.2.2 length . . . 904
8.897.2.3 notificationType . . . 904
8.898 sMSEtwsMessageTlv Struct Reference . . . 904
8.898.1 Detailed Description . . . 904
8.898.2 Field Documentation . . . 904
8.898.2.1 EtwsMessageInfo . . . 904
8.898.2.2 TlvPresent . . . 904
8.899 SMSEtwsPlmn Struct Reference . . . 904
8.899.1 Detailed Description . . . 905
8.899.2 Field Documentation . . . 905
8.899.2.1 mobileCountryCode . . . 905
8.899.2.2 mobileNetworkCode . . . 905
8.900 sMSEtwsPlmn Struct Reference . . . 905
8.900.1 Detailed Description . . . 905
8.900.2 Field Documentation . . . 905
8.900.2.1 mobileCountryCode . . . 905
8.900.2.2 mobileNetworkCode . . . 905
8.901 SMSEventInfo_s Struct Reference . . . 905
8.901.1 Detailed Description . . . 906
8.901.2 Field Documentation . . . 906
8.901.2.1 pEtwsMessageInfo . . . 906
8.901.2.2 pEtwsPlmnInfo . . . 906
8.901.2.3 pMessageModeInfo . . . 907
8.901.2.4 pMTMessageInfo . . . 907
8.901.2.5 pSMSCAddressInfo . . . 907
8.901.2.6 pSMSOnIMSInfo . . . 907
8.901.2.7 pTransferRouteMTMessageInfo . . . 907
8.901.2.8 smsEventType . . . 907
8.902 smsMaxStorageSizeReq Struct Reference . . . 907
8.902.1 Detailed Description . . . 907
8.902.2 Field Documentation . . . 907
8.902.2.1 pMessageMode . . . 907
8.902.2.2 storageType . . . 907
8.903 smsMaxStorageSizeResp Struct Reference . . . 907
8.903.1 Detailed Description . . . 908
8.903.2 Field Documentation . . . 908
8.903.2.1 freeSlots . . . 908

8.903.2.2 maxStorageSize . . . 908
8.904 SMSMemoryInfo Struct Reference . . . 908
8.904.1 Detailed Description . . . 908
8.904.2 Field Documentation . . . 908
8.904.2.1 messageMode . . . 908
8.904.2.2 storageType . . . 908
8.905 SMSMessageMode Struct Reference . . . 908
8.905.1 Detailed Description . . . 909
8.905.2 Field Documentation . . . 909
8.905.2.1 messageMode . . . 909
8.906 sMSMessageMode Struct Reference . . . 909
8.906.1 Detailed Description . . . 909
8.906.2 Field Documentation . . . 909
8.906.2.1 messageMode . . . 909
8.907 smsMsgprotocolResp Struct Reference . . . 909
8.907.1 Detailed Description . . . 909
8.907.2 Field Documentation . . . 910
8.907.2.1 msgProtocol . . . 910
8.908 SMSMTMessage Struct Reference . . . 910
8.908.1 Detailed Description . . . 910
8.908.2 Field Documentation . . . 910
8.908.2.1 messageIndex . . . 910
8.908.2.2 storageType . . . 910
8.909 sMSMTMessage Struct Reference . . . 910
8.909.1 Detailed Description . . . 910
8.909.2 Field Documentation . . . 910
8.909.2.1 messageIndex . . . 910
8.909.2.2 storageType . . . 910
8.910 sMSOnIMS Struct Reference . . . 911
8.910.1 Detailed Description . . . 911
8.910.2 Field Documentation . . . 911
8.910.2.1 smsOnIMS . . . 911
8.911 SMSOnIMS Struct Reference . . . 911
8.911.1 Detailed Description . . . 911
8.911.2 Field Documentation . . . 911
8.911.2.1 smsOnIMS . . . 911
8.912 sMSOnIMSTlv Struct Reference . . . 911
8.912.1 Detailed Description . . . 911
8.912.2 Field Documentation . . . 912
8.912.2.1 IMSInfo . . . 912

8.912.2.2 TlvPresent . . . 912
8.913 smsRouteEntry Struct Reference . . . 912
8.913.1 Detailed Description . . . 912
8.913.2 Field Documentation . . . 913
8.913.2.1 messageClass . . . 913
8.913.2.2 messageType . . . 913
8.913.2.3 receiptAction . . . 913
8.913.2.4 routeStorage . . . 913
8.914 smsSetRoutesReq Struct Reference . . . 913
8.914.1 Detailed Description . . . 913
8.914.2 Field Documentation . . . 913
8.914.2.1 numOfRoutes . . . 913
8.914.2.2 pTransferStatusReport . . . 913
8.914.2.3 routeList . . . 913
8.915 SMSTransferRouteMTMessage Struct Reference . . . 914
8.915.1 Detailed Description . . . 914
8.915.2 Field Documentation . . . 914
8.915.2.1 ackIndicator . . . 914
8.915.2.2 data . . . 914
8.915.2.3 format . . . 914
8.915.2.4 length . . . 914
8.915.2.5 transactionID . . . 914
8.916 sMSTransferRouteMTMessage Struct Reference . . . 914
8.916.1 Detailed Description . . . 915
8.916.2 Field Documentation . . . 915
8.916.2.1 ackIndicator . . . 915
8.916.2.2 data . . . 915
8.916.2.3 format . . . 915
8.916.2.4 length . . . 915
8.916.2.5 transactionID . . . 915
8.917 sQosFlowStat Struct Reference . . . 915
8.917.1 Detailed Description . . . 915
8.917.2 Field Documentation . . . 916
8.917.2.1 bearerId . . . 916
8.917.2.2 tx_bytes . . . 916
8.917.2.3 tx_bytes_drp . . . 916
8.917.2.4 tx_pkt . . . 916
8.917.2.5 tx_pkt_drp . . . 916
8.918 sQosStat Struct Reference . . . 916
8.918.1 Detailed Description . . . 916

8.918.2 Field Documentation . . . 917
8.918.2.1 apnId . . . 917
8.918.2.2 numQosFlow . . . 917
8.918.2.3 qosFlow . . . 917
8.918.2.4 total_rx_bytes . . . 917
8.918.2.5 total_rx_pkt . . . 917
8.918.2.6 total_tx_bytes . . . 917
8.918.2.7 total_tx_bytes_drp . . . 917
8.918.2.8 total_tx_pkt . . . 917
8.918.2.9 total_tx_pkt_drp . . . 917
8.919 SrvStatusInfo Struct Reference . . . 917
8.919.1 Detailed Description . . . 918
8.919.2 Field Documentation . . . 918
8.919.2.1 isPrefDataPath . . . 918
8.919.2.2 srvStatus . . . 918
8.920 ssdatasession_params Struct Reference . . . 918
8.920.1 Detailed Description . . . 919
8.920.2 Field Documentation . . . 920
8.920.2.1 action . . . 920
8.920.2.2 failureReason . . . 920
8.920.2.3 failureReasonv4 . . . 920
8.920.2.4 failureReasonv6 . . . 920
8.920.2.5 instanceId . . . 920
8.920.2.6 ipfamily . . . 920
8.920.2.7 pAuthentication . . . 920
8.920.2.8 pPassword . . . 920
8.920.2.9 pProfileId3GPP . . . 920
8.920.2.10 pProfileId3GPP2 . . . 920
8.920.2.11 pTechnology . . . 921
8.920.2.12 pUsername . . . 921
8.920.2.13 rcv4 . . . 921
8.920.2.14 rcv6 . . . 921
8.920.2.15 sessionId . . . 921
8.920.2.16 v4sessionId . . . 921
8.920.2.17 v6sessionId . . . 921
8.920.2.18 verbFailReason . . . 921
8.920.2.19 verbFailReasonType . . . 921
8.921 SupportedMsgList Struct Reference . . . 921
8.921.1 Detailed Description . . . 921
8.921.2 Field Documentation . . . 921

8.921.2.1 supportedMsgLen . . . 921
8.921.2.2 supportedMsgs . . . 921
8.922 SUPSInfo Struct Reference . . . 921
8.922.1 Detailed Description . . . 922
8.922.2 Field Documentation . . . 922
8.922.2.1 isModByCC . . . 922
8.922.2.2 svcType . . . 922
8.923 SV Struct Reference . . . 922
8.923.1 Detailed Description . . . 922
8.923.2 Field Documentation . . . 923
8.923.2.1 id . . . 923
8.923.2.2 mask . . . 923
8.923.2.3 system . . . 923
8.924 SVInfo Struct Reference . . . 923
8.924.1 Detailed Description . . . 923
8.924.2 Field Documentation . . . 923
8.924.2.1 len . . . 924
8.924.2.2 pSV . . . 924
8.925 svUsedforFix_s Struct Reference . . . 924
8.925.1 Detailed Description . . . 924
8.925.2 Field Documentation . . . 924
8.925.2.1 gnssSvUsedList . . . 924
8.925.2.2 gnssSvUsedList_len . . . 924
8.926 SWI_STRUCT_CarrierImage Struct Reference . . . 924
8.926.1 Detailed Description . . . 924
8.926.2 Field Documentation . . . 925
8.926.2.1 m_FwBuildId . . . 925
8.926.2.2 m_FwImageId . . . 925
8.926.2.3 m_nCarrierId . . . 925
8.926.2.4 m_nFolderId . . . 925
8.926.2.5 m_nStorage . . . 925
8.926.2.6 m_PriBuildId . . . 925
8.926.2.7 m_PriImageId . . . 925
8.927 SwiLocGetAutoStartResp Struct Reference . . . 925
8.927.1 Detailed Description . . . 926
8.927.2 Field Documentation . . . 927
8.927.2.1 fix_rate . . . 927
8.927.2.2 fix_rate_reported . . . 927
8.927.2.3 fix_type . . . 927
8.927.2.4 fix_type_reported . . . 927

8.927.2.5 function . . . 927
8.927.2.6 function_reported . . . 927
8.927.2.7 max_dist . . . 927
8.927.2.8 max_dist_reported . . . 927
8.927.2.9 max_time . . . 927
8.927.2.10 max_time_reported . . . 927
8.928 SwiLocSetAutoStartReq Struct Reference . . . 927
8.928.1 Detailed Description . . . 927
8.928.2 Field Documentation . . . 928
8.928.2.1 fix_rate . . . 928
8.928.2.2 fix_type . . . 928
8.928.2.3 function . . . 928
8.928.2.4 max_dist . . . 928
8.928.2.5 max_time . . . 928
8.928.2.6 set_fix_rate . . . 929
8.928.2.7 set_fix_type . . . 929
8.928.2.8 set_function . . . 929
8.928.2.9 set_max_dist . . . 929
8.928.2.10 set_max_time . . . 929
8.929 swiModemStatusResp Struct Reference . . . 929
8.929.1 Detailed Description . . . 929
8.929.2 Field Documentation . . . 929
8.929.2.1 commonInfo . . . 929
8.929.2.2 pLTEInfo . . . 929
8.930 SwiOTAMsg_s Struct Reference . . . 929
8.930.1 Detailed Description . . . 929
8.930.2 Field Documentation . . . 930
8.930.2.1 data . . . 930
8.930.2.2 data_len . . . 930
8.930.2.3 pLteNasRelInfo . . . 930
8.930.2.4 pTime . . . 930
8.930.2.5 type . . . 930
8.931 swiPDPRuntimeSettingsReq Struct Reference . . . 930
8.931.1 Detailed Description . . . 930
8.931.2 Field Documentation . . . 931
8.931.2.1 contextId . . . 931
8.931.2.2 contextType . . . 931
8.932 swiPDPRuntimeSettingsResp Struct Reference . . . 931
8.932.1 Detailed Description . . . 931
8.932.2 Field Documentation . . . 932

8.932.2.1 pAPNName . . . 933
8.932.2.2 pBearerId . . . 933
8.932.2.3 pContextId . . . 933
8.932.2.4 pIPv4Address . . . 933
8.932.2.5 pIPv4GWAddress . . . 933
8.932.2.6 pIPv6Address . . . 933
8.932.2.7 pIPv6GWAddress . . . 933
8.932.2.8 pPrDNSIPv4Address . . . 933
8.932.2.9 pPrDNSIPv6Address . . . 933
8.932.2.10 pPrPCSCFIPv4Address . . . 933
8.932.2.11 pPrPCSCFIPv6Address . . . 933
8.932.2.12 pSeDNSIPv4Address . . . 933
8.932.2.13 pSeDNSIPv6Address . . . 933
8.932.2.14 pSePCSCFIPv4Address . . . 933
8.932.2.15 pSePCSCFIPv6Address . . . 933
8.933 swiQosFilter Struct Reference . . . 933
8.933.1 Detailed Description . . . 934
8.933.2 Field Documentation . . . 935
8.933.2.1 index . . . 935
8.933.2.2 pEspSpi . . . 935
8.933.2.3 pId . . . 935
8.933.2.4 pIPv4DstAddr . . . 935
8.933.2.5 pIPv4SrcAddr . . . 935
8.933.2.6 pIPv6DstAddr . . . 935
8.933.2.7 pIPv6Label . . . 935
8.933.2.8 pIPv6SrcAddr . . . 935
8.933.2.9 pIPv6TrafCls . . . 935
8.933.2.10 pNxtHdrProto . . . 935
8.933.2.11 pPrecedence . . . 935
8.933.2.12 pTCPDstPort . . . 935
8.933.2.13 pTCPSrcPort . . . 935
8.933.2.14 pTos . . . 935
8.933.2.15 pTranDstPort . . . 935
8.933.2.16 pTranSrcPort . . . 935
8.933.2.17 pUDPDstPort . . . 935
8.933.2.18 pUDPSrcPort . . . 935
8.933.2.19 version . . . 936
8.934 swiQosFlow Struct Reference . . . 936
8.934.1 Detailed Description . . . 936
8.934.2 Field Documentation . . . 938

8.934.2.1 index . . . 938
8.934.2.2 p3GPP2Pri . . . 938
8.934.2.3 p3GPPImCn . . . 938
8.934.2.4 p3GPPResResidualBER . . . 938
8.934.2.5 p3GPPSigInd . . . 938
8.934.2.6 p3GPPTraHdlPri . . . 938
8.934.2.7 pDataRate . . . 938
8.934.2.8 pJitter . . . 938
8.934.2.9 pLatency . . . 938
8.934.2.10 pLteQci . . . 938
8.934.2.11 pMaxAllowedPktSz . . . 939
8.934.2.12 pMinPolicedPktSz . . . 939
8.934.2.13 pPktErrRate . . . 939
8.934.2.14 pProfileId3GPP2 . . . 939
8.934.2.15 pTokenBucket . . . 939
8.934.2.16 pTrafficClass . . . 939
8.935 swiQosGranted Struct Reference . . . 939
8.935.1 Detailed Description . . . 939
8.935.2 Field Documentation . . . 939
8.935.2.1 pRxFlow . . . 939
8.935.2.2 pTxFlow . . . 939
8.936 swiQosIds Struct Reference . . . 939
8.936.1 Detailed Description . . . 939
8.936.2 Field Documentation . . . 939
8.936.2.1 pIds . . . 940
8.936.2.2 sz . . . 940
8.937 swiQosModifyReq Struct Reference . . . 940
8.937.1 Detailed Description . . . 940
8.937.2 Field Documentation . . . 940
8.937.2.1 id . . . 940
8.937.2.2 pRxFilter . . . 940
8.937.2.3 pRxFlow . . . 940
8.937.2.4 pTxFilter . . . 940
8.937.2.5 pTxFlow . . . 940
8.938 swiQosReq Struct Reference . . . 940
8.938.1 Detailed Description . . . 940
8.938.2 Field Documentation . . . 941
8.938.2.1 index . . . 941
8.938.2.2 pRxFilter . . . 941
8.938.2.3 pRxFlow . . . 941

8.938.2.4 pTxFilter . . . 941
8.938.2.5 pTxFlow . . . 941
8.939 swiRMTrasnferStaticsReq Struct Reference . . . 941
8.939.1 Detailed Description . . . 941
8.939.2 Field Documentation . . . 942
8.939.2.1 bResetStatistics . . . 942
8.939.2.2 ulMask . . . 942
8.940 sysInfoCommon Struct Reference . . . 942
8.940.1 Detailed Description . . . 942
8.940.2 Field Documentation . . . 944
8.940.2.1 isSysForbidden . . . 944
8.940.2.2 isSysForbiddenValid . . . 944
8.940.2.3 roamStatus . . . 944
8.940.2.4 roamStatusValid . . . 944
8.940.2.5 srvCapability . . . 944
8.940.2.6 srvCapabilityValid . . . 944
8.940.2.7 srvDomain . . . 944
8.940.2.8 srvDomainValid . . . 944
8.941 t_gpsTime Struct Reference . . . 944
8.941.1 Field Documentation . . . 944
8.941.1.1 gpsTimeOfWeekMs . . . 944
8.941.1.2 gpsWeek . . . 944
8.942 t_sensor Struct Reference . . . 944
8.942.1 Field Documentation . . . 944
8.942.1.1 aidingIndicatorMask . . . 944
8.942.1.2 usageMask . . . 944
8.943 t_Sv Struct Reference . . . 945
8.943.1 Field Documentation . . . 945
8.943.1.1 entries . . . 945
8.943.1.2 len . . . 945
8.944 TDSCDMAECIOThresh Struct Reference . . . 945
8.944.1 Detailed Description . . . 945
8.944.2 Field Documentation . . . 945
8.944.2.1 pTDSCDMAECIOThreshList . . . 945
8.944.2.2 TDSCDMAECIOThreshListLen . . . 945
8.945 TDSCDMARSCPThresh Struct Reference . . . 945
8.945.1 Detailed Description . . . 945
8.945.2 Field Documentation . . . 946
8.945.2.1 pTDSCDMARSCPThreshList . . . 946
8.945.2.2 TDSCDMARSCPThreshListLen . . . 946

8.946TDSCDMARSSIThresh Struct Reference . . . 946
8.946.1 Detailed Description . . . 946
8.946.2 Field Documentation . . . 946
8.946.2.1 pTDSCDMARSSIThreshList . . . 946
8.946.2.2 TDSCDMARSSIThreshListLen . . . 946
8.947tdscdmaSigInfoExt Struct Reference . . . 946
8.947.1 Detailed Description . . . 947
8.947.2 Field Documentation . . . 947
8.947.2.1 ecio . . . 947
8.947.2.2 rscp . . . 947
8.947.2.3 rssi . . . 947
8.947.2.4 sinr . . . 947
8.948TDSCDMASigInfoExt Struct Reference . . . 947
8.948.1 Detailed Description . . . 947
8.948.2 Field Documentation . . . 948
8.948.2.1 ecio . . . 948
8.948.2.2 rscp . . . 948
8.948.2.3 rssi . . . 948
8.948.2.4 sinr . . . 948
8.949TDSCDMASINRCONFThresh Struct Reference . . . 948
8.949.1 Detailed Description . . . 948
8.949.2 Field Documentation . . . 948
8.949.2.1 pTDSCDMASINRCONFThreshList . . . 948
8.949.2.2 TDSCDMASINRCONFThreshListLen . . . 948
8.950TDSCDMASINRThresh Struct Reference . . . 948
8.950.1 Detailed Description . . . 948
8.950.2 Field Documentation . . . 949
8.950.2.1 pTDSCDMASINRThreshList . . . 949
8.950.2.2 TDSCDMASINRThreshListLen . . . 949
8.951tempData_t Struct Reference . . . 949
8.951.1 Detailed Description . . . 949
8.951.2 Field Documentation . . . 950
8.951.2.1 temperature . . . 950
8.951.2.2 temperatureDataLen . . . 950
8.951.2.3 timeOfFirstSample . . . 950
8.951.2.4 timeOffset . . . 950
8.951.2.5 timeSource . . . 950
8.952tempratureData Struct Reference . . . 950
8.952.1 Detailed Description . . . 950
8.952.2 Field Documentation . . . 951

8.952.2.1 temperature . . . 951
8.952.2.2 temperatureDataLen . . . 951
8.952.2.3 timeOfFirstSample . . . 951
8.952.2.4 timeOffset . . . 951
8.952.2.5 timeSource . . . 951
8.953 TFTIDParams Struct Reference . . . 951
8.953.1 Detailed Description . . . 951
8.953.2 Field Documentation . . . 952
8.953.2.1 destPortRangeEnd . . . 952
8.953.2.2 destPortRangeStart . . . 952
8.953.2.3 eValid . . . 952
8.953.2.4 filterId . . . 953
8.953.2.5 flowLabel . . . 953
8.953.2.6 IPSECSPI . . . 953
8.953.2.7 ipVersion . . . 953
8.953.2.8 nextHeader . . . 953
8.953.2.9 pSourceIP . . . 953
8.953.2.10 sourceIPMask . . . 953
8.953.2.11 srcPortRangeEnd . . . 953
8.953.2.12 srcPortRangeStart . . . 953
8.953.2.13 tosMask . . . 953
8.954 timeInfo Struct Reference . . . 953
8.954.1 Detailed Description . . . 953
8.954.2 Field Documentation . . . 954
8.954.2.1 day . . . 954
8.954.2.2 dayLtSavingAdj . . . 954
8.954.2.3 dayOfWeek . . . 954
8.954.2.4 hour . . . 954
8.954.2.5 minute . . . 954
8.954.2.6 month . . . 955
8.954.2.7 radioInterface . . . 955
8.954.2.8 second . . . 955
8.954.2.9 timeZone . . . 955
8.954.2.10 TlvPresent . . . 955
8.954.2.11 year . . . 955
8.955 TmdDeRegNotMitigationLvlReq Struct Reference . . . 955
8.955.1 Detailed Description . . . 955
8.955.2 Field Documentation . . . 955
8.955.2.1 mitigationDevID . . . 955
8.955.2.2 mitigationDevIDLen . . . 955

8.956 TmdGetMitigationDevListResp Struct Reference . . . 955
8.956.1 Detailed Description . . . 955
8.956.2 Field Documentation . . . 956
8.956.2.1 pMitigationDevList . . . 956
8.956.2.2 pMitigationDevListLen . . . 956
8.957 TmdGetMitigationLvlReq Struct Reference . . . 956
8.957.1 Detailed Description . . . 956
8.957.2 Field Documentation . . . 956
8.957.2.1 mitigationDevID . . . 956
8.957.2.2 mitigationDevIDLen . . . 956
8.958 TmdGetMitigationLvlResp Struct Reference . . . 956
8.958.1 Detailed Description . . . 957
8.958.2 Field Documentation . . . 957
8.958.2.1 pCurrentmitigationLvl . . . 957
8.958.2.2 pReqMitigationLvl . . . 957
8.959 TmdMitigationLvlIndReq Struct Reference . . . 957
8.959.1 Detailed Description . . . 957
8.959.2 Field Documentation . . . 957
8.959.2.1 mitigationDevID . . . 957
8.959.2.2 mitigationDevIDLen . . . 957
8.960 TmdRegNotMitigationLvlReq Struct Reference . . . 957
8.960.1 Detailed Description . . . 958
8.960.2 Field Documentation . . . 958
8.960.2.1 mitigationDevID . . . 958
8.960.2.2 mitigationDevIDLen . . . 958
8.961 tokenBucket Struct Reference . . . 958
8.961.1 Detailed Description . . . 958
8.961.2 Field Documentation . . . 958
8.961.2.1 bucketSz . . . 958
8.961.2.2 peakRate . . . 958
8.961.2.3 tokenRate . . . 958
8.962 Tos Struct Reference . . . 958
8.962.1 Detailed Description . . . 959
8.962.2 Field Documentation . . . 959
8.962.2.1 mask . . . 959
8.962.2.2 val . . . 959
8.963 transferRouteMessageTlv Struct Reference . . . 959
8.963.1 Detailed Description . . . 959
8.963.2 Field Documentation . . . 959
8.963.2.1 TlvPresent . . . 959

8.963.2.2 TransferRouteMTMessageInfo . . . 959
8.964 TransferStatInd Struct Reference . . . 960
8.964.1 Detailed Description . . . 960
8.964.2 Field Documentation . . . 960
8.964.2.1 StatsMask . . . 960
8.964.2.2 StatsPeriod . . . 960
8.965 transferStatInd Struct Reference . . . 960
8.965.1 Detailed Description . . . 960
8.965.2 Field Documentation . . . 960
8.965.2.1 StatsMask . . . 960
8.965.2.2 StatsPeriod . . . 960
8.966 TransferStatsDataType Struct Reference . . . 961
8.966.1 Field Documentation . . . 961
8.966.1.1 interval . . . 961
8.967 TrStatInd Struct Reference . . . 961
8.967.1 Detailed Description . . . 961
8.967.2 Field Documentation . . . 961
8.967.2.1 statsMask . . . 961
8.967.2.2 statsPeriod . . . 961
8.968 trueIMSI Struct Reference . . . 961
8.968.1 Detailed Description . . . 962
8.968.2 Field Documentation . . . 962
8.968.2.1 imsiT1112 . . . 962
8.968.2.2 imsiTaddrNum . . . 962
8.968.2.3 imsiTS1 . . . 962
8.968.2.4 imsiTS2 . . . 962
8.968.2.5 mccT . . . 962
8.969 TXAGCList Struct Reference . . . 962
8.969.1 Detailed Description . . . 963
8.969.2 Field Documentation . . . 963
8.969.2.1 pTXAIG . . . 963
8.969.2.2 pTXComprSlope . . . 963
8.969.2.3 pTXComprThres . . . 963
8.969.2.4 pTXExpSlope . . . 963
8.969.2.5 pTXExpThres . . . 963
8.969.2.6 pTXStaticGain . . . 963
8.970 txInfo Struct Reference . . . 963
8.970.1 Detailed Description . . . 963
8.970.2 Field Documentation . . . 964
8.970.2.1 isInTraffic . . . 964

8.970.2.2 txPower . . . 964
8.971 TXPCMIIRFltr Struct Reference . . . 964
8.971.1 Detailed Description . . . 964
8.971.2 Field Documentation . . . 965
8.971.2.1 pFlag . . . 965
8.971.2.2 pStage0Val . . . 965
8.971.2.3 pStage1Val . . . 965
8.971.2.4 pStage2Val . . . 965
8.971.2.5 pStage3Val . . . 965
8.971.2.6 pStage4Val . . . 965
8.971.2.7 pStageCnt . . . 965
8.972 uim_appStatus Struct Reference . . . 965
8.972.1 Detailed Description . . . 966
8.972.2 Field Documentation . . . 968
8.972.2.1 aidLength . . . 968
8.972.2.2 aidVal . . . 968
8.972.2.3 appState . . . 968
8.972.2.4 appType . . . 968
8.972.2.5 persoFeature . . . 968
8.972.2.6 persoRetries . . . 968
8.972.2.7 persoState . . . 968
8.972.2.8 persoUnblockRetries . . . 968
8.972.2.9 pin1Retries . . . 968
8.972.2.10 pin1State . . . 968
8.972.2.11 pin2Retries . . . 968
8.972.2.12 pin2State . . . 968
8.972.2.13 puk1Retries . . . 968
8.972.2.14 puk2Retries . . . 968
8.972.2.15 univPin . . . 968
8.973 uim_authenticateResult Struct Reference . . . 968
8.973.1 Detailed Description . . . 969
8.973.2 Field Documentation . . . 969
8.973.2.1 content . . . 969
8.973.2.2 contentLen . . . 969
8.974 uim_authenticationData Struct Reference . . . 969
8.974.1 Detailed Description . . . 969
8.974.2 Field Documentation . . . 970
8.974.2.1 context . . . 970
8.974.2.2 data . . . 970
8.974.2.3 dataLen . . . 970

8.975 uim_cardResult Struct Reference . . . 970
8.975.1 Detailed Description . . . 970
8.975.2 Field Documentation . . . 971
8.975.2.1 sw1 . . . 971
8.975.2.2 sw2 . . . 971
8.976 uim_cardStatus Struct Reference . . . 971
8.976.1 Detailed Description . . . 971
8.976.2 Field Documentation . . . 972
8.976.2.1 index1xPri . . . 972
8.976.2.2 index1xSec . . . 972
8.976.2.3 indexGwPri . . . 972
8.976.2.4 indexGwSec . . . 972
8.976.2.5 numSlot . . . 972
8.976.2.6 SlotInfo . . . 972
8.977 uim_changeUIMPIN Struct Reference . . . 972
8.977.1 Detailed Description . . . 972
8.977.2 Field Documentation . . . 973
8.977.2.1 oldPINLen . . . 973
8.977.2.2 oldPINVal . . . 973
8.977.2.3 pinID . . . 973
8.977.2.4 pinLen . . . 973
8.977.2.5 pinVal . . . 973
8.978 uim_depersonalizationInformation Struct Reference . . . 973
8.978.1 Detailed Description . . . 973
8.978.2 Field Documentation . . . 974
8.978.2.1 ckLen . . . 974
8.978.2.2 ckVal . . . 974
8.978.2.3 feature . . . 974
8.978.2.4 operation . . . 974
8.979 uim_encryptedPIN1 Struct Reference . . . 974
8.979.1 Detailed Description . . . 974
8.979.2 Field Documentation . . . 975
8.979.2.1 pin1Len . . . 975
8.979.2.2 pin1Val . . . 975
8.980 uim_fileAttributes Struct Reference . . . 975
8.980.1 Detailed Description . . . 975
8.980.2 Field Documentation . . . 978
8.980.2.1 fileID . . . 978
8.980.2.2 fileSize . . . 978
8.980.2.3 fileType . . . 978

8.980.2.4 rawLen . . . 978
8.980.2.5 rawValue . . . 978
8.980.2.6 recordCount . . . 978
8.980.2.7 recordSize . . . 978
8.980.2.8 secActivate . . . 978
8.980.2.9 secActivateMask . . . 978
8.980.2.10 secDeactivate . . . 978
8.980.2.11 secDeactivateMask . . . 978
8.980.2.12 secIncrease . . . 978
8.980.2.13 secIncreaseMask . . . 978
8.980.2.14 secRead . . . 978
8.980.2.15 secReadMask . . . 978
8.980.2.16 secWrite . . . 978
8.980.2.17 secWriteMask . . . 978
8.981 uim_fileInfo Struct Reference . . . 978
8.981.1 Detailed Description . . . 978
8.981.2 Field Documentation . . . 979
8.981.2.1 fileID . . . 979
8.981.2.2 path . . . 979
8.981.2.3 pathLen . . . 979
8.982 uim_hotSwapStatus Struct Reference . . . 979
8.982.1 Detailed Description . . . 979
8.982.2 Field Documentation . . . 979
8.982.2.1 hotSwap . . . 979
8.982.2.2 hotSwapLength . . . 979
8.983 uim_personalizationStatus Struct Reference . . . 979
8.983.1 Detailed Description . . . 980
8.983.2 Field Documentation . . . 980
8.983.2.1 feature . . . 980
8.983.2.2 numFeatures . . . 980
8.983.2.3 unblockLeft . . . 980
8.983.2.4 verifyLeft . . . 980
8.984 uim_readResult Struct Reference . . . 980
8.984.1 Detailed Description . . . 981
8.984.2 Field Documentation . . . 981
8.984.2.1 content . . . 981
8.984.2.2 contentLen . . . 981
8.985 uim_readTransparentInfo Struct Reference . . . 981
8.985.1 Detailed Description . . . 981
8.985.2 Field Documentation . . . 981

8.985.2.1 length . . . 981
8.985.2.2 offset . . . 981
8.986 uim_refreshevent Struct Reference . . . 981
8.986.1 Detailed Description . . . 982
8.986.2 Field Documentation . . . 983
8.986.2.1 aid . . . 983
8.986.2.2 aidLength . . . 983
8.986.2.3 arrfileInfo . . . 983
8.986.2.4 mode . . . 983
8.986.2.5 numOfFiles . . . 983
8.986.2.6 sessionType . . . 983
8.986.2.7 stage . . . 983
8.987 uim_registerRefresh Struct Reference . . . 983
8.987.1 Detailed Description . . . 983
8.987.2 Field Documentation . . . 984
8.987.2.1 arrfileInfo . . . 984
8.987.2.2 numFiles . . . 984
8.987.2.3 registerFlag . . . 984
8.987.2.4 voteForInit . . . 984
8.988 uim_remainingRetries Struct Reference . . . 984
8.988.1 Detailed Description . . . 984
8.988.2 Field Documentation . . . 984
8.988.2.1 unblockLeft . . . 984
8.988.2.2 verifyLeft . . . 985
8.989 uim_sessionInformation Struct Reference . . . 985
8.989.1 Detailed Description . . . 985
8.989.2 Field Documentation . . . 985
8.989.2.1 aid . . . 985
8.989.2.2 aidLength . . . 985
8.989.2.3 sessionType . . . 985
8.990 uim_setPINProtection Struct Reference . . . 985
8.990.1 Detailed Description . . . 986
8.990.2 Field Documentation . . . 986
8.990.2.1 pinID . . . 986
8.990.2.2 pinLength . . . 986
8.990.2.3 pinOperation . . . 986
8.990.2.4 pinValue . . . 986
8.991 uim_slotInfo Struct Reference . . . 986
8.991.1 Detailed Description . . . 987
8.991.2 Field Documentation . . . 988

8.991.2.1 AppStatus . . . 988
8.991.2.2 cardState . . . 988
8.991.2.3 errorState . . . 988
8.991.2.4 numApp . . . 988
8.991.2.5 upinRetries . . . 988
8.991.2.6 upinState . . . 988
8.991.2.7 upukRetries . . . 988
8.992 uim_UIMSessionInformation Struct Reference . . . 988
8.992.1 Detailed Description . . . 988
8.992.2 Field Documentation . . . 989
8.992.2.1 aid . . . 989
8.992.2.2 aidLength . . . 989
8.992.2.3 sessionType . . . 989
8.993 uim_unblockUIMPIN Struct Reference . . . 989
8.993.1 Detailed Description . . . 989
8.993.2 Field Documentation . . . 989
8.993.2.1 newPINLen . . . 989
8.993.2.2 newPINVal . . . 989
8.993.2.3 pinID . . . 989
8.993.2.4 pukLen . . . 990
8.993.2.5 pukVal . . . 990
8.994 uim_verifyUIMPIN Struct Reference . . . 990
8.994.1 Detailed Description . . . 990
8.994.2 Field Documentation . . . 990
8.994.2.1 pinID . . . 990
8.994.2.2 pinLen . . . 990
8.994.2.3 pinVal . . . 990
8.995 UIMAuthenticateReq Struct Reference . . . 990
8.995.1 Detailed Description . . . 990
8.995.2 Field Documentation . . . 991
8.995.2.1 authData . . . 991
8.995.2.2 pIndicationToken . . . 991
8.995.2.3 sessionInfo . . . 991
8.996 UIMAuthenticateResp Struct Reference . . . 991
8.996.1 Detailed Description . . . 991
8.996.2 Field Documentation . . . 992
8.996.2.1 pAuthenticateResult . . . 992
8.996.2.2 pCardResult . . . 992
8.996.2.3 pIndicationToken . . . 992
8.997 UIMChangePinReq Struct Reference . . . 992

8.997.1 Detailed Description . . . 992
8.997.2 Field Documentation . . . 992
8.997.2.1 changePIN . . . 992
8.997.2.2 pIndicationToken . . . 992
8.997.2.3 pKeyReferenceID . . . 992
8.997.2.4 sessionInfo . . . 992
8.998 UIMDepersonalizationReq Struct Reference . . . 993
8.998.1 Detailed Description . . . 993
8.998.2 Field Documentation . . . 993
8.998.2.1 depersonilisationInfo . . . 993
8.999 UIMDepersonalizationResp Struct Reference . . . 993
8.999.1 Detailed Description . . . 993
8.999.2 Field Documentation . . . 993
8.999.2.1 pRemainingRetries . . . 993
8.1000 UIMEventRegisterReqResp Struct Reference . . . 993
8.1000.1 Detailed Description . . . 994
8.1000.2 Field Documentation . . . 994
8.1000.2.1 eventMask . . . 994
8.1001 UIMGetCardStatusResp Struct Reference . . . 994
8.1001.1 Detailed Description . . . 994
8.1001.2 Field Documentation . . . 994
8.1001.2.1 pCardStatus . . . 994
8.1001.2.2 pHotSwapStatus . . . 994
8.1002 UIMGetConfigurationReq Struct Reference . . . 994
8.1002.1 Detailed Description . . . 995
8.1002.2 Field Documentation . . . 995
8.1002.2.1 pConfigurationMask . . . 995
8.1003 UIMGetConfigurationResp Struct Reference . . . 995
8.1003.1 Detailed Description . . . 995
8.1003.2 Field Documentation . . . 996
8.1003.2.1 pAutoSelection . . . 996
8.1003.2.2 pHaltSubscription . . . 996
8.1003.2.3 pPersonalizationStatus . . . 996
8.1004 UIMGetFileAttributesReq Struct Reference . . . 996
8.1004.1 Detailed Description . . . 996
8.1004.2 Field Documentation . . . 996
8.1004.2.1 fileIndex . . . 996
8.1004.2.2 pIndicationToken . . . 996
8.1004.2.3 sessionInfo . . . 996
8.1005 UIMGetFileAttributesResp Struct Reference . . . 996

8.1005.1 Detailed Description . . . 997
8.1005.2 Field Documentation . . . 997
8.1005.2.1 pCardResult . . . 997
8.1005.2.2 pFileAttributes . . . 997
8.1005.2.3 pIndicationToken . . . 997
8.1006 UIMGetSlotsStatusResp Struct Reference . . . 997
8.1006.1 Detailed Description . . . 997
8.1006.2 Field Documentation . . . 997
8.1006.2.1 pNumberOfPhySlot . . . 997
8.1006.2.2 pUimSlotsStatus . . . 998
8.1007 UIMPinResp Struct Reference . . . 998
8.1007.1 Detailed Description . . . 998
8.1007.2 Field Documentation . . . 998
8.1007.2.1 pEncryptedPIN1 . . . 998
8.1007.2.2 pIndicationToken . . . 998
8.1007.2.3 pRemainingRetries . . . 998
8.1008 UIMPowerDownReq Struct Reference . . . 998
8.1008.1 Detailed Description . . . 998
8.1008.2 Field Documentation . . . 999
8.1008.2.1 slot . . . 999
8.1009 UIMPowerUpReq Struct Reference . . . 999
8.1009.1 Detailed Description . . . 999
8.1009.2 Field Documentation . . . 999
8.1009.2.1 pIgnoreHotSwapSwitch . . . 999
8.1009.2.2 slot . . . 999
8.1010 UIMReadTransparentReq Struct Reference . . . 999
8.1010.1 Detailed Description . . . 1000
8.1010.2 Field Documentation . . . 1000
8.1010.2.1 fileIndex . . . 1000
8.1010.2.2 pEncryptData . . . 1000
8.1010.2.3 pIndicationToken . . . 1000
8.1010.2.4 readTransparent . . . 1000
8.1010.2.5 sessionInfo . . . 1000
8.1011 UIMReadTransparentResp Struct Reference . . . 1000
8.1011.1 Detailed Description . . . 1000
8.1011.2 Field Documentation . . . 1001
8.1011.2.1 pCardResult . . . 1001
8.1011.2.2 pEncryptedData . . . 1001
8.1011.2.3 pIndicationToken . . . 1001
8.1011.2.4 pReadResult . . . 1001

8.1012 UIMRefreshCompleteReq Struct Reference . . . 1001
8.1012.1 Detailed Description . . . 1001
8.1012.2 Field Documentation . . . 1002
8.1012.2.1 refreshComplete . . . 1002
8.1012.2.2 sessionInfo . . . 1002
8.1013 UIMRefreshEvent Struct Reference . . . 1002
8.1013.1 Detailed Description . . . 1002
8.1013.2 Field Documentation . . . 1003
8.1013.2.1 aid . . . 1003
8.1013.2.2 aidLength . . . 1003
8.1013.2.3 arrfileInfo . . . 1003
8.1013.2.4 mode . . . 1003
8.1013.2.5 numOfFiles . . . 1003
8.1013.2.6 sessionType . . . 1003
8.1013.2.7 stage . . . 1003
8.1014 UIMRefreshGetLastEventReq Struct Reference . . . 1003
8.1014.1 Detailed Description . . . 1004
8.1014.2 Field Documentation . . . 1004
8.1014.2.1 sessionInfo . . . 1004
8.1015 UIMRefreshGetLastEventResp Struct Reference . . . 1004
8.1015.1 Detailed Description . . . 1004
8.1015.2 Field Documentation . . . 1004
8.1015.2.1 pRefreshEvent . . . 1004
8.1016 UIMRefreshOKReq Struct Reference . . . 1004
8.1016.1 Detailed Description . . . 1004
8.1016.2 Field Documentation . . . 1005
8.1016.2.1 OKtoRefresh . . . 1005
8.1016.2.2 sessionInfo . . . 1005
8.1017 UIMRefreshRegisterReq Struct Reference . . . 1005
8.1017.1 Detailed Description . . . 1005
8.1017.2 Field Documentation . . . 1005
8.1017.2.1 regRefresh . . . 1005
8.1017.2.2 sessionInfo . . . 1005
8.1018 UIMSessionInformation Struct Reference . . . 1005
8.1018.1 Detailed Description . . . 1006
8.1018.2 Field Documentation . . . 1006
8.1018.2.1 aid . . . 1006
8.1018.2.2 aidLength . . . 1006
8.1018.2.3 sessionType . . . 1006
8.1019 UIMSetPinProtectionReq Struct Reference . . . 1006

8.1019.1 Detailed Description . . . 1006
8.1019.2 Field Documentation . . . 1007
8.1019.2.1 pIndicationToken . . . 1007
8.1019.2.2 pinProtection . . . 1007
8.1019.2.3 pKeyReferenceID . . . 1007
8.1019.2.4 sessionInfo . . . 1007
8.1020 UIMSlotsStatus Struct Reference . . . 1007
8.1020.1 Detailed Description . . . 1007
8.1020.2 Field Documentation . . . 1007
8.1020.2.1 uimSlotStatus . . . 1007
8.1021 UIMSlotStatus Struct Reference . . . 1007
8.1021.1 Detailed Description . . . 1008
8.1021.2 Field Documentation . . . 1008
8.1021.2.1 bICCID . . . 1008
8.1021.2.2 bICCIDLength . . . 1008
8.1021.2.3 bLogicalSlot . . . 1008
8.1021.2.4 uPhyCardStatus . . . 1008
8.1021.2.5 uPhySlotStatus . . . 1008
8.1022 UIMSlotStatusChangeInfo Struct Reference . . . 1009
8.1022.1 Detailed Description . . . 1009
8.1022.2 Field Documentation . . . 1009
8.1022.2.1 bNumberOfPhySlots . . . 1009
8.1022.2.2 slotsstatusChange . . . 1009
8.1023 UIMStatusChangeInfo Struct Reference . . . 1009
8.1023.1 Detailed Description . . . 1009
8.1023.2 Field Documentation . . . 1009
8.1023.2.1 statusChange . . . 1009
8.1024 UIMSwitchSlotReq Struct Reference . . . 1009
8.1024.1 Detailed Description . . . 1010
8.1024.2 Field Documentation . . . 1010
8.1024.2.1 bLogicalSlot . . . 1010
8.1024.2.2 ulPhysicalSlot . . . 1010
8.1025 UIMUnblockPinReq Struct Reference . . . 1010
8.1025.1 Detailed Description . . . 1010
8.1025.2 Field Documentation . . . 1011
8.1025.2.1 pIndicationToken . . . 1011
8.1025.2.2 pKeyReferenceID . . . 1011
8.1025.2.3 sessionInfo . . . 1011
8.1025.2.4 unblockPIN . . . 1011
8.1026 UIMVerifyPinReq Struct Reference . . . 1011

8.1026.1 Detailed Description . . . 1011
8.1026.2 Field Documentation . . . 1012
8.1026.2.1 pEncryptedPIN1 . . . 1012
8.1026.2.2 pIndicationToken . . . 1012
8.1026.2.3 pKeyReferenceID . . . 1012
8.1026.2.4 sessionInfo . . . 1012
8.1026.2.5 verifyPIN . . . 1012
8.1027 UMTSInfo Struct Reference . . . 1012
8.1027.1 Detailed Description . . . 1012
8.1027.2 Field Documentation . . . 1013
8.1027.2.1 cellID . . . 1013
8.1027.2.2 ecio . . . 1013
8.1027.2.3 geranInst . . . 1013
8.1027.2.4 GeranInstInfo . . . 1013
8.1027.2.5 lac . . . 1013
8.1027.2.6 plmn . . . 1014
8.1027.2.7 psc . . . 1014
8.1027.2.8 rscp . . . 1014
8.1027.2.9 uarfcn . . . 1014
8.1027.2.10 umtsInst . . . 1014
8.1027.2.11 UMTSInstInfo . . . 1014
8.1028 UMTSinstInfo Struct Reference . . . 1014
8.1028.1 Detailed Description . . . 1014
8.1028.2 Field Documentation . . . 1014
8.1028.2.1 umtsEcio . . . 1014
8.1028.2.2 umtsPsc . . . 1014
8.1028.2.3 umtsRscp . . . 1014
8.1028.2.4 umtsUarfcn . . . 1014
8.1029 umtsLTENbrCell Struct Reference . . . 1014
8.1029.1 Detailed Description . . . 1015
8.1029.2 Field Documentation . . . 1015
8.1029.2.1 cellIsTDD . . . 1015
8.1029.2.2 earfcn . . . 1015
8.1029.2.3 pci . . . 1015
8.1029.2.4 rsrp . . . 1015
8.1029.2.5 rsrq . . . 1015
8.1029.2.6 srxlev . . . 1015
8.1030 UMTSMinQoS Struct Reference . . . 1015
8.1030.1 Detailed Description . . . 1016
8.1030.2 Field Documentation . . . 1017

8.1030.2.1 deliveryErrSDU . . . 1017
8.1030.2.2 grntDownlinkBitrate . . . 1018
8.1030.2.3 grntUplinkBitrate . . . 1018
8.1030.2.4 maxDownlinkBitrate . . . 1018
8.1030.2.5 maxSDUSize . . . 1018
8.1030.2.6 maxUplinkBitrate . . . 1018
8.1030.2.7 qosDeliveryOrder . . . 1018
8.1030.2.8 resBerRatio . . . 1018
8.1030.2.9 sduErrorRatio . . . 1018
8.1030.2.10 trafficClass . . . 1018
8.1030.2.11 trafficPriority . . . 1018
8.1030.2.12 transferDelay . . . 1018
8.1031 UMTSQoS Struct Reference . . . 1018
8.1031.1 Detailed Description . . . 1018
8.1031.2 Field Documentation . . . 1020
8.1031.2.1 deliveryErrSDU . . . 1020
8.1031.2.2 grntDownlinkBitrate . . . 1020
8.1031.2.3 grntUplinkBitrate . . . 1020
8.1031.2.4 maxDownlinkBitrate . . . 1020
8.1031.2.5 maxSDUSize . . . 1020
8.1031.2.6 maxUplinkBitrate . . . 1020
8.1031.2.7 qosDeliveryOrder . . . 1020
8.1031.2.8 resBerRatio . . . 1020
8.1031.2.9 sduErrorRatio . . . 1020
8.1031.2.10 trafficClass . . . 1020
8.1031.2.11 trafficPriority . . . 1020
8.1031.2.12 transferDelay . . . 1020
8.1032 UMTSReqQoSSigInd Struct Reference . . . 1020
8.1032.1 Detailed Description . . . 1021
8.1032.2 Field Documentation . . . 1021
8.1032.2.1 SigInd . . . 1021
8.1032.2.2 UMTSReqQoS . . . 1021
8.1033 unblockUIMPIN Struct Reference . . . 1021
8.1033.1 Detailed Description . . . 1021
8.1033.2 Field Documentation . . . 1022
8.1033.2.1 newPINLen . . . 1022
8.1033.2.2 newPINVal . . . 1022
8.1033.2.3 pinID . . . 1022
8.1033.2.4 pukLen . . . 1022
8.1033.2.5 pukVal . . . 1022

8.1034 UniversalTime Struct Reference . . . 1022
8.1034.1 Detailed Description . . . 1022
8.1034.2 Field Documentation . . . 1023
8.1034.2.1 day . . . 1023
8.1034.2.2 dayOfWeek . . . 1023
8.1034.2.3 hour . . . 1023
8.1034.2.4 minute . . . 1023
8.1034.2.5 month . . . 1023
8.1034.2.6 second . . . 1023
8.1034.2.7 year . . . 1023
8.1035 unpack_dms_GetActivationState_t Struct Reference . . . 1023
8.1035.1 Detailed Description . . . 1023
8.1035.2 Field Documentation . . . 1024
8.1035.2.1 state . . . 1024
8.1036 unpack_dms_GetBandCapability_t Struct Reference . . . 1024
8.1036.1 Detailed Description . . . 1024
8.1036.2 Field Documentation . . . 1025
8.1036.2.1 BandCapability . . . 1026
8.1036.2.2 Tlvresult . . . 1026
8.1037 unpack_dms_GetCrashAction_t Struct Reference . . . 1026
8.1037.1 Detailed Description . . . 1026
8.1037.2 Field Documentation . . . 1026
8.1037.2.1 DevCrashState . . . 1026
8.1037.2.2 Tlvresult . . . 1026
8.1038 unpack_dms_GetCustFeature_t Struct Reference . . . 1026
8.1038.1 Detailed Description . . . 1027
8.1038.2 Field Documentation . . . 1028
8.1038.2.1 DHCPRelayEnabled . . . 1028
8.1038.2.2 DisableIMSI . . . 1028
8.1038.2.3 GpsEnable . . . 1028
8.1038.2.4 GPSLPM . . . 1028
8.1038.2.5 GPSSel . . . 1028
8.1038.2.6 IPFamSupport . . . 1028
8.1038.2.7 IsVoiceEnabled . . . 1028
8.1038.2.8 RMAutoConnect . . . 1028
8.1038.2.9 SMSSupport . . . 1028
8.1038.2.10 Tlvresult . . . 1028
8.1039 unpack_dms_GetCustFeaturesV2_t Struct Reference . . . 1028
8.1039.1 Detailed Description . . . 1029
8.1039.2 Field Documentation . . . 1029

8.1039.2.1 GetCustomFeatureV2 . . . 1029
8.1039.2.2 Tlvresult . . . 1029
8.1040 unpack_dms_GetDeviceCap_t Struct Reference . . . 1029
8.1040.1 Detailed Description . . . 1029
8.1040.2 Field Documentation . . . 1030
8.1040.2.1 DataServiceCapability . . . 1030
8.1040.2.2 MaxRXChannelRate . . . 1030
8.1040.2.3 MaxTXChannelRate . . . 1030
8.1040.2.4 RadioIfaces . . . 1030
8.1040.2.5 RadioIfacesSize . . . 1030
8.1040.2.6 SimCapability . . . 1030
8.1040.2.7 Tlvresult . . . 1030
8.1041 unpack_dms_GetDeviceCapabilities_t Struct Reference . . . 1030
8.1041.1 Detailed Description . . . 1030
8.1041.2 Field Documentation . . . 1031
8.1041.2.1 dataServiceCaCapability . . . 1031
8.1041.2.2 maxRxChannelRate . . . 1031
8.1041.2.3 maxTxChannelRate . . . 1031
8.1041.2.4 RadioIfaces . . . 1031
8.1041.2.5 radioIfacesSize . . . 1031
8.1041.2.6 simCapability . . . 1031
8.1042 unpack_dms_GetDeviceHardwareRev_t Struct Reference . . . 1031
8.1042.1 Detailed Description . . . 1032
8.1042.2 Field Documentation . . . 1032
8.1042.2.1 String . . . 1032
8.1042.2.2 stringSize . . . 1032
8.1042.2.3 Tlvresult . . . 1032
8.1043 unpack_dms_GetDeviceMfr_t Struct Reference . . . 1032
8.1043.1 Detailed Description . . . 1032
8.1043.2 Field Documentation . . . 1032
8.1043.2.1 String . . . 1033
8.1043.2.2 stringSize . . . 1033
8.1043.2.3 Tlvresult . . . 1033
8.1044 unpack_dms_GetDeviceSerialNumbers_t Struct Reference . . . 1033
8.1044.1 Detailed Description . . . 1033
8.1044.2 Field Documentation . . . 1034
8.1044.2.1 esnSize . . . 1034
8.1044.2.2 ESNString . . . 1034
8.1044.2.3 meiSize . . . 1034
8.1044.2.4 MEIString . . . 1034

8.1044.2.5 meiSvnSize . . . 1034
8.1044.2.6 meiSvnString . . . 1034
8.1044.2.7 meidSize . . . 1034
8.1044.2.8 MEIDString . . . 1034
8.1044.2.9 Tlvresult . . . 1034
8.1045 unpack_dms_GetFirmwareInfo_t Struct Reference . . . 1034
8.1045.1 Detailed Description . . . 1034
8.1045.2 Field Documentation . . . 1035
8.1045.2.1 appversion_str . . . 1035
8.1045.2.2 bootversion_str . . . 1035
8.1045.2.3 carrier_str . . . 1035
8.1045.2.4 cur_carr_name . . . 1035
8.1045.2.5 cur_carr_rev . . . 1035
8.1045.2.6 modelid_str . . . 1035
8.1045.2.7 packageid_str . . . 1035
8.1045.2.8 priversion_str . . . 1035
8.1045.2.9 sku_str . . . 1035
8.1045.2.10 Tlvresult . . . 1035
8.1046 unpack_dms_GetFirmwareRevision_t Struct Reference . . . 1036
8.1046.1 Detailed Description . . . 1036
8.1046.2 Field Documentation . . . 1036
8.1046.2.1 amssSize . . . 1036
8.1046.2.2 AMSSString . . . 1036
8.1046.2.3 PRIString . . . 1036
8.1046.2.4 Tlvresult . . . 1036
8.1047 unpack_dms_GetFirmwareRevisions_t Struct Reference . . . 1036
8.1047.1 Detailed Description . . . 1037
8.1047.2 Field Documentation . . . 1037
8.1047.2.1 amssSize . . . 1037
8.1047.2.2 AMSSString . . . 1037
8.1047.2.3 bootSize . . . 1037
8.1047.2.4 BootString . . . 1037
8.1047.2.5 priSize . . . 1037
8.1047.2.6 PRIString . . . 1037
8.1047.2.7 Tlvresult . . . 1037
8.1048 unpack_dms_GetFSN_t Struct Reference . . . 1037
8.1048.1 Detailed Description . . . 1038
8.1048.2 Field Documentation . . . 1038
8.1048.2.1 String . . . 1038
8.1048.2.2 Tlvresult . . . 1038

8.1049 npack_dms_GetHardwareRevision_t Struct Reference . . . 1038
8.1049.1 Detailed Description . . . 1038
8.1049.2 Field Documentation . . . 1038
8.1049.2.1 hwVer . . . 1038
8.1050 npack_dms_GetIMSI_t Struct Reference . . . 1038
8.1050.1 Detailed Description . . . 1039
8.1050.2 Field Documentation . . . 1039
8.1050.2.1 imsi . . . 1039
8.1050.2.2 TlvResult . . . 1039
8.1051 npack_dms_GetManufacturer_t Struct Reference . . . 1039
8.1051.1 Detailed Description . . . 1039
8.1051.2 Field Documentation . . . 1039
8.1051.2.1 manufacturer . . . 1039
8.1051.2.2 TlvResult . . . 1039
8.1052 npack_dms_GetModelID_t Struct Reference . . . 1039
8.1052.1 Detailed Description . . . 1039
8.1052.2 Field Documentation . . . 1040
8.1052.2.1 modelId . . . 1040
8.1052.2.2 TlvResult . . . 1040
8.1053 npack_dms_GetNetworkTime_t Struct Reference . . . 1040
8.1053.1 Detailed Description . . . 1040
8.1053.2 Field Documentation . . . 1040
8.1053.2.1 source . . . 1040
8.1053.2.2 timestamp . . . 1040
8.1053.2.3 TlvResult . . . 1040
8.1054 npack_dms_GetOfflineReason_t Struct Reference . . . 1041
8.1054.1 Detailed Description . . . 1041
8.1054.2 Field Documentation . . . 1041
8.1054.2.1 pbPlatform . . . 1041
8.1054.2.2 pReasonMask . . . 1041
8.1054.2.3 TlvResult . . . 1041
8.1055 npack_dms_GetPower_t Struct Reference . . . 1041
8.1055.1 Detailed Description . . . 1042
8.1055.2 Field Documentation . . . 1042
8.1055.2.1 HardwareControlledMode . . . 1042
8.1055.2.2 OfflineReason . . . 1042
8.1055.2.3 OperationMode . . . 1042
8.1055.2.4 TlvResult . . . 1042
8.1056 npack_dms_GetPRLVersion_t Struct Reference . . . 1042
8.1056.1 Detailed Description . . . 1043

8.1056.2 Field Documentation . . . 1043
8.1056.2.1 Tlvresult . . . 1043
8.1056.2.2 u16PRLVersion . . . 1043
8.1056.2.3 u8PRLPreference . . . 1043
8.1057 npack_dms_GetSerialNumbers_t Struct Reference . . . 1043
8.1057.1 Detailed Description . . . 1043
8.1057.2 Field Documentation . . . 1044
8.1057.2.1 esn . . . 1044
8.1057.2.2 mei_no . . . 1044
8.1057.2.3 meisv_svn . . . 1044
8.1057.2.4 meid . . . 1044
8.1058 npack_dms_GetUSBComp_t Struct Reference . . . 1044
8.1058.1 Detailed Description . . . 1044
8.1058.2 Field Documentation . . . 1045
8.1058.2.1 NumSupUSBComps . . . 1045
8.1058.2.2 SupUSBComps . . . 1045
8.1058.2.3 Tlvresult . . . 1045
8.1058.2.4 USBComp . . . 1045
8.1059 npack_dms_GetVoiceNumber_t Struct Reference . . . 1045
8.1059.1 Detailed Description . . . 1046
8.1059.2 Field Documentation . . . 1046
8.1059.2.1 MIN . . . 1046
8.1059.2.2 minSize . . . 1046
8.1059.2.3 Tlvresult . . . 1046
8.1059.2.4 VoiceNumber . . . 1046
8.1059.2.5 voiceNumberSize . . . 1046
8.1060 npack_dms_ResetToFactoryDefaults_t Struct Reference . . . 1046
8.1060.1 Detailed Description . . . 1046
8.1060.2 Field Documentation . . . 1047
8.1060.2.1 Tlvresult . . . 1047
8.1061 npack_dms_SetActivationStatusCallback_t Struct Reference . . . 1047
8.1061.1 Detailed Description . . . 1047
8.1061.2 Field Documentation . . . 1047
8.1061.2.1 Tlvresult . . . 1047
8.1062 npack_dms_SetCrashAction_t Struct Reference . . . 1047
8.1062.1 Detailed Description . . . 1047
8.1062.2 Field Documentation . . . 1047
8.1062.2.1 notused . . . 1047
8.1063 npack_dms_SetCustFeature_t Struct Reference . . . 1047
8.1063.1 Detailed Description . . . 1048

8.1063.2 Field Documentation . . . 1048
8.1063.2.1 Tlvresult . . . 1048
8.1064 unpack_dms_SetCustFeaturesV2_t Struct Reference . . . 1048
8.1064.1 Detailed Description . . . 1048
8.1064.2 Field Documentation . . . 1048
8.1064.2.1 Tlvresult . . . 1048
8.1065 unpack_dms_SetEventReport_ind_t Struct Reference . . . 1048
8.1065.1 Detailed Description . . . 1048
8.1065.2 Field Documentation . . . 1049
8.1065.2.1 ActivationStatusTlv . . . 1049
8.1065.2.2 OperatingModeTlv . . . 1049
8.1065.2.3 Tlvresult . . . 1049
8.1066 unpack_dms_SetEventReport_t Struct Reference . . . 1049
8.1066.1 Detailed Description . . . 1049
8.1066.2 Field Documentation . . . 1049
8.1066.2.1 Tlvresult . . . 1049
8.1067 unpack_dms_SetFirmwarePreference_t Struct Reference . . . 1049
8.1067.1 Detailed Description . . . 1049
8.1067.2 Field Documentation . . . 1049
8.1067.2.1 Tlvresult . . . 1050
8.1068 unpack_dms_SetPower_t Struct Reference . . . 1050
8.1068.1 Detailed Description . . . 1050
8.1068.2 Field Documentation . . . 1050
8.1068.2.1 Tlvresult . . . 1050
8.1069 unpack_dms_SetUSBComp_t Struct Reference . . . 1050
8.1069.1 Detailed Description . . . 1050
8.1069.2 Field Documentation . . . 1050
8.1069.2.1 Tlvresult . . . 1050
8.1070 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference . . . 1050
8.1070.1 Detailed Description . . . 1051
8.1070.2 Field Documentation . . . 1051
8.1070.2.1 source . . . 1051
8.1070.2.2 Tlvresult . . . 1051
8.1070.2.3 type . . . 1051
8.1071 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference . . . 1051
8.1071.1 Detailed Description . . . 1051
8.1071.2 Field Documentation . . . 1052
8.1071.2.1 source . . . 1052
8.1071.2.2 Tlvresult . . . 1052
8.1071.2.3 type . . . 1052

8.1072 npack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 1052
8.1072.1 Detailed Description . . . 1052
8.1072.2 Field Documentation . . . 1053
8.1072.2.1 TlvResult . . . 1053
8.1073 npack_dms_SLQSGetBandCapability_t Struct Reference . . . 1053
8.1073.1 Detailed Description . . . 1053
8.1073.2 Field Documentation . . . 1056
8.1073.2.1 bandCapability . . . 1056
8.1073.2.2 is_LteBandCapability_Available . . . 1056
8.1073.2.3 is_TdsBandCapability_Available . . . 1056
8.1073.2.4 LteBandCapability . . . 1056
8.1073.2.5 TdsBandCapability . . . 1056
8.1074 npack_dms_SLQSGetERIFile_t Struct Reference . . . 1056
8.1074.1 Detailed Description . . . 1056
8.1074.2 Field Documentation . . . 1056
8.1074.2.1 eriFile . . . 1056
8.1074.2.2 TlvResult . . . 1056
8.1075 npack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference . . . 1056
8.1075.1 Detailed Description . . . 1057
8.1075.2 Field Documentation . . . 1057
8.1075.2.1 TlvResult . . . 1057
8.1076 npack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 1057
8.1076.1 Detailed Description . . . 1057
8.1076.2 Field Documentation . . . 1057
8.1076.2.1 crashInfoParam . . . 1057
8.1076.2.2 TlvResult . . . 1057
8.1077 npack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference . . . 1057
8.1077.1 Detailed Description . . . 1058
8.1077.2 Field Documentation . . . 1058
8.1077.2.1 pGetDyingGaspCfg . . . 1058
8.1077.2.2 TlvResult . . . 1058
8.1078 npack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference . . . 1058
8.1078.1 Detailed Description . . . 1058
8.1078.2 Field Documentation . . . 1058
8.1078.2.1 pGetDyingGaspStatistics . . . 1058
8.1078.2.2 TlvResult . . . 1058
8.1079 npack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference . . . 1058
8.1079.1 Detailed Description . . . 1059
8.1079.2 Field Documentation . . . 1059
8.1079.2.1 carrier . . . 1059

8.1079.2.2 fwvers . . . 1059
8.1079.2.3 numEntries . . . 1059
8.1079.2.4 pCurrImgInfo . . . 1059
8.1079.2.5 pkgver . . . 1059
8.1079.2.6 priver . . . 1059
8.1080 unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference . . . 1059
8.1080.1 Detailed Description . . . 1060
8.1080.2 Field Documentation . . . 1060
8.1080.2.1 imgType . . . 1060
8.1080.2.2 logString . . . 1060
8.1080.2.3 refData . . . 1060
8.1080.2.4 refString . . . 1061
8.1080.2.5 ResCode . . . 1061
8.1080.2.6 Tlvresult . . . 1061
8.1081 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference . . . 1061
8.1081.1 Detailed Description . . . 1061
8.1081.2 Field Documentation . . . 1061
8.1081.2.1 manString . . . 1061
8.1081.2.2 modelString . . . 1061
8.1081.2.3 plasmaIDString . . . 1061
8.1081.2.4 swVerString . . . 1061
8.1081.2.5 Tlvresult . . . 1061
8.1082 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference . . . 1062
8.1082.1 Detailed Description . . . 1062
8.1082.2 Field Documentation . . . 1062
8.1082.2.1 nameString . . . 1062
8.1082.2.2 Tlvresult . . . 1062
8.1082.2.3 versionString . . . 1062
8.1083 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference . . . 1062
8.1083.1 Detailed Description . . . 1062
8.1083.2 Field Documentation . . . 1063
8.1083.2.1 meidString . . . 1063
8.1083.2.2 Tlvresult . . . 1063
8.1084 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 1063
8.1084.1 Detailed Description . . . 1063
8.1084.2 Field Documentation . . . 1063
8.1084.2.1 Tlvresult . . . 1063
8.1085 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 1063
8.1085.1 Detailed Description . . . 1063
8.1085.2 Field Documentation . . . 1063

8.1085.2.1 TlvResult . . . 1063
8.1086 Unpack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 1063
8.1086.1 Detailed Description . . . 1064
8.1086.2 Field Documentation . . . 1064
8.1086.2.1 TlvResult . . . 1064
8.1087 Unpack_dms_SLQSUIMGetState_t Struct Reference . . . 1064
8.1087.1 Detailed Description . . . 1064
8.1087.2 Field Documentation . . . 1064
8.1087.2.1 state . . . 1064
8.1087.2.2 TlvResult . . . 1064
8.1088 Unpack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 1064
8.1088.1 Detailed Description . . . 1064
8.1088.2 Field Documentation . . . 1065
8.1088.2.1 facilityState . . . 1065
8.1088.2.2 TlvResult . . . 1065
8.1088.2.3 unblockRetriesLeft . . . 1065
8.1088.2.4 verifyRetriesLeft . . . 1065
8.1089 Unpack_dms_UIMGetICCID_t Struct Reference . . . 1065
8.1089.1 Detailed Description . . . 1065
8.1089.2 Field Documentation . . . 1066
8.1089.2.1 String . . . 1066
8.1089.2.2 stringSize . . . 1066
8.1089.2.3 TlvResult . . . 1066
8.1090 Unpack_dms_UIMGetPINStatus_t Struct Reference . . . 1066
8.1090.1 Detailed Description . . . 1066
8.1090.2 Field Documentation . . . 1067
8.1090.2.1 p1Status . . . 1067
8.1090.2.2 p1UnblockRetriesLeft . . . 1067
8.1090.2.3 p1VerifyRetriesLeft . . . 1067
8.1090.2.4 p2Status . . . 1067
8.1090.2.5 p2UnblockRetriesLeft . . . 1067
8.1090.2.6 p2VerifyRetriesLeft . . . 1067
8.1090.2.7 TlvResult . . . 1067
8.1091 Unpack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 1067
8.1091.1 Detailed Description . . . 1067
8.1091.2 Field Documentation . . . 1068
8.1091.2.1 TlvResult . . . 1068
8.1091.2.2 verifyRetriesLeft . . . 1068
8.1092 Unpack_dms_UIMSetPINProtection_t Struct Reference . . . 1068
8.1092.1 Detailed Description . . . 1068

8.1092.2 Field Documentation . . . 1068
8.1092.2.1 tlvresult . . . 1068
8.1092.2.2 unblockRetriesLeft . . . 1068
8.1092.2.3 verifyRetriesLeft . . . 1068
8.1093 unpack_dms_UIMUnblockControlKey_t Struct Reference . . . 1069
8.1093.1 Detailed Description . . . 1069
8.1093.2 Field Documentation . . . 1069
8.1093.2.1 tlvresult . . . 1069
8.1093.2.2 unblockRetriesLeft . . . 1069
8.1094 unpack_fms_GetImagesPreference_t Struct Reference . . . 1069
8.1094.1 Detailed Description . . . 1069
8.1094.2 Field Documentation . . . 1070
8.1094.2.1 imageListSize . . . 1070
8.1094.2.2 pImageList . . . 1070
8.1094.2.3 tlvresult . . . 1070
8.1095 unpack_fms_GetStoredImages_t Struct Reference . . . 1070
8.1095.1 Detailed Description . . . 1070
8.1095.2 Field Documentation . . . 1070
8.1095.2.1 imageList . . . 1070
8.1095.2.2 imagelistSize . . . 1070
8.1095.2.3 tlvresult . . . 1070
8.1096 unpack_fms_SetImagesPreference_t Struct Reference . . . 1070
8.1096.1 Detailed Description . . . 1070
8.1096.2 Field Documentation . . . 1071
8.1096.2.1 imageTypes . . . 1071
8.1096.2.2 imageTypesSize . . . 1071
8.1096.2.3 tlvresult . . . 1071
8.1097 unpack_loc_BestAvailPos_Ind_t Struct Reference . . . 1071
8.1097.1 Detailed Description . . . 1072
8.1097.2 Field Documentation . . . 1075
8.1097.2.1 pAltitudeWrtEllipsoid . . . 1075
8.1097.2.2 pAltitudeWrtMeanSeaLevel . . . 1075
8.1097.2.3 pGpsTime . . . 1075
8.1097.2.4 pHeading . . . 1075
8.1097.2.5 pHeadingUnc . . . 1075
8.1097.2.6 pHorCirConf . . . 1075
8.1097.2.7 pHorEllpConf . . . 1075
8.1097.2.8 pHorReliability . . . 1076
8.1097.2.9 pHorUncCircular . . . 1076
8.1097.2.10 pHorUncEllipseOrientAzimuth . . . 1076

8.1097.2.1 pHorUncEllipseSemiMajor . . . 1076
8.1097.2.1 pHorUncEllipseSemiMinor . . . 1076
8.1097.2.1 pLatitude . . . 1076
8.1097.2.1 pLongitude . . . 1076
8.1097.2.1 pMagneticDeviation . . . 1076
8.1097.2.1 pPrecisionDilution . . . 1076
8.1097.2.1 pSensorDataUsage . . . 1076
8.1097.2.1 pSpeedHorizontal . . . 1076
8.1097.2.1 pSpeedUnc . . . 1076
8.1097.2.2 pSpeedVertical . . . 1076
8.1097.2.2 pSpeedVerticalUnc . . . 1076
8.1097.2.2 pSvUsedforFix . . . 1076
8.1097.2.2 pTechnologyMask . . . 1076
8.1097.2.2 pTimeSrc . . . 1076
8.1097.2.2 pTimestampUtc . . . 1076
8.1097.2.2 pTimeUnc . . . 1076
8.1097.2.2 pVertConfidence . . . 1076
8.1097.2.2 pVertReliability . . . 1076
8.1097.2.2 pVertUnc . . . 1076
8.1097.2.3 pXid . . . 1076
8.1097.2.3 tstatus . . . 1076
8.1097.2.3 tlvresult . . . 1076
8.1098 unpack_loc_Delete_Assist_Data_t Struct Reference . . . 1076
8.1098.1 Detailed Description . . . 1077
8.1098.2 Field Documentation . . . 1077
8.1098.2.1 Tlvresult . . . 1077
8.1099 unpack_loc_DeleteAssistData_Ind_t Struct Reference . . . 1077
8.1099.1 Detailed Description . . . 1077
8.1099.2 Field Documentation . . . 1077
8.1099.2.1 status . . . 1078
8.1099.2.2 Tlvresult . . . 1078
8.1100 unpack_loc_EngineState_Ind_t Struct Reference . . . 1078
8.1100.1 Detailed Description . . . 1078
8.1100.2 Field Documentation . . . 1078
8.1100.2.1 engineState . . . 1078
8.1100.2.2 Tlvresult . . . 1078
8.1101 unpack_loc_EventNMEA_Ind_t Struct Reference . . . 1078
8.1101.1 Detailed Description . . . 1078
8.1101.2 Field Documentation . . . 1079
8.1101.2.1 NMEAData . . . 1079

8.1101.2.2 Tlvresult . . . 1079
8.1102 qmi_loc_EventRegister_t Struct Reference . . . 1079
8.1102.1 Detailed Description . . . 1079
8.1102.2 Field Documentation . . . 1079
8.1102.2.1 Tlvresult . . . 1079
8.1103 qmi_loc_GetServer_Ind_t Struct Reference . . . 1079
8.1103.1 Detailed Description . . . 1079
8.1103.2 Field Documentation . . . 1080
8.1103.2.1 pIPv4AddrInfo . . . 1080
8.1103.2.2 pIPv6AddrInfo . . . 1080
8.1103.2.3 pURL . . . 1080
8.1103.2.4 serverStatus . . . 1080
8.1103.2.5 serverType . . . 1080
8.1103.2.6 Tlvresult . . . 1080
8.1104 qmi_loc_GnssSvInfo_Ind_t Struct Reference . . . 1081
8.1104.1 Detailed Description . . . 1081
8.1104.2 Field Documentation . . . 1081
8.1104.2.1 altitudeAssumed . . . 1081
8.1104.2.2 pSatelliteInfo . . . 1081
8.1104.2.3 Tlvresult . . . 1081
8.1105 qmi_loc_PositionRpt_Ind_t Struct Reference . . . 1081
8.1105.1 Detailed Description . . . 1082
8.1105.2 Field Documentation . . . 1086
8.1105.2.1 pAltitudeAssumed . . . 1086
8.1105.2.2 pAltitudeWrtEllipsoid . . . 1086
8.1105.2.3 pAltitudeWrtMeanSeaLevel . . . 1086
8.1105.2.4 pFixId . . . 1086
8.1105.2.5 pGpsTime . . . 1086
8.1105.2.6 pHeading . . . 1087
8.1105.2.7 pHeadingUnc . . . 1087
8.1105.2.8 pHorConfidence . . . 1087
8.1105.2.9 pHorReliability . . . 1087
8.1105.2.10 pHorUncCircular . . . 1087
8.1105.2.11 pHorUncEllipseOrientAzimuth . . . 1087
8.1105.2.12 pHorUncEllipseSemiMajor . . . 1087
8.1105.2.13 pHorUncEllipseSemiMinor . . . 1087
8.1105.2.14 pLatitude . . . 1087
8.1105.2.15 pLeapSeconds . . . 1087
8.1105.2.16 pLongitude . . . 1087
8.1105.2.17 pMagneticDeviation . . . 1087

8.1105.2.18 pPrecisionDilution . . . 1087
8.1105.2.19 pSensorDataUsage . . . 1087
8.1105.2.20 pSpeedHorizontal . . . 1087
8.1105.2.21 pSpeedUnc . . . 1087
8.1105.2.22 pSpeedVertical . . . 1087
8.1105.2.23 pSvUsedforFix . . . 1087
8.1105.2.24 pTechnologyMask . . . 1087
8.1105.2.25 pTimeSrc . . . 1087
8.1105.2.26 pTimestampUtc . . . 1087
8.1105.2.27 pTimeUnc . . . 1087
8.1105.2.28 pVertConfidence . . . 1087
8.1105.2.29 pVertReliability . . . 1087
8.1105.2.30 pVertUnc . . . 1087
8.1105.2.31 sessionId . . . 1087
8.1105.2.32 sessionStatus . . . 1087
8.1105.2.33 tlvresult . . . 1087
8.1106 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference . . . 1088
8.1106.1 Detailed Description . . . 1088
8.1106.2 Field Documentation . . . 1088
8.1106.2.1 status . . . 1088
8.1106.2.2 tlvresult . . . 1088
8.1107 unpack_loc_SetExtPowerState_t Struct Reference . . . 1088
8.1107.1 Detailed Description . . . 1088
8.1107.2 Field Documentation . . . 1089
8.1107.2.1 tlvresult . . . 1089
8.1108 unpack_loc_SetOperationMode_Ind_t Struct Reference . . . 1089
8.1108.1 Detailed Description . . . 1089
8.1108.2 Field Documentation . . . 1089
8.1108.2.1 status . . . 1089
8.1108.2.2 tlvresult . . . 1089
8.1109 unpack_loc_SetOperationMode_t Struct Reference . . . 1089
8.1109.1 Detailed Description . . . 1089
8.1109.2 Field Documentation . . . 1090
8.1109.2.1 tlvresult . . . 1090
8.1110 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 1090
8.1110.1 Detailed Description . . . 1090
8.1110.2 Field Documentation . . . 1090
8.1110.2.1 tlvresult . . . 1090
8.1111 unpack_loc_Start_t Struct Reference . . . 1090
8.1111.1 Detailed Description . . . 1090

8.1111.2 Field Documentation . . . 1090
8.1111.2.1 Tlvresult . . . 1090
8.1112 qmunpack_loc_Stop_t Struct Reference . . . 1090
8.1112.1 Detailed Description . . . 1091
8.1112.2 Field Documentation . . . 1091
8.1112.2.1 Tlvresult . . . 1091
8.1113 qmunpack_nas_GetCDMANetworkParameters_t Struct Reference . . . 1091
8.1113.1 Detailed Description . . . 1091
8.1113.2 Field Documentation . . . 1093
8.1113.2.1 Application . . . 1093
8.1113.2.2 Broadcast . . . 1093
8.1113.2.3 CustomSCP . . . 1093
8.1113.2.4 ForceRev0 . . . 1093
8.1113.2.5 Protocol . . . 1093
8.1113.2.6 RegForeignNID . . . 1093
8.1113.2.7 RegForeignSID . . . 1093
8.1113.2.8 RegHomeSID . . . 1093
8.1113.2.9 Roaming . . . 1093
8.1113.2.10 SCI . . . 1093
8.1113.2.11 SCM . . . 1093
8.1114 qmunpack_nas_GetHomeNetwork3GPP2_t Struct Reference . . . 1093
8.1114.1 Detailed Description . . . 1094
8.1114.2 Field Documentation . . . 1095
8.1114.2.1 nameSize . . . 1095
8.1114.2.2 pMCC . . . 1095
8.1114.2.3 pMNC . . . 1095
8.1114.2.4 pName . . . 1095
8.1114.2.5 pNID . . . 1095
8.1114.2.6 pNw2DescDisp . . . 1095
8.1114.2.7 pNw2DescEnc . . . 1095
8.1114.2.8 pNw2DescLen . . . 1095
8.1114.2.9 pNw2MCC . . . 1095
8.1114.2.10 pNw2MNC . . . 1095
8.1114.2.11 pNw2Name . . . 1095
8.1114.2.12 pSID . . . 1095
8.1115 qmunpack_nas_GetHomeNetwork_t Struct Reference . . . 1095
8.1115.1 Detailed Description . . . 1095
8.1115.2 Field Documentation . . . 1096
8.1115.2.1 mcc . . . 1096
8.1115.2.2 mnc . . . 1096

8.1115.2.3 name . . . 1096
8.1115.2.4 nid . . . 1096
8.1115.2.5 sid . . . 1096
8.1116 npack_nas_GetNetworkPreference_t Struct Reference . . . 1096
8.1116.1 Detailed Description . . . 1096
8.1116.2 Field Documentation . . . 1097
8.1116.2.1 ActiveTechPref . . . 1097
8.1116.2.2 Duration . . . 1097
8.1116.2.3 PersistentTechPref . . . 1097
8.1116.2.4 Tlvresult . . . 1097
8.1117 npack_nas_GetRFInfo_t Struct Reference . . . 1097
8.1117.1 Detailed Description . . . 1097
8.1117.2 Field Documentation . . . 1098
8.1117.2.1 instancesSize . . . 1098
8.1117.2.2 RFBandInfoElements . . . 1098
8.1118 npack_nas_GetServingNetwork_t Struct Reference . . . 1098
8.1118.1 Detailed Description . . . 1098
8.1118.2 Field Documentation . . . 1100
8.1118.2.1 CSDomain . . . 1100
8.1118.2.2 DataCaps . . . 1100
8.1118.2.3 DataCapsLen . . . 1100
8.1118.2.4 MCC . . . 1100
8.1118.2.5 MNC . . . 1100
8.1118.2.6 Name . . . 1100
8.1118.2.7 nameSize . . . 1100
8.1118.2.8 PSDomain . . . 1100
8.1118.2.9 RadioIfaces . . . 1100
8.1118.2.10 RadioIfacesSize . . . 1100
8.1118.2.11 RAN . . . 1100
8.1118.2.12 RegistrationState . . . 1100
8.1118.2.13 Roaming . . . 1100
8.1119 npack_nas_GetServingNetworkCapabilities_t Struct Reference . . . 1100
8.1119.1 Detailed Description . . . 1101
8.1119.2 Field Documentation . . . 1101
8.1119.2.1 DataCaps . . . 1101
8.1119.2.2 DataCapsLen . . . 1101
8.1120 npack_nas_GetSignalStrengths_t Struct Reference . . . 1101
8.1120.1 Detailed Description . . . 1101
8.1120.2 Field Documentation . . . 1101
8.1120.2.1 len . . . 1101

8.1120.2.2 radio . . . 1101
8.1120.2.3 rssi . . . 1102
8.1121 unpack_nas_PerformNetworkScan_t Struct Reference . . . 1102
8.1121.1 Detailed Description . . . 1102
8.1121.2 Field Documentation . . . 1102
8.1121.2.1 p3GppNetworkInfoInstances . . . 1103
8.1121.2.2 p3GppNetworkInstanceSize . . . 1103
8.1121.2.3 pPCSInstance . . . 1103
8.1121.2.4 pPCSInstanceSize . . . 1103
8.1121.2.5 pRATINstance . . . 1103
8.1121.2.6 pRATInstanceSize . . . 1103
8.1121.2.7 pScanResult . . . 1103
8.1122 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference . . . 1103
8.1122.1 Detailed Description . . . 1103
8.1122.2 Field Documentation . . . 1103
8.1122.2.1 dataCaps . . . 1103
8.1122.2.2 dataCapsSize . . . 1103
8.1123 unpack_nas_SetEventReportInd_t Struct Reference . . . 1103
8.1123.1 Detailed Description . . . 1103
8.1123.2 Field Documentation . . . 1104
8.1123.2.1 RFTlv . . . 1104
8.1123.2.2 RRTlv . . . 1104
8.1123.2.3 SLQSSSTlv . . . 1104
8.1123.2.4 SSTlv . . . 1104
8.1124 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference . . . 1104
8.1124.1 Detailed Description . . . 1104
8.1124.2 Field Documentation . . . 1104
8.1124.2.1 sPhyCaAggPcellInfo . . . 1104
8.1124.2.2 sPhyCaAggScellDlBw . . . 1104
8.1124.2.3 sPhyCaAggScellIndex . . . 1104
8.1124.2.4 sPhyCaAggScellIndType . . . 1104
8.1124.2.5 sPhyCaAggScellInfo . . . 1104
8.1125 unpack_nas_SetNetworkPreference_t Struct Reference . . . 1105
8.1125.1 Detailed Description . . . 1105
8.1125.2 Field Documentation . . . 1105
8.1125.2.1 Tlvresult . . . 1105
8.1126 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . 1105
8.1126.1 Detailed Description . . . 1105
8.1126.2 Field Documentation . . . 1106
8.1126.2.1 roaming . . . 1106

8.1127 unpack_nas_SetServingSystemCallback_ind_t Struct Reference . . . 1106
8.1127.1 Detailed Description . . . 1106
8.1127.2 Field Documentation . . . 1106
8.1127.2.1 SSInfo . . . 1106
8.1127.2.2 Tlvresult . . . 1106
8.1128 unpack_nas_SLQSGetErrorRate_t Struct Reference . . . 1106
8.1128.1 Detailed Description . . . 1106
8.1128.2 Field Documentation . . . 1107
8.1128.2.1 pCDMAFrameErrRate . . . 1107
8.1128.2.2 pGSMBER . . . 1107
8.1128.2.3 pHDRPackErrRate . . . 1107
8.1128.2.4 pWCDMABER . . . 1107
8.1129 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . 1107
8.1129.1 Detailed Description . . . 1107
8.1129.2 Field Documentation . . . 1108
8.1129.2.1 LTECphyCaInfo . . . 1108
8.1129.2.2 Tlvresult . . . 1108
8.1130 unpack_nas_SLQSGetNetworkTime_t Struct Reference . . . 1108
8.1130.1 Detailed Description . . . 1108
8.1130.2 Field Documentation . . . 1108
8.1130.2.1 p3GPP2TimeInfo . . . 1108
8.1130.2.2 p3GPPTimeInfo . . . 1108
8.1131 unpack_nas_SLQSGetOperatorNameData_t Struct Reference . . . 1108
8.1131.1 Detailed Description . . . 1108
8.1131.2 Field Documentation . . . 1109
8.1131.2.1 pNITZInformation . . . 1109
8.1131.2.2 pOperatorNameString . . . 1109
8.1131.2.3 pOperatorPLMNList . . . 1109
8.1131.2.4 pPLMNNetworkName . . . 1109
8.1131.2.5 pSrvcProviderName . . . 1109
8.1132 unpack_nas_SLQSGetPLMNName_t Struct Reference . . . 1109
8.1132.1 Field Documentation . . . 1109
8.1132.1.1 longName . . . 1109
8.1132.1.2 longNameCI . . . 1110
8.1132.1.3 longNameEn . . . 1110
8.1132.1.4 longNameLen . . . 1110
8.1132.1.5 longNameSB . . . 1110
8.1132.1.6 shortName . . . 1110
8.1132.1.7 shortNameCI . . . 1110
8.1132.1.8 shortNameEn . . . 1110

8.1132.1.9 shortNameLen . . . 1110
8.1132.1.10 shortNameSB . . . 1110
8.1132.1.11 spn . . . 1110
8.1132.1.12 spnEncoding . . . 1110
8.1132.1.13 spnLength . . . 1110
8.1133 npack_nas_SLQSGetServingSystem_t Struct Reference . . . 1110
8.1133.1 Detailed Description . . . 1111
8.1133.2 Field Documentation . . . 1113
8.1133.2.1 BasestationID . . . 1113
8.1133.2.2 BasestationLatitude . . . 1113
8.1133.2.3 BasestationLongitude . . . 1113
8.1133.2.4 CallBarStatus . . . 1113
8.1133.2.5 CDMA_P_Rev . . . 1113
8.1133.2.6 CDMASystemInfoExt . . . 1113
8.1133.2.7 CellID . . . 1113
8.1133.2.8 ConcSvcInfo . . . 1113
8.1133.2.9 CurrentPLMN . . . 1113
8.1133.2.10 DataSrvCapabilities . . . 1113
8.1133.2.11 DefaultRoamInd . . . 1113
8.1133.2.12 DetailedSvcInfo . . . 1113
8.1133.2.13 DTMInd . . . 1113
8.1133.2.14 gpp2TimeZone . . . 1113
8.1133.2.15 GppNetworkDSTAdjustment . . . 1113
8.1133.2.16 GppTimeZone . . . 1113
8.1133.2.17 HdrPersonality . . . 1113
8.1133.2.18 lac . . . 1113
8.1133.2.19 NetworkID . . . 1114
8.1133.2.20 PRLInd . . . 1114
8.1133.2.21 RoamIndicatorVal . . . 1114
8.1133.2.22 RoamingIndicatorList . . . 1114
8.1133.2.23 ServingSystem . . . 1114
8.1133.2.24 SystemID . . . 1114
8.1133.2.25 TrackAreaCode . . . 1114
8.1134 npack_nas_SLQSGetSignalStrength_t Struct Reference . . . 1114
8.1134.1 Detailed Description . . . 1114
8.1134.2 Field Documentation . . . 1116
8.1134.2.1 ecioList . . . 1116
8.1134.2.2 ecioListLen . . . 1116
8.1134.2.3 errorRateList . . . 1116
8.1134.2.4 errorRateListLen . . . 1116

8.1134.2.5 io . . . 1116
8.1134.2.6 rsrp . . . 1116
8.1134.2.7 snr . . . 1116
8.1134.2.8 srqInfo . . . 1116
8.1134.2.9 rxSignalStrengthList . . . 1116
8.1134.2.10 rxSignalStrengthListLen . . . 1116
8.1134.2.11 signalStrengthReqMask . . . 1116
8.1134.2.12 snr . . . 1116
8.1135 unpack_nas_SLQSGetSysInfo_t Struct Reference . . . 1116
8.1135.1 Detailed Description . . . 1117
8.1135.2 Field Documentation . . . 1118
8.1135.2.1 pAddCDMASysInfo . . . 1118
8.1135.2.2 pAddGSMSysInfo . . . 1118
8.1135.2.3 pAddHDRSysInfo . . . 1118
8.1135.2.4 pAddLTESysInfo . . . 1118
8.1135.2.5 pAddWCDMASysInfo . . . 1119
8.1135.2.6 pCDMASrvStatusInfo . . . 1119
8.1135.2.7 pCDMASysInfo . . . 1119
8.1135.2.8 pGSMCallBarringSysInfo . . . 1119
8.1135.2.9 pGSMCipherDomainSysInfo . . . 1119
8.1135.2.10 pGSMSrvStatusInfo . . . 1119
8.1135.2.11 pGSMSysInfo . . . 1119
8.1135.2.12 pHDRSrvStatusInfo . . . 1119
8.1135.2.13 pHDRSysInfo . . . 1119
8.1135.2.14 pLTESrvStatusInfo . . . 1119
8.1135.2.15 pLTESysInfo . . . 1119
8.1135.2.16 pLTEVoiceSupportSysInfo . . . 1119
8.1135.2.17 pWCDMACallBarringSysInfo . . . 1119
8.1135.2.18 pWCDMACipherDomainSysInfo . . . 1119
8.1135.2.19 pWCDMASrvStatusInfo . . . 1119
8.1135.2.20 pWCDMASysInfo . . . 1119
8.1136 unpack_nas_SLQSGetSysSelectionPref_t Struct Reference . . . 1119
8.1136.1 Detailed Description . . . 1119
8.1136.2 Field Documentation . . . 1123
8.1136.2.1 pBandPref . . . 1123
8.1136.2.2 pEmerMode . . . 1123
8.1136.2.3 pGWAcqOrderPref . . . 1123
8.1136.2.4 pLTEBandPref . . . 1123
8.1136.2.5 pModePref . . . 1123
8.1136.2.6 pNetSelPref . . . 1123

8.1136.2.7 pPRLPref . . . 1123
8.1136.2.8 pRoamPref . . . 1123
8.1136.2.9 pSrvDomainPref . . . 1123
8.1137 unpack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 1123
8.1137.1 Detailed Description . . . 1124
8.1137.2 Field Documentation . . . 1124
8.1137.2.1 pCDMAChannel . . . 1124
8.1137.2.2 pDirNum . . . 1124
8.1137.2.3 pHomeSIDNID . . . 1124
8.1137.2.4 pMinBasedIMSI . . . 1124
8.1137.2.5 pNAMNameInfo . . . 1124
8.1137.2.6 pTrueIMSI . . . 1124
8.1138 unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference . . . 1124
8.1138.1 Detailed Description . . . 1125
8.1138.2 Field Documentation . . . 1125
8.1138.2.1 pCDMAInfo . . . 1125
8.1138.2.2 pGERANInfo . . . 1125
8.1138.2.3 pLTEInfoInterfreq . . . 1125
8.1138.2.4 pLTEInfoIntrafreq . . . 1125
8.1138.2.5 pLTEInfoNeighboringGSM . . . 1125
8.1138.2.6 pLTEInfoNeighboringWCDMA . . . 1125
8.1138.2.7 pUMTSCellID . . . 1125
8.1138.2.8 pUMTSInfo . . . 1125
8.1138.2.9 pWCDMAInfoLTENeighborCell . . . 1126
8.1139 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference . . . 1126
8.1139.1 Detailed Description . . . 1126
8.1139.2 Field Documentation . . . 1126
8.1139.2.1 pColorCode . . . 1126
8.1140 unpack_nas_SLQSNasGetSigInfo_t Struct Reference . . . 1126
8.1140.1 Detailed Description . . . 1126
8.1140.2 Field Documentation . . . 1127
8.1140.2.1 CDMASSInfo . . . 1127
8.1140.2.2 GSMSSInfo . . . 1127
8.1140.2.3 HDRSSInfo . . . 1127
8.1140.2.4 LTESSInfo . . . 1127
8.1140.2.5 WCDMASSInfo . . . 1127
8.1141 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 1127
8.1141.1 Detailed Description . . . 1127
8.1141.2 Field Documentation . . . 1127
8.1141.2.1 pRXChain0Info . . . 1127

8.1141.2.2 pRXChain1Info . . . 1127
8.1141.2.3 pTXInfo . . . 1127
8.1142 qmi_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . 1128
8.1142.1 Detailed Description . . . 1128
8.1142.2 Field Documentation . . . 1128
8.1142.2.1 pDayltSavAdj . . . 1128
8.1142.2.2 pRadioInterface . . . 1128
8.1142.2.3 pTimeZone . . . 1128
8.1142.2.4 universalTime . . . 1128
8.1143 qmi_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . 1128
8.1143.1 Detailed Description . . . 1129
8.1143.2 Field Documentation . . . 1129
8.1143.2.1 pCDMASigInfo . . . 1129
8.1143.2.2 pGSMSigInfo . . . 1129
8.1143.2.3 pHDRSigInfo . . . 1129
8.1143.2.4 pLTESigInfo . . . 1129
8.1143.2.5 pRscp . . . 1129
8.1143.2.6 pTDSCDMASigInfoExt . . . 1129
8.1143.2.7 pWCDMASigInfo . . . 1130
8.1144 qmi_nas_SLQSNASSwiGetChannelLock_t Struct Reference . . . 1130
8.1144.1 Detailed Description . . . 1130
8.1144.2 Field Documentation . . . 1130
8.1144.2.1 pLteEARFCN . . . 1130
8.1144.2.2 pLtePCI . . . 1130
8.1144.2.3 pWcdmaUARFCN . . . 1130
8.1145 qmi_nas_SLQSNasSwiModemStatus_t Struct Reference . . . 1130
8.1145.1 Detailed Description . . . 1130
8.1145.2 Field Documentation . . . 1131
8.1145.2.1 commonInfo . . . 1131
8.1145.2.2 pLTEInfo . . . 1131
8.1146 qmi_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . 1131
8.1146.1 Detailed Description . . . 1131
8.1146.2 Field Documentation . . . 1131
8.1146.2.1 Info . . . 1131
8.1146.2.2 Tlvresult . . . 1131
8.1147 qmi_nas_SLQSNasTimerCallback_ind_t Struct Reference . . . 1131
8.1147.1 Detailed Description . . . 1131
8.1147.2 Field Documentation . . . 1132
8.1147.2.1 t3396_apn . . . 1132
8.1147.2.2 t3396_plmn_id . . . 1132

8.1147.2.3 i3396_val . . . 1132
8.1148 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . 1132
8.1148.1 Detailed Description . . . 1132
8.1148.2 Field Documentation . . . 1132
8.1148.2.1 tlvInfo . . . 1132
8.1148.2.2 tlvresult . . . 1132
8.1149 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference . . . 1132
8.1149.1 Detailed Description . . . 1132
8.1149.2 Field Documentation . . . 1133
8.1149.2.1 pCurrentPersonality . . . 1133
8.1149.2.2 pPersonalityListLength . . . 1133
8.1149.2.3 pProtocolSubtypeElement . . . 1133
8.1150 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference . . . 1133
8.1150.1 Detailed Description . . . 1133
8.1150.2 Field Documentation . . . 1133
8.1150.2.1 pAppSubType . . . 1133
8.1150.2.2 pCurrentPrsnlty . . . 1133
8.1150.2.3 pPersonalityListLength . . . 1133
8.1150.2.4 pProtoSubTypElmnt . . . 1133
8.1151 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference . . . 1134
8.1151.1 Detailed Description . . . 1134
8.1151.2 Field Documentation . . . 1134
8.1151.2.1 pDRCParams . . . 1134
8.1151.2.2 pPilotSetData . . . 1134
8.1151.2.3 pUATI . . . 1134
8.1152 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . 1134
8.1152.1 Detailed Description . . . 1134
8.1152.2 Field Documentation . . . 1135
8.1152.2.1 CQIValueCW0 . . . 1135
8.1152.2.2 CQIValueCW1 . . . 1135
8.1152.2.3 ValidityCW0 . . . 1135
8.1152.2.4 ValidityCW1 . . . 1135
8.1153 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference . . . 1135
8.1153.1 Detailed Description . . . 1135
8.1153.2 Field Documentation . . . 1135
8.1153.2.1 pSccRxInfo . . . 1135
8.1154 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference . . . 1135
8.1154.1 Detailed Description . . . 1136
8.1154.2 Field Documentation . . . 1136
8.1154.2.1 pDataStatusDetail . . . 1136

8.1154.2.2 pDeviceConfigDetail . . . 1136
8.1154.2.3 pNetworkStat1x . . . 1136
8.1154.2.4 pNetworkStatEVDO . . . 1136
8.1154.2.5 pObjectVer . . . 1136
8.1155 npack_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . 1136
8.1155.1 Detailed Description . . . 1137
8.1155.2 Field Documentation . . . 1138
8.1155.2.1 pAddCDMASysInfo . . . 1138
8.1155.2.2 pAddGSMSysInfo . . . 1138
8.1155.2.3 pAddHDRSysInfo . . . 1138
8.1155.2.4 pAddLTESysInfo . . . 1138
8.1155.2.5 pAddWCDMASysInfo . . . 1138
8.1155.2.6 pCDMASrvStatusInfo . . . 1138
8.1155.2.7 pCDMASysInfo . . . 1139
8.1155.2.8 pGSMCallBarringSysInfo . . . 1139
8.1155.2.9 pGSMCipherDomainSysInfo . . . 1139
8.1155.2.10 pGSMSrvStatusInfo . . . 1139
8.1155.2.11 pGSMSysInfo . . . 1139
8.1155.2.12 pHDRSrvStatusInfo . . . 1139
8.1155.2.13 pHDRSysInfo . . . 1139
8.1155.2.14 pLTESrvStatusInfo . . . 1139
8.1155.2.15 pLTESysInfo . . . 1139
8.1155.2.16 pLTEVoiceSupportSysInfo . . . 1139
8.1155.2.17 pSysInfoNoChange . . . 1139
8.1155.2.18 pWCDMACallBarringSysInfo . . . 1139
8.1155.2.19 pWCDMACipherDomainSysInfo . . . 1139
8.1155.2.20 pWCDMASrvStatusInfo . . . 1139
8.1155.2.21 pWCDMASysInfo . . . 1139
8.1156 npack_omaDmConfigTlv_t Struct Reference . . . 1139
8.1156.1 Detailed Description . . . 1139
8.1156.2 Field Documentation . . . 1140
8.1156.2.1 alertmsg . . . 1140
8.1156.2.2 alertmsglength . . . 1140
8.1156.2.3 state . . . 1140
8.1156.2.4 userInputReq . . . 1140
8.1156.2.5 userInputTimeout . . . 1140
8.1157 npack_omaDmFotaTlv_t Struct Reference . . . 1140
8.1157.1 Detailed Description . . . 1140
8.1157.2 Field Documentation . . . 1142
8.1157.2.1 description . . . 1142

8.1157.2.2 descriptionlength . . . 1142
8.1157.2.3 fwdloadsize . . . 1142
8.1157.2.4 fwloadComplete . . . 1142
8.1157.2.5 namelength . . . 1142
8.1157.2.6 package_name . . . 1142
8.1157.2.7 sessionType . . . 1142
8.1157.2.8 severity . . . 1142
8.1157.2.9 state . . . 1142
8.1157.2.10 updateCompleteStatus . . . 1142
8.1157.2.11 userInputReq . . . 1142
8.1157.2.12 userInputTimeout . . . 1142
8.1157.2.13 version . . . 1142
8.1157.2.14 versionlength . . . 1142
8.1158 unpack_omaDmNotificationsTlv_t Struct Reference . . . 1142
8.1158.1 Detailed Description . . . 1142
8.1158.2 Field Documentation . . . 1143
8.1158.2.1 notification . . . 1143
8.1158.2.2 sessionStatus . . . 1143
8.1159 unpack_pds_GetPDSDefaults_t Struct Reference . . . 1143
8.1159.1 Detailed Description . . . 1143
8.1159.2 Field Documentation . . . 1144
8.1159.2.1 pAccuracy . . . 1144
8.1159.2.2 pInterval . . . 1144
8.1159.2.3 pOperation . . . 1144
8.1159.2.4 pTimeout . . . 1144
8.1160 unpack_pds_GetPDSState_t Struct Reference . . . 1144
8.1160.1 Detailed Description . . . 1144
8.1160.2 Field Documentation . . . 1144
8.1160.2.1 pEnabledStatus . . . 1144
8.1160.2.2 pTrackingStatus . . . 1144
8.1161 unpack_pds_GetPortAutomaticTracking_t Struct Reference . . . 1145
8.1161.1 Detailed Description . . . 1145
8.1161.2 Field Documentation . . . 1145
8.1161.2.1 pbAuto . . . 1145
8.1162 unpack_pds_GetServiceAutomaticTracking_t Struct Reference . . . 1145
8.1162.1 Detailed Description . . . 1145
8.1162.2 Field Documentation . . . 1145
8.1162.2.1 pbAuto . . . 1145
8.1163 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference . . . 1145
8.1163.1 Detailed Description . . . 1146

8.1163.2 Field Documentation . . . 1146
8.1163.2.1 pbEnabled . . . 1146
8.1163.2.2 pInterval . . . 1146
8.1164 unpack_pds_GetXTRANetwork_t Struct Reference . . . 1146
8.1164.1 Detailed Description . . . 1146
8.1164.2 Field Documentation . . . 1146
8.1164.2.1 pPreference . . . 1146
8.1165 unpack_pds_GetXTRAValidity_t Struct Reference . . . 1146
8.1165.1 Detailed Description . . . 1147
8.1165.2 Field Documentation . . . 1147
8.1165.2.1 pDuration . . . 1147
8.1165.2.2 pGPSWeek . . . 1147
8.1165.2.3 pGPSWeekOffset . . . 1147
8.1166 unpack_pds_SetEventReport_Ind_t Struct Reference . . . 1147
8.1166.1 Detailed Description . . . 1147
8.1166.2 Field Documentation . . . 1148
8.1166.2.1 dLatitude . . . 1148
8.1166.2.2 dLongitude . . . 1148
8.1166.2.3 has_dLatitude . . . 1148
8.1166.2.4 has_dLongitude . . . 1149
8.1166.2.5 has_PositionDataNMEA . . . 1149
8.1166.2.6 has_posSrc . . . 1149
8.1166.2.7 has_SessionStatus . . . 1149
8.1166.2.8 PositionDataNMEA . . . 1149
8.1166.2.9 posSrc . . . 1149
8.1166.2.10 SessionStatus . . . 1149
8.1167 unpack_pds_SetPdsState_Ind_t Struct Reference . . . 1149
8.1167.1 Detailed Description . . . 1149
8.1167.2 Field Documentation . . . 1149
8.1167.2.1 EnabledStatus . . . 1149
8.1167.2.2 TlvPresent . . . 1149
8.1167.2.3 TrackingStatus . . . 1149
8.1168 unpack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 1149
8.1168.1 Detailed Description . . . 1150
8.1168.2 Field Documentation . . . 1150
8.1168.2.1 pServerAddress . . . 1150
8.1168.2.2 pServerPort . . . 1150
8.1168.2.3 pServerURL . . . 1150
8.1168.2.4 pServerURLLength . . . 1150
8.1169 unpack_pds_SLQSGetGPSStateInfo_t Struct Reference . . . 1150

8.1169.1 Detailed Description . . . 1151
8.1169.2 Field Documentation . . . 1153
8.1169.2.1 Altitude . . . 1153
8.1169.2.2 EngineState . . . 1153
8.1169.2.3 glo_almanac_sv_msk . . . 1153
8.1169.2.4 glo_ephemeris_sv_msk . . . 1153
8.1169.2.5 glo_health_sv_msk . . . 1153
8.1169.2.6 glo_visible_sv_msk . . . 1153
8.1169.2.7 gps_almanac_sv_msk . . . 1153
8.1169.2.8 gps_ephemeris_sv_msk . . . 1154
8.1169.2.9 gps_health_sv_msk . . . 1154
8.1169.2.10 gps_visible_sv_msk . . . 1154
8.1169.2.11 HorizontalUncertainty . . . 1154
8.1169.2.12 sno_valid . . . 1154
8.1169.2.13 atitude . . . 1154
8.1169.2.14 ongitude . . . 1154
8.1169.2.15 bas_almanac_sv_msk . . . 1154
8.1169.2.16 bas_ephemeris_sv_msk . . . 1154
8.1169.2.17 bas_health_sv_msk . . . 1154
8.1169.2.18 bas_visible_sv_msk . . . 1154
8.1169.2.19 ime_uncert_ms . . . 1154
8.1169.2.20 imeStmp_gps_week . . . 1154
8.1169.2.21 imeStmp_tow_ms . . . 1154
8.1169.2.22 alidMask . . . 1154
8.1169.2.23 erticalUncertainty . . . 1154
8.1169.2.24 ra_start_gps_minutes . . . 1154
8.1169.2.25 ra_start_gps_week . . . 1154
8.1169.2.26 ra_valid_duration_hours . . . 1154
8.1170 npack_qmi_t Struct Reference . . . 1154
8.1170.1 Detailed Description . . . 1154
8.1170.2 Field Documentation . . . 1155
8.1170.2.1 msgid . . . 1155
8.1170.2.2 ype . . . 1155
8.1170.2.3 xid . . . 1155
8.1171 npack_qos_dataRate_t Struct Reference . . . 1155
8.1171.1 Detailed Description . . . 1155
8.1171.2 Field Documentation . . . 1155
8.1171.2.1 dataRateMax . . . 1155
8.1171.2.2 guaranteedRate . . . 1155
8.1172 npack_qos_IPv4Addr_t Struct Reference . . . 1155

8.1172.1 Detailed Description . . . 1155
8.1172.2 Field Documentation . . . 1155
8.1172.2.1 addr . . . 1156
8.1172.2.2 subnetMask . . . 1156
8.1173 qmi_qos_IPv6Addr_t Struct Reference . . . 1156
8.1173.1 Detailed Description . . . 1156
8.1173.2 Field Documentation . . . 1156
8.1173.2.1 addr . . . 1156
8.1173.2.2 prefixLen . . . 1156
8.1174 qmi_qos_IPv6TrafCls_t Struct Reference . . . 1156
8.1174.1 Detailed Description . . . 1156
8.1174.2 Field Documentation . . . 1156
8.1174.2.1 mask . . . 1157
8.1174.2.2 val . . . 1157
8.1175 qmi_qos_pktErrRate_t Struct Reference . . . 1157
8.1175.1 Detailed Description . . . 1157
8.1175.2 Field Documentation . . . 1157
8.1175.2.1 exponent . . . 1157
8.1175.2.2 multiplier . . . 1157
8.1176 qmi_qos_Port_t Struct Reference . . . 1157
8.1176.1 Detailed Description . . . 1157
8.1176.2 Field Documentation . . . 1157
8.1176.2.1 port . . . 1157
8.1176.2.2 range . . . 1157
8.1177 qmi_qos_QosFlowInfo_t Struct Reference . . . 1157
8.1177.1 Detailed Description . . . 1158
8.1177.2 Field Documentation . . . 1159
8.1177.2.1 BearerID . . . 1159
8.1177.2.2 is_RxQFlowGranted_Available . . . 1159
8.1177.2.3 is_TxQFlowGranted_Available . . . 1159
8.1177.2.4 NumRxFilters . . . 1159
8.1177.2.5 NumTxFilters . . . 1159
8.1177.2.6 QFlowState . . . 1159
8.1177.2.7 RxQFilter . . . 1159
8.1177.2.8 RxQFlowGranted . . . 1159
8.1177.2.9 TxQFilter . . . 1159
8.1177.2.10 TxQFlowGranted . . . 1159
8.1178 qmi_qos_QosFlowInfoState_t Struct Reference . . . 1159
8.1178.1 Detailed Description . . . 1159
8.1178.2 Field Documentation . . . 1159

8.1178.2.1 id . . . 1159
8.1178.2.2 isNewFlow . . . 1160
8.1178.2.3 state . . . 1160
8.1179 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . 1160
8.1179.1 Detailed Description . . . 1160
8.1179.2 Field Documentation . . . 1160
8.1179.2.1 NWQoSStatus . . . 1160
8.1180 unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 1160
8.1180.1 Detailed Description . . . 1160
8.1180.2 Field Documentation . . . 1161
8.1180.2.1 ambr_dl . . . 1161
8.1180.2.2 ambr_dl_ext . . . 1161
8.1180.2.3 ambr_dl_ext2 . . . 1161
8.1180.2.4 ambr_ul . . . 1161
8.1180.2.5 ambr_ul_ext . . . 1161
8.1180.2.6 ambr_ul_ext2 . . . 1161
8.1180.2.7 apnId . . . 1161
8.1181 unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 1161
8.1181.1 Detailed Description . . . 1162
8.1181.2 Field Documentation . . . 1162
8.1181.2.1 apnId . . . 1162
8.1181.2.2 numQosFlow . . . 1162
8.1181.2.3 qosFlow . . . 1162
8.1181.2.4 total_rx_bytes . . . 1162
8.1181.2.5 total_rx_pkt . . . 1162
8.1181.2.6 total_tx_bytes . . . 1162
8.1181.2.7 total_tx_bytes_drp . . . 1162
8.1181.2.8 total_tx_pkt . . . 1162
8.1181.2.9 total_tx_pkt_drp . . . 1163
8.1182 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . 1163
8.1182.1 Detailed Description . . . 1163
8.1182.2 Field Documentation . . . 1163
8.1182.2.1 NumFlows . . . 1163
8.1182.2.2 QosFlowInfo . . . 1163
8.1183 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . 1163
8.1183.1 Detailed Description . . . 1163
8.1183.2 Field Documentation . . . 1163
8.1183.2.1 status . . . 1164
8.1184 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . 1164
8.1184.1 Detailed Description . . . 1164

8.1184.2 Field Documentation . . . 1164
8.1184.2.1 event . . . 1164
8.1185 npack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . 1164
8.1185.1 Detailed Description . . . 1164
8.1185.2 Field Documentation . . . 1165
8.1185.2.1 event . . . 1165
8.1185.2.2 id . . . 1165
8.1185.2.3 reason . . . 1165
8.1185.2.4 status . . . 1165
8.1186 npack_qos_swiQosFilter_t Struct Reference . . . 1165
8.1186.1 Detailed Description . . . 1166
8.1186.2 Field Documentation . . . 1167
8.1186.2.1 EspSpi . . . 1167
8.1186.2.2 id . . . 1168
8.1186.2.3 index . . . 1168
8.1186.2.4 IPv4DstAddr . . . 1168
8.1186.2.5 IPv4SrcAddr . . . 1168
8.1186.2.6 IPv4Tos . . . 1168
8.1186.2.7 IPv6DstAddr . . . 1168
8.1186.2.8 IPv6Label . . . 1168
8.1186.2.9 IPv6SrcAddr . . . 1168
8.1186.2.10 IPv6TrafCls . . . 1168
8.1186.2.11 is_EspSpi_Available . . . 1168
8.1186.2.12 is_Id_Available . . . 1168
8.1186.2.13 is_IPv4DstAddr_Available . . . 1168
8.1186.2.14 is_IPv4SrcAddr_Available . . . 1168
8.1186.2.15 is_IPv4Tos_Available . . . 1168
8.1186.2.16 is_IPv6DstAddr_Available . . . 1168
8.1186.2.17 is_IPv6Label_Available . . . 1168
8.1186.2.18 is_IPv6SrcAddr_Available . . . 1168
8.1186.2.19 is_IPv6TrafCls_Available . . . 1168
8.1186.2.20 is_NxtHdrProto_Available . . . 1168
8.1186.2.21 is_Precedence_Available . . . 1168
8.1186.2.22 is_TCPDstPort_Available . . . 1168
8.1186.2.23 is_TCPSrcPort_Available . . . 1168
8.1186.2.24 is_TranDstPort_Available . . . 1168
8.1186.2.25 is_TranSrcPort_Available . . . 1168
8.1186.2.26 is_UDPDstPort_Available . . . 1168
8.1186.2.27 is_UDPSrcPort_Available . . . 1168
8.1186.2.28 NxtHdrProto . . . 1168

8.1186.2.29 Precedence . . . 1168
8.1186.2.30 TCPDstPort . . . 1169
8.1186.2.31 TCPSrcPort . . . 1169
8.1186.2.32 TranDstPort . . . 1169
8.1186.2.33 TranSrcPort . . . 1169
8.1186.2.34 UDPDstPort . . . 1169
8.1186.2.35 UDPSrcPort . . . 1169
8.1186.2.36 Version . . . 1169
8.1187 unpack_qos_swiQosFlow_t Struct Reference . . . 1169
8.1187.1 Detailed Description . . . 1170
8.1187.2 Field Documentation . . . 1172
8.1187.2.1 DataRate . . . 1172
8.1187.2.2 Index . . . 1172
8.1187.2.3 is_DataRate_Available . . . 1172
8.1187.2.4 is_Jitter_Available . . . 1172
8.1187.2.5 is_Latency_Available . . . 1172
8.1187.2.6 is_LteQci_Available . . . 1172
8.1187.2.7 is_MaxAllowedPktSz_Available . . . 1172
8.1187.2.8 is_MinPolicedPktSz_Available . . . 1172
8.1187.2.9 is_PktErrRate_Available . . . 1172
8.1187.2.10 is_ProfileId3GPP2_Available . . . 1172
8.1187.2.11 is_TokenBucket_Available . . . 1172
8.1187.2.12 is_TrafficClass_Available . . . 1172
8.1187.2.13 is_val_3GPP2Pri_Available . . . 1172
8.1187.2.14 is_val_3GPPImCn_Available . . . 1172
8.1187.2.15 is_val_3GPPResResidualBER_Available . . . 1172
8.1187.2.16 is_val_3GPPSigInd_Available . . . 1172
8.1187.2.17 is_val_3GPPTraHdlPri_Available . . . 1172
8.1187.2.18 Jitter . . . 1172
8.1187.2.19 Latency . . . 1172
8.1187.2.20 LteQci . . . 1172
8.1187.2.21 MaxAllowedPktSz . . . 1172
8.1187.2.22 MinPolicedPktSz . . . 1173
8.1187.2.23 PktErrRate . . . 1173
8.1187.2.24 ProfileId3GPP2 . . . 1173
8.1187.2.25 TokenBucket . . . 1173
8.1187.2.26 TrafficClass . . . 1173
8.1187.2.27 val_3GPP2Pri . . . 1173
8.1187.2.28 val_3GPPImCn . . . 1173
8.1187.2.29 val_3GPPResResidualBER . . . 1173

8.1187.2.30 val_3GPPSigInd . . . 1173
8.1187.2.31 val_3GPPTraHdlPri . . . 1173
8.1188 unpack_qos_tokenBucket_t Struct Reference . . . 1173
8.1188.1 Detailed Description . . . 1173
8.1188.2 Field Documentation . . . 1173
8.1188.2.1 bucketSz . . . 1173
8.1188.2.2 peakRate . . . 1173
8.1188.2.3 tokenRate . . . 1173
8.1189 unpack_qos_Tos_t Struct Reference . . . 1173
8.1189.1 Detailed Description . . . 1174
8.1189.2 Field Documentation . . . 1174
8.1189.2.1 mask . . . 1174
8.1189.2.2 val . . . 1174
8.1190 unpack_QosFlowStat_t Struct Reference . . . 1174
8.1190.1 Detailed Description . . . 1174
8.1190.2 Field Documentation . . . 1175
8.1190.2.1 bearerId . . . 1175
8.1190.2.2 tx_bytes . . . 1175
8.1190.2.3 tx_bytes_drp . . . 1175
8.1190.2.4 tx_pkt . . . 1175
8.1190.2.5 tx_pkt_drp . . . 1175
8.1191 unpack_rms_GetSMSWake_t Struct Reference . . . 1175
8.1191.1 Detailed Description . . . 1175
8.1191.2 Field Documentation . . . 1175
8.1191.2.1 enabled . . . 1175
8.1191.2.2 wake_mask . . . 1175
8.1192 unpack_RMTransferStatistics_ind_t Struct Reference . . . 1175
8.1192.1 Detailed Description . . . 1175
8.1192.2 Field Documentation . . . 1176
8.1192.2.1 RxDropConutTlv . . . 1176
8.1192.2.2 RxOkByteCountTlv . . . 1176
8.1192.2.3 RxOkConutTlv . . . 1176
8.1192.2.4 TxDropConutTlv . . . 1176
8.1192.2.5 TxOkByteCountTlv . . . 1176
8.1192.2.6 TxOkConutTlv . . . 1176
8.1193 unpack_sar_SLQSGetRfSarState_t Struct Reference . . . 1176
8.1193.1 Detailed Description . . . 1176
8.1193.2 Field Documentation . . . 1177
8.1193.2.1 pSarRFState . . . 1177
8.1194 unpack_sms_GetSMSCAddress_t Struct Reference . . . 1177

8.1194.1 Detailed Description . . . 1177
8.1194.2 Field Documentation . . . 1177
8.1194.2.1 addressSize . . . 1177
8.1194.2.2 pSMSCAddress . . . 1178
8.1194.2.3 pSMSCType . . . 1178
8.1194.2.4 typeSize . . . 1178
8.1195 qmi_wms_unpack_sms_SaveSMS_t Struct Reference . . . 1178
8.1195.1 Detailed Description . . . 1178
8.1195.2 Field Documentation . . . 1178
8.1195.2.1 pMessageIndex . . . 1178
8.1196 qmi_wms_unpack_sms_SendSMS_t Struct Reference . . . 1178
8.1196.1 Detailed Description . . . 1178
8.1196.2 Field Documentation . . . 1178
8.1196.2.1 messageFailureCode . . . 1178
8.1196.2.2 messageID . . . 1178
8.1197 qmi_wms_unpack_sms_SetNewSMSCallback_ind_t Struct Reference . . . 1178
8.1197.1 Detailed Description . . . 1179
8.1197.2 Field Documentation . . . 1179
8.1197.2.1 ETWSPLMNTlv . . . 1179
8.1197.2.2 ETWSTlv . . . 1179
8.1197.2.3 IMSTlv . . . 1179
8.1197.2.4 MMTlv . . . 1179
8.1197.2.5 NewMMTlv . . . 1179
8.1197.2.6 SMSCTlv . . . 1180
8.1197.2.7 TRMessageTlv . . . 1180
8.1198 qmi_wms_unpack_sms_SetNewSMSCallback_t Struct Reference . . . 1180
8.1199 qmi_wms_unpack_sms_SLQSDeleteSMS_t Struct Reference . . . 1180
8.1200 qmi_wms_unpack_sms_SLQSGetIndicationRegister_t Struct Reference . . . 1180
8.1200.1 Detailed Description . . . 1180
8.1200.2 Field Documentation . . . 1180
8.1200.2.1 pGetIndicationRegInfo . . . 1180
8.1201 qmi_wms_unpack_sms_SLQSGetMessageWaiting_t Struct Reference . . . 1180
8.1201.1 Detailed Description . . . 1180
8.1201.2 Field Documentation . . . 1180
8.1201.2.1 pGetMsgWaitingInfoResp . . . 1180
8.1202 qmi_wms_unpack_sms_SLQSGetSMS_t Struct Reference . . . 1181
8.1202.1 Detailed Description . . . 1181
8.1202.2 Field Documentation . . . 1181
8.1202.2.1 message . . . 1181
8.1202.2.2 messageFormat . . . 1181

8.1202.2.3 messageSize . . . 1181
8.1202.2.4 messageTag . . . 1181
8.1203 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 1181
8.1203.1 Detailed Description . . . 1182
8.1203.2 Field Documentation . . . 1182
8.1203.2.1 pBroadcastConfig . . . 1182
8.1203.2.2 pCDMABroadcastConfig . . . 1182
8.1204 unpack_sms_SLQSGetSMSList_t Struct Reference . . . 1182
8.1204.1 Detailed Description . . . 1182
8.1204.2 Field Documentation . . . 1182
8.1204.2.1 messageList . . . 1182
8.1204.2.2 messageListSize . . . 1182
8.1205 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference . . . 1182
8.1205.1 Detailed Description . . . 1183
8.1205.2 Field Documentation . . . 1183
8.1205.2.1 pGetTransLayerInfo . . . 1183
8.1206 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference . . . 1183
8.1206.1 Detailed Description . . . 1183
8.1206.2 Field Documentation . . . 1183
8.1206.2.1 pGetTransNWRegInfo . . . 1183
8.1207 unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . 1183
8.1208 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 1183
8.1208.1 Detailed Description . . . 1184
8.1208.2 Field Documentation . . . 1184
8.1208.2.1 pMaxStorageSizeResp . . . 1184
8.1209 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference . . . 1184
8.1209.1 Detailed Description . . . 1184
8.1209.2 Field Documentation . . . 1184
8.1209.2.1 pMessageProtocol . . . 1184
8.1210 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference . . . 1184
8.1210.1 Detailed Description . . . 1184
8.1210.2 Field Documentation . . . 1185
8.1210.2.1 pSmsStorage . . . 1185
8.1211 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . 1185
8.1211.1 Detailed Description . . . 1185
8.1211.2 Field Documentation . . . 1185
8.1211.2.1 messageMode . . . 1185
8.1211.2.2 storageType . . . 1185
8.1212 unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . 1185
8.1212.1 Detailed Description . . . 1186

8.1212.2 Field Documentation . . . 1186
8.1212.2.1 fix_rate . . . 1187
8.1212.2.2 fix_rate_reported . . . 1187
8.1212.2.3 fix_type . . . 1187
8.1212.2.4 fix_type_reported . . . 1187
8.1212.2.5 function . . . 1187
8.1212.2.6 function_reported . . . 1187
8.1212.2.7 max_dist . . . 1187
8.1212.2.8 max_dist_reported . . . 1187
8.1212.2.9 max_time . . . 1187
8.1212.2.10 max_time_reported . . . 1187
8.1213 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference . . . 1187
8.1213.1 Detailed Description . . . 1187
8.1213.2 Field Documentation . . . 1187
8.1213.2.1 eventType . . . 1188
8.1213.2.2 SessionInfoConfig . . . 1188
8.1213.2.3 SessionInfoFota . . . 1188
8.1213.2.4 SessionInfoNotification . . . 1188
8.1214 unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 1188
8.1214.1 Detailed Description . . . 1188
8.1214.2 Field Documentation . . . 1190
8.1214.2.1 Date . . . 1190
8.1214.2.2 DateLength . . . 1190
8.1214.2.3 PkgDescLength . . . 1190
8.1214.2.4 PkgDescription . . . 1190
8.1214.2.5 PkgName . . . 1190
8.1214.2.6 PkgNameLength . . . 1190
8.1214.2.7 RetryCount . . . 1190
8.1214.2.8 SessionState . . . 1190
8.1214.2.9 SessionType . . . 1190
8.1214.2.10 Severity . . . 1190
8.1214.2.11 Source . . . 1190
8.1214.2.12 SourceLength . . . 1190
8.1214.2.13 Status . . . 1190
8.1214.2.14 Time . . . 1190
8.1214.2.15 TimeLength . . . 1190
8.1214.2.16 UpdateCompleteStatus . . . 1190
8.1215 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . 1190
8.1215.1 Detailed Description . . . 1191
8.1215.2 Field Documentation . . . 1192

8.1215.2.1 Autosdm . . . 1192
8.1215.2.2 FOTAdownload . . . 1192
8.1215.2.3 FOTAUpdate . . . 1192
8.1215.2.4 FwAutoCheck . . . 1192
8.1215.2.5 OMADMEnabled . . . 1192
8.1216 qmi_wds_unpack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 1192
8.1216.1 Detailed Description . . . 1192
8.1216.2 Field Documentation . . . 1192
8.1216.2.1 FwAvailability . . . 1192
8.1217 qmi_uim_unpack_uim_ChangePin_t Struct Reference . . . 1192
8.1217.1 Detailed Description . . . 1193
8.1217.2 Field Documentation . . . 1193
8.1217.2.1 pEncryptedPIN1 . . . 1193
8.1217.2.2 pIndicationToken . . . 1193
8.1217.2.3 pRemainingRetries . . . 1193
8.1217.2.4 Tlvresult . . . 1193
8.1218 qmi_uim_unpack_uim_GetCardStatus_t Struct Reference . . . 1193
8.1218.1 Detailed Description . . . 1193
8.1218.2 Field Documentation . . . 1194
8.1218.2.1 pCardStatus . . . 1194
8.1218.2.2 pHotSwapStatus . . . 1194
8.1218.2.3 Tlvresult . . . 1194
8.1219 qmi_uim_unpack_uim_ReadTransparent_t Struct Reference . . . 1194
8.1219.1 Detailed Description . . . 1194
8.1219.2 Field Documentation . . . 1194
8.1219.2.1 pCardResult . . . 1194
8.1219.2.2 pEncryptedData . . . 1194
8.1219.2.3 pIndicationToken . . . 1194
8.1219.2.4 pReadResult . . . 1195
8.1219.2.5 Tlvresult . . . 1195
8.1220 qmi_uim_unpack_uim_SetPinProtection_t Struct Reference . . . 1195
8.1220.1 Detailed Description . . . 1195
8.1220.2 Field Documentation . . . 1195
8.1220.2.1 pEncryptedPIN1 . . . 1195
8.1220.2.2 pIndicationToken . . . 1195
8.1220.2.3 pRemainingRetries . . . 1195
8.1220.2.4 Tlvresult . . . 1195
8.1221 qmi_uim_unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference . . . 1195
8.1221.1 Detailed Description . . . 1196
8.1221.2 Field Documentation . . . 1196

8.1221.2.1 bNumberOfPhySlots . . . 1196
8.1221.2.2 slotsstatusChange . . . 1196
8.1222 qmi_uim_SLQSUIMAuthenticate_t Struct Reference . . . 1196
8.1222.1 Detailed Description . . . 1196
8.1222.2 Field Documentation . . . 1196
8.1222.2.1 pAuthenticateResult . . . 1196
8.1222.2.2 pCardResult . . . 1196
8.1222.2.3 pIndicationToken . . . 1197
8.1223 qmi_uim_SLQSUIMDepersonalization_t Struct Reference . . . 1197
8.1223.1 Detailed Description . . . 1197
8.1223.2 Field Documentation . . . 1197
8.1223.2.1 pRemainingRetries . . . 1197
8.1224 qmi_uim_SLQSUIMEventRegister_t Struct Reference . . . 1197
8.1224.1 Detailed Description . . . 1197
8.1224.2 Field Documentation . . . 1197
8.1224.2.1 eventMask . . . 1197
8.1225 qmi_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 1197
8.1225.1 Detailed Description . . . 1198
8.1225.2 Field Documentation . . . 1198
8.1225.2.1 pAutoSelection . . . 1198
8.1225.2.2 pHaltSubscription . . . 1198
8.1225.2.3 pPersonalizationStatus . . . 1198
8.1226 qmi_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 1198
8.1226.1 Detailed Description . . . 1198
8.1226.2 Field Documentation . . . 1199
8.1226.2.1 pCardResult . . . 1199
8.1226.2.2 pFileAttributes . . . 1199
8.1226.2.3 pIndicationToken . . . 1199
8.1227 qmi_uim_SLQSUIMGetSlotsStatus_t Struct Reference . . . 1199
8.1227.1 Detailed Description . . . 1199
8.1227.2 Field Documentation . . . 1199
8.1227.2.1 pNumberOfPhySlot . . . 1199
8.1227.2.2 pUimSlotsStatus . . . 1199
8.1228 qmi_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 1199
8.1228.1 Detailed Description . . . 1199
8.1228.2 Field Documentation . . . 1200
8.1228.2.1 pRefreshEvent . . . 1200
8.1229 qmi_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference . . . 1200
8.1229.1 Detailed Description . . . 1200
8.1229.2 Field Documentation . . . 1200

8.1229.2.1 pCardStatus . . . 1200
8.1230 unpack_uim_UnblockPin_t Struct Reference . . . 1200
8.1230.1 Detailed Description . . . 1200
8.1230.2 Field Documentation . . . 1201
8.1230.2.1 pEncryptedPIN1 . . . 1201
8.1230.2.2 pIndicationToken . . . 1201
8.1230.2.3 pRemainingRetries . . . 1201
8.1230.2.4 Tlvresult . . . 1201
8.1231 unpack_uim_VerifyPin_t Struct Reference . . . 1201
8.1231.1 Detailed Description . . . 1201
8.1231.2 Field Documentation . . . 1201
8.1231.2.1 pEncryptedPIN1 . . . 1201
8.1231.2.2 pIndicationToken . . . 1201
8.1231.2.3 pRemainingRetries . . . 1201
8.1231.2.4 Tlvresult . . . 1201
8.1232 unpack_voice_SLQSOriginateUSSD_t Struct Reference . . . 1201
8.1232.1 Field Documentation . . . 1202
8.1232.1.1 pAlphaIDInfo . . . 1202
8.1232.1.2 pCallId . . . 1202
8.1232.1.3 pCcResultType . . . 1202
8.1232.1.4 pCCSuppsType . . . 1202
8.1232.1.5 pfailureCause . . . 1202
8.1232.1.6 pUSSDInfo . . . 1202
8.1233 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 1202
8.1233.1 Detailed Description . . . 1202
8.1233.2 Field Documentation . . . 1202
8.1233.2.1 pCallId . . . 1202
8.1234 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 1202
8.1234.1 Detailed Description . . . 1202
8.1234.2 Field Documentation . . . 1203
8.1234.2.1 pCallID . . . 1203
8.1235 unpack_voice_SLQSVoiceDialCall_t Struct Reference . . . 1203
8.1235.1 Detailed Description . . . 1203
8.1235.2 Field Documentation . . . 1203
8.1235.2.1 pAlphaIDInfo . . . 1203
8.1235.2.2 pCallID . . . 1204
8.1235.2.3 pCCResultType . . . 1204
8.1235.2.4 pCCSUPSType . . . 1204
8.1236 unpack_voice_SLQSVoiceEndCall_t Struct Reference . . . 1204
8.1236.1 Detailed Description . . . 1204

8.1236.2 Field Documentation . . . 1204
8.1236.2.1 pCallId . . . 1204
8.1237 npack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference . . . 1204
8.1237.1 Detailed Description . . . 1204
8.1237.2 Field Documentation . . . 1206
8.1237.2.1 pArrAlertingPattern . . . 1206
8.1237.2.2 pArrAlertingType . . . 1206
8.1237.2.3 pArrAlphaID . . . 1206
8.1237.2.4 pArrCalledPartyNum . . . 1206
8.1237.2.5 pArrCallEndReason . . . 1206
8.1237.2.6 pArrCallInfo . . . 1206
8.1237.2.7 pArrConnectPartyNum . . . 1206
8.1237.2.8 pArrDiagInfo . . . 1206
8.1237.2.9 pArrRedirPartyNum . . . 1206
8.1237.2.10 pArrRemotePartyName . . . 1206
8.1237.2.11 pArrRemotePartyNum . . . 1206
8.1237.2.12 pArrSvcOption . . . 1207
8.1237.2.13 pArrUUSInfo . . . 1207
8.1237.2.14 pOTASPStatus . . . 1207
8.1237.2.15 pVoicePrivacy . . . 1207
8.1238 npack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 1207
8.1238.1 Detailed Description . . . 1207
8.1238.2 Field Documentation . . . 1208
8.1238.2.1 pAlphaIDInfo . . . 1208
8.1238.2.2 pCallID . . . 1208
8.1238.2.3 pCCResType . . . 1208
8.1238.2.4 pCCSUPSType . . . 1208
8.1238.2.5 pFailCause . . . 1208
8.1238.2.6 pSvcClass . . . 1208
8.1239 npack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 1208
8.1239.1 Detailed Description . . . 1208
8.1239.2 Field Documentation . . . 1209
8.1239.2.1 pAlphaIDInfo . . . 1209
8.1239.2.2 pCallID . . . 1209
8.1239.2.3 pCCResType . . . 1209
8.1239.2.4 pCCSUPSType . . . 1209
8.1239.2.5 pFailCause . . . 1209
8.1239.2.6 pGetCallFWExtInfo . . . 1209
8.1239.2.7 pGetCallFWInfo . . . 1209
8.1240 npack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 1209

8.1240.1 Detailed Description . . . 1210
8.1240.2 Field Documentation . . . 1212
8.1240.2.1 pAlertingPattern . . . 1212
8.1240.2.2 pAlertType . . . 1212
8.1240.2.3 pAlphaIDInfo . . . 1212
8.1240.2.4 pCallInfo . . . 1212
8.1240.2.5 pConnectNumInfo . . . 1212
8.1240.2.6 pDiagInfo . . . 1212
8.1240.2.7 pOTASPStatus . . . 1212
8.1240.2.8 pRemotePartyName . . . 1212
8.1240.2.9 pRemotePartyNum . . . 1212
8.1240.2.10 pSrvOpt . . . 1212
8.1240.2.11 pUUSInfo . . . 1212
8.1240.2.12 pVoicePrivacy . . . 1212
8.1241 unpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 1212
8.1241.1 Detailed Description . . . 1213
8.1241.2 Field Documentation . . . 1213
8.1241.2.1 pAlphaIDInfo . . . 1213
8.1241.2.2 pCallID . . . 1213
8.1241.2.3 pCCResType . . . 1213
8.1241.2.4 pCCSUPSType . . . 1213
8.1241.2.5 pFailCause . . . 1213
8.1241.2.6 pSvcClass . . . 1214
8.1242 unpack_voice_SLQSVoiceGetCLIP_t Struct Reference . . . 1214
8.1242.1 Detailed Description . . . 1214
8.1242.2 Field Documentation . . . 1215
8.1242.2.1 pAlphaIDInfo . . . 1215
8.1242.2.2 pCallID . . . 1215
8.1242.2.3 pCCResType . . . 1215
8.1242.2.4 pCCSUPSType . . . 1215
8.1242.2.5 pCLIPResp . . . 1215
8.1242.2.6 pFailCause . . . 1215
8.1243 unpack_voice_SLQSVoiceGetCLIR_t Struct Reference . . . 1215
8.1243.1 Detailed Description . . . 1215
8.1243.2 Field Documentation . . . 1216
8.1243.2.1 pAlphaIDInfo . . . 1216
8.1243.2.2 pCallID . . . 1216
8.1243.2.3 pCCResType . . . 1216
8.1243.2.4 pCCSUPSType . . . 1216
8.1243.2.5 pCLIRResp . . . 1216

8.1243.2.6 pFailCause . . . 1216
8.1244 npack_voice_SLQSVoiceGetCNAP_t Struct Reference . . . 1216
8.1244.1 Detailed Description . . . 1216
8.1244.2 Field Documentation . . . 1217
8.1244.2.1 pAlphaIDInfo . . . 1217
8.1244.2.2 pCallID . . . 1217
8.1244.2.3 pCCResType . . . 1217
8.1244.2.4 pCCSUPSType . . . 1217
8.1244.2.5 pCNAPResp . . . 1217
8.1244.2.6 pFailCause . . . 1217
8.1245 npack_voice_SLQSVoiceGetCOLP_t Struct Reference . . . 1217
8.1245.1 Detailed Description . . . 1218
8.1245.2 Field Documentation . . . 1218
8.1245.2.1 pAlphaIDInfo . . . 1218
8.1245.2.2 pCallID . . . 1218
8.1245.2.3 pCCResType . . . 1218
8.1245.2.4 pCCSUPSType . . . 1219
8.1245.2.5 pCOLPResp . . . 1219
8.1245.2.6 pFailCause . . . 1219
8.1246 npack_voice_SLQSVoiceGetCOLR_t Struct Reference . . . 1219
8.1246.1 Detailed Description . . . 1219
8.1246.2 Field Documentation . . . 1220
8.1246.2.1 pAlphaIDInfo . . . 1220
8.1246.2.2 pCallID . . . 1220
8.1246.2.3 pCCResType . . . 1220
8.1246.2.4 pCCSUPSType . . . 1220
8.1246.2.5 pCOLRResp . . . 1220
8.1246.2.6 pFailCause . . . 1220
8.1247 npack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 1220
8.1247.1 Detailed Description . . . 1220
8.1247.2 Field Documentation . . . 1221
8.1247.2.1 pAirTimerCnt . . . 1221
8.1247.2.2 pAutoAnswerStat . . . 1221
8.1247.2.3 pCurAMRConfig . . . 1221
8.1247.2.4 pCurPrefVoiceSO . . . 1221
8.1247.2.5 pCurrTTYMode . . . 1221
8.1247.2.6 pCurVoiceDomainPref . . . 1222
8.1247.2.7 pCurVoicePrivacyPref . . . 1222
8.1247.2.8 pRoamTimerCnt . . . 1222
8.1248 npack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 1222

8.1248.1 Detailed Description . . . 1222
8.1248.2 Field Documentation . . . 1222
8.1248.2.1 pFailCause . . . 1222
8.1249 unpack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 1222
8.1249.1 Detailed Description . . . 1222
8.1249.2 Field Documentation . . . 1222
8.1249.2.1 pCallID . . . 1222
8.1250 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 1222
8.1250.1 Detailed Description . . . 1223
8.1250.2 Field Documentation . . . 1223
8.1250.2.1 pAlphaIDInfo . . . 1223
8.1250.2.2 pCallID . . . 1223
8.1250.2.3 pCCResType . . . 1223
8.1250.2.4 pCCSUPSType . . . 1224
8.1250.2.5 pFailCause . . . 1224
8.1251 unpack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 1224
8.1251.1 Detailed Description . . . 1224
8.1251.2 Field Documentation . . . 1225
8.1251.2.1 pAirTimerStatus . . . 1225
8.1251.2.2 pAutoAnsStatus . . . 1225
8.1251.2.3 pPrefVoiceSOStatus . . . 1225
8.1251.2.4 pRoamTimerStatus . . . 1225
8.1251.2.5 pTTYConfigStatus . . . 1225
8.1251.2.6 pVoiceDomainPrefStatus . . . 1225
8.1252 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 1225
8.1252.1 Detailed Description . . . 1225
8.1252.2 Field Documentation . . . 1226
8.1252.2.1 pAlphaIDInfo . . . 1226
8.1252.2.2 pCallID . . . 1226
8.1252.2.3 pCCResultType . . . 1226
8.1252.2.4 pCCSUPSType . . . 1226
8.1252.2.5 pFailCause . . . 1226
8.1253 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 1226
8.1253.1 Detailed Description . . . 1226
8.1253.2 Field Documentation . . . 1227
8.1253.2.1 pCallID . . . 1227
8.1254 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 1227
8.1254.1 Detailed Description . . . 1227
8.1254.2 Field Documentation . . . 1227
8.1254.2.1 callID . . . 1227

8.1255 npack_wds_DHCPv4ClientLease_ind_t Struct Reference . . . 1227
8.1255.1 Detailed Description . . . 1227
8.1255.2 Field Documentation . . . 1228
8.1255.2.1 DHCPv4LeaseOptTlv . . . 1228
8.1255.2.2 DHCPv4LeaseStateTlv . . . 1228
8.1255.2.3 IPv4AddrTlv . . . 1228
8.1255.2.4 ProfileIdTlv . . . 1228
8.1256 npack_wds_GetAutoconnect_t Struct Reference . . . 1228
8.1256.1 Detailed Description . . . 1228
8.1256.2 Field Documentation . . . 1228
8.1256.2.1 psetting . . . 1228
8.1257 npack_wds_GetByteTotals_t Struct Reference . . . 1228
8.1257.1 Detailed Description . . . 1228
8.1257.2 Field Documentation . . . 1229
8.1257.2.1 pRXTotalBytes . . . 1229
8.1257.2.2 pTXTotalBytes . . . 1229
8.1258 npack_wds_GetConnectionRate_t Struct Reference . . . 1229
8.1258.1 Detailed Description . . . 1229
8.1258.2 Field Documentation . . . 1229
8.1258.2.1 currentChannelRXRate . . . 1229
8.1258.2.2 currentChannelTXRate . . . 1229
8.1258.2.3 maxChannelRXRate . . . 1229
8.1258.2.4 maxChannelTXRate . . . 1229
8.1259 npack_wds_GetDataBearerTechnology_t Struct Reference . . . 1229
8.1259.1 Detailed Description . . . 1230
8.1259.2 Field Documentation . . . 1230
8.1259.2.1 pDataBearer . . . 1230
8.1260 npack_wds_GetDefaultProfile_t Struct Reference . . . 1230
8.1260.1 Detailed Description . . . 1231
8.1260.2 Field Documentation . . . 1232
8.1260.2.1 apnname . . . 1232
8.1260.2.2 apnsize . . . 1232
8.1260.2.3 auth . . . 1232
8.1260.2.4 ipaddr . . . 1232
8.1260.2.5 ipaddrv6 . . . 1232
8.1260.2.6 name . . . 1232
8.1260.2.7 namesize . . . 1232
8.1260.2.8 pdptype . . . 1232
8.1260.2.9 pridns . . . 1232
8.1260.2.10 pridnsv6 . . . 1232

8.1260.2.1 secdns . . . 1232
8.1260.2.2 secdnsv6 . . . 1232
8.1260.2.3 username . . . 1232
8.1260.2.4 usersize . . . 1232
8.1261 unpack_wds_GetDefaultProfileNum_t Struct Reference . . . 1232
8.1261.1 Detailed Description . . . 1232
8.1261.2 Field Documentation . . . 1233
8.1261.2.1 index . . . 1233
8.1262 unpack_wds_GetDormancyState_t Struct Reference . . . 1233
8.1262.1 Detailed Description . . . 1233
8.1262.2 Field Documentation . . . 1233
8.1262.2.1 dormancyState . . . 1233
8.1263 unpack_wds_GetLastMobileIPError_t Struct Reference . . . 1233
8.1263.1 Detailed Description . . . 1233
8.1263.2 Field Documentation . . . 1233
8.1263.2.1 error . . . 1233
8.1264 unpack_wds_GetMobileIP_t Struct Reference . . . 1234
8.1264.1 Detailed Description . . . 1234
8.1264.2 Field Documentation . . . 1234
8.1264.2.1 mipMode . . . 1234
8.1265 unpack_wds_GetMobileIPProfile_t Struct Reference . . . 1234
8.1265.1 Detailed Description . . . 1234
8.1265.2 Field Documentation . . . 1235
8.1265.2.1 AAASPI . . . 1235
8.1265.2.2 AAAState . . . 1235
8.1265.2.3 address . . . 1235
8.1265.2.4 enabled . . . 1235
8.1265.2.5 HASPI . . . 1236
8.1265.2.6 HAState . . . 1236
8.1265.2.7 NAI . . . 1236
8.1265.2.8 naiSize . . . 1236
8.1265.2.9 primaryHA . . . 1236
8.1265.2.10 revTunneling . . . 1236
8.1265.2.11 secondaryHA . . . 1236
8.1266 unpack_wds_GetPacketStatistics_t Struct Reference . . . 1236
8.1266.1 Detailed Description . . . 1236
8.1266.2 Field Documentation . . . 1237
8.1266.2.1 pRXDroppedCount . . . 1237
8.1266.2.2 pRXOkBytesCount . . . 1237
8.1266.2.3 pRXOKBytesLastCall . . . 1237

8.1266.2.4 pRXPacketErrors . . . 1237
8.1266.2.5 pRXPacketOverflows . . . 1237
8.1266.2.6 pRXPacketSuccesses . . . 1237
8.1266.2.7 pTXDroppedCount . . . 1237
8.1266.2.8 pTXOkBytesCount . . . 1237
8.1266.2.9 pTXOKBytesLastCall . . . 1237
8.1266.2.10 pTXPacketErrors . . . 1237
8.1266.2.11 pTXPacketOverflows . . . 1237
8.1266.2.12 pTXPacketSuccesses . . . 1237
8.1267 unpack_wds_GetPacketStatus_t Struct Reference . . . 1237
8.1267.1 Detailed Description . . . 1238
8.1267.2 Field Documentation . . . 1239
8.1267.2.1 rXDroppedCount . . . 1239
8.1267.2.2 rXOkBytesCount . . . 1239
8.1267.2.3 rXOKBytesLastCall . . . 1239
8.1267.2.4 rXPacketErrors . . . 1239
8.1267.2.5 rXPacketOverflows . . . 1239
8.1267.2.6 rXPacketSuccesses . . . 1239
8.1267.2.7 tXDroppedCount . . . 1239
8.1267.2.8 tXOkBytesCount . . . 1239
8.1267.2.9 tXOKBytesLastCall . . . 1239
8.1267.2.10 tXPacketErrors . . . 1239
8.1267.2.11 tXPacketOverflows . . . 1239
8.1267.2.12 tXPacketSuccesses . . . 1239
8.1268 unpack_wds_GetSessionDuration_t Struct Reference . . . 1239
8.1268.1 Detailed Description . . . 1239
8.1268.2 Field Documentation . . . 1239
8.1268.2.1 callDuration . . . 1239
8.1269 unpack_wds_GetSessionState_t Struct Reference . . . 1240
8.1269.1 Detailed Description . . . 1240
8.1269.2 Field Documentation . . . 1240
8.1269.2.1 connectionStatus . . . 1240
8.1270 unpack_wds_RMSetTransferStatistics_t Struct Reference . . . 1240
8.1270.1 Detailed Description . . . 1240
8.1271 unpack_wds_SetMobileIPProfile_t Struct Reference . . . 1240
8.1271.1 Detailed Description . . . 1240
8.1272 unpack_wds_SLQSCreateProfile_t Struct Reference . . . 1240
8.1272.1 Detailed Description . . . 1240
8.1272.2 Field Documentation . . . 1241
8.1272.2.1 pCreateProfileOut . . . 1241

8.1272.2.2 pProfileID . . . 1241
8.1272.2.3 Tlvresult . . . 1241
8.1273 npack_wds_SLQSDeleteProfile_t Struct Reference . . . 1241
8.1273.1 Detailed Description . . . 1241
8.1273.2 Field Documentation . . . 1241
8.1273.2.1 extendedErrorCode . . . 1241
8.1274 npack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . 1241
8.1274.1 Detailed Description . . . 1241
8.1274.2 Field Documentation . . . 1242
8.1274.2.1 _3gppRelease . . . 1242
8.1274.2.2 defaultPDNEnabled . . . 1242
8.1274.2.3 LTEAttachProfile . . . 1243
8.1274.2.4 LTEAttachProfileList . . . 1243
8.1274.2.5 LTEAttachProfileListLen . . . 1243
8.1274.2.6 profileList . . . 1243
8.1275 npack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 1243
8.1275.1 Detailed Description . . . 1243
8.1275.2 Field Documentation . . . 1243
8.1275.2.1 currNetworkInfo . . . 1243
8.1275.2.2 networkInfoLen . . . 1243
8.1275.2.3 prefNetwork . . . 1243
8.1276 npack_wds_SLQSGetCurrentChannelRate_t Struct Reference . . . 1243
8.1276.1 Detailed Description . . . 1244
8.1276.2 Field Documentation . . . 1244
8.1276.2.1 current_channel_rx_rate . . . 1244
8.1276.2.2 current_channel_tx_rate . . . 1244
8.1276.2.3 max_channel_rx_rate . . . 1244
8.1276.2.4 max_channel_tx_rate . . . 1244
8.1277 npack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 1244
8.1277.1 Detailed Description . . . 1244
8.1277.2 Field Documentation . . . 1245
8.1277.2.1 curDataBearerTechnology . . . 1245
8.1277.2.2 dataBearerMask . . . 1245
8.1277.2.3 lastCallDataBearerTechnology . . . 1245
8.1278 npack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 1245
8.1278.1 Detailed Description . . . 1245
8.1278.2 Field Documentation . . . 1247
8.1278.2.1 callEndReason . . . 1247
8.1278.2.2 channelRate . . . 1247
8.1278.2.3 connectionStatus . . . 1247

8.1278.2.4 dataBearerTech . . . 1247
8.1278.2.5 dormancyStatus . . . 1247
8.1278.2.6 lastCallDataBearerTech . . . 1247
8.1278.2.7 lastCallRXOKBytesCnt . . . 1247
8.1278.2.8 lastCallTXOKBytesCnt . . . 1247
8.1278.2.9 mdmCallDurationActive . . . 1247
8.1278.2.10 rxOKBytesCount . . . 1248
8.1278.2.11 txOKBytesCount . . . 1248
8.1279 qmi_wds_SLQSGetProfileSettings_t Struct Reference . . . 1248
8.1279.1 Detailed Description . . . 1248
8.1279.2 Field Documentation . . . 1248
8.1279.2.1 pProfileSettings . . . 1248
8.1279.2.2 ProfileType . . . 1248
8.1279.2.3 Tlvresult . . . 1248
8.1280 qmi_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 1248
8.1280.1 Detailed Description . . . 1249
8.1280.2 Field Documentation . . . 1249
8.1280.2.1 APNName . . . 1250
8.1280.2.2 Authentication . . . 1250
8.1280.2.3 DomainList . . . 1250
8.1280.2.4 GPRSGrantedQoS . . . 1250
8.1280.2.5 GWAddressV4 . . . 1250
8.1280.2.6 IMCNflag . . . 1250
8.1280.2.7 IPFamilyPreference . . . 1250
8.1280.2.8 IPv4 . . . 1250
8.1280.2.9 IPV6AddrInfo . . . 1250
8.1280.2.10 IPV6GWAddrInfo . . . 1250
8.1280.2.11 Mtu . . . 1250
8.1280.2.12 PCSCFAddrPCO . . . 1250
8.1280.2.13 PCSCFFQDNAddrList . . . 1250
8.1280.2.14 PDPType . . . 1250
8.1280.2.15 PrimaryDNSV4 . . . 1250
8.1280.2.16 PrimaryDNSV6 . . . 1250
8.1280.2.17 ProfileID . . . 1250
8.1280.2.18 ProfileName . . . 1250
8.1280.2.19 SecondaryDNSV4 . . . 1250
8.1280.2.20 SecondaryDNSV6 . . . 1250
8.1280.2.21 ServerAddrList . . . 1250
8.1280.2.22 SubnetMaskV4 . . . 1250
8.1280.2.23 Technology . . . 1250

8.1280.2.24 MTSGrantedQoS . . . 1250
8.1280.2.25 username . . . 1250
8.1281 unpack_wds_SLQSModifyProfile_t Struct Reference . . . 1250
8.1281.1 Detailed Description . . . 1251
8.1281.2 Field Documentation . . . 1251
8.1281.2.1 pExtErrorCode . . . 1251
8.1282 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 1251
8.1282.1 Detailed Description . . . 1251
8.1282.2 Field Documentation . . . 1251
8.1282.2.1 Tlvresult . . . 1251
8.1283 unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference . . . 1251
8.1283.1 Detailed Description . . . 1251
8.1283.2 Field Documentation . . . 1252
8.1283.2.1 bearerID . . . 1252
8.1283.2.2 conn_status . . . 1252
8.1283.2.3 ipFamily . . . 1252
8.1283.2.4 reconfigReqd . . . 1252
8.1283.2.5 sessionEndReason . . . 1252
8.1283.2.6 techName . . . 1252
8.1283.2.7 verboseSessnEndReason . . . 1252
8.1283.2.8 verboseSessnEndReasonType . . . 1252
8.1284 unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference . . . 1252
8.1284.1 Detailed Description . . . 1253
8.1284.2 Field Documentation . . . 1253
8.1284.2.1 currDBTechAvail . . . 1253
8.1284.2.2 currNWInfo . . . 1253
8.1284.2.3 dataSysStatAvail . . . 1253
8.1284.2.4 dBTechAvail . . . 1253
8.1284.2.5 dBtechExtAvail . . . 1253
8.1284.2.6 dBTechExtRatValue . . . 1253
8.1284.2.7 dBTechExtSoMask . . . 1253
8.1284.2.8 dBTechnology . . . 1253
8.1284.2.9 dBtechnologyExt . . . 1254
8.1284.2.10 dormancyStatAvail . . . 1254
8.1284.2.11 dormancyStatus . . . 1254
8.1284.2.12 ipstatAvail . . . 1254
8.1284.2.13 ipStatus . . . 1254
8.1284.2.14 netInfoLen . . . 1254
8.1284.2.15 prefNetwork . . . 1254
8.1284.2.16 ratMask . . . 1254

8.1284.2.17 tx_bytes . . . 1254
8.1284.2.18 tx_pkts . . . 1254
8.1284.2.19 wdsMask . . . 1254
8.1284.2.20 rx_bytes . . . 1254
8.1284.2.21 rx_pkts . . . 1254
8.1284.2.22 perStatAvail . . . 1254
8.1285 qmi_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 1254
8.1285.1 Detailed Description . . . 1254
8.1285.2 Field Documentation . . . 1254
8.1285.2.1 pHwConfig . . . 1254
8.1285.2.2 pRequestOptionList . . . 1254
8.1286 qmi_wds_SLQSSGetLoopback_t Struct Reference . . . 1255
8.1286.1 Detailed Description . . . 1255
8.1286.2 Field Documentation . . . 1255
8.1286.2.1 byteLoopbackMode . . . 1255
8.1286.2.2 byteLoopbackMultiplier . . . 1255
8.1287 qmi_wds_SLQSStartDataSession_t Struct Reference . . . 1255
8.1287.1 Detailed Description . . . 1255
8.1287.2 Field Documentation . . . 1256
8.1287.2.1 pFailureReason . . . 1256
8.1287.2.2 psid . . . 1256
8.1287.2.3 pVerboseFailReasonType . . . 1256
8.1287.2.4 pVerboseFailureReason . . . 1256
8.1288 qmi_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 1256
8.1288.1 Detailed Description . . . 1256
8.1288.2 Field Documentation . . . 1257
8.1288.2.1 apnName . . . 1257
8.1288.2.2 bearerId . . . 1257
8.1288.2.3 contextId . . . 1257
8.1288.2.4 ipv4Address . . . 1257
8.1288.2.5 ipv4GWAddress . . . 1258
8.1288.2.6 ipv6Address . . . 1258
8.1288.2.7 ipv6GWAddress . . . 1258
8.1288.2.8 prDNSIPv4Address . . . 1258
8.1288.2.9 prDNSIPv6Address . . . 1258
8.1288.2.10 prPCSCFIPv4Address . . . 1258
8.1288.2.11 prPCSCFIPv6Address . . . 1258
8.1288.2.12 seDNSIPv4Address . . . 1258
8.1288.2.13 seDNSIPv6Address . . . 1258
8.1288.2.14 sePCSCFIPv4Address . . . 1258

8.1288.2.15 ePCSCFIPv6Address . . . 1258
8.1289 UnPackGetProfileSettingOut Struct Reference . . . 1258
8.1289.1 Detailed Description . . . 1258
8.1289.2 Field Documentation . . . 1258
8.1289.2.1 curProfile . . . 1258
8.1289.2.2 pExtErrCode . . . 1258
8.1290 unpackWdsProfileParam Union Reference . . . 1258
8.1290.1 Detailed Description . . . 1259
8.1290.2 Field Documentation . . . 1259
8.1290.2.1 SlqsProfile3GPP . . . 1259
8.1290.2.2 SlqsProfile3GPP2 . . . 1259
8.1291 UrlAddr Struct Reference . . . 1259
8.1291.1 Detailed Description . . . 1259
8.1291.2 Field Documentation . . . 1259
8.1291.2.1 address . . . 1259
8.1291.2.2 TlvPresent . . . 1259
8.1292 USBCompConfig Struct Reference . . . 1259
8.1292.1 Detailed Description . . . 1260
8.1292.2 Field Documentation . . . 1260
8.1292.2.1 pUSBComp . . . 1260
8.1293 USBCompParams Struct Reference . . . 1260
8.1293.1 Detailed Description . . . 1260
8.1293.2 Field Documentation . . . 1262
8.1293.2.1 pNumSupUSBComps . . . 1262
8.1293.2.2 pSupUSBComps . . . 1262
8.1293.2.3 pUSBComp . . . 1262
8.1294 USSDNoWaitIndicationInfo Struct Reference . . . 1262
8.1294.1 Detailed Description . . . 1262
8.1294.2 Field Documentation . . . 1263
8.1294.2.1 pAlphaIdentifier . . . 1263
8.1294.2.2 pError . . . 1263
8.1294.2.3 pFailureCause . . . 1263
8.1294.2.4 pUSSDData . . . 1263
8.1295 USSDRespFNetwork Struct Reference . . . 1263
8.1295.1 Detailed Description . . . 1263
8.1295.2 Field Documentation . . . 1263
8.1295.2.1 pRespData . . . 1263
8.1295.2.2 pTypeCode . . . 1263
8.1296 USSInfo Struct Reference . . . 1263
8.1296.1 Detailed Description . . . 1264

8.1296.2 Field Documentation . . . 1264
8.1296.2.1 ussData . . . 1264
8.1296.2.2 ussDCS . . . 1264
8.1296.2.3 ussLen . . . 1264
8.1297 USSResp Struct Reference . . . 1264
8.1297.1 Field Documentation . . . 1264
8.1297.1.1 pAlphaIDInfo . . . 1264
8.1297.1.2 pCallId . . . 1264
8.1297.1.3 pCcResultType . . . 1264
8.1297.1.4 pCCSuppsType . . . 1264
8.1297.1.5 pfailureCause . . . 1265
8.1297.1.6 pUSSDInfo . . . 1265
8.1298 UUSInfo Struct Reference . . . 1265
8.1298.1 Detailed Description . . . 1265
8.1298.2 Field Documentation . . . 1266
8.1298.2.1 UUSData . . . 1266
8.1298.2.2 UUSDatalen . . . 1266
8.1298.2.3 UUSDcs . . . 1266
8.1298.2.4 UUSType . . . 1266
8.1299 VerifyUIMPIN Struct Reference . . . 1266
8.1299.1 Detailed Description . . . 1266
8.1299.2 Field Documentation . . . 1266
8.1299.2.1 pinID . . . 1266
8.1299.2.2 pinLen . . . 1266
8.1299.2.3 pinVal . . . 1266
8.1300 Voice_airTimer Struct Reference . . . 1266
8.1300.1 Detailed Description . . . 1267
8.1300.2 Field Documentation . . . 1267
8.1300.2.1 airTimerValue . . . 1267
8.1300.2.2 namID . . . 1267
8.1301 Voice_allCallsAlphaIDInfo Struct Reference . . . 1267
8.1301.1 Detailed Description . . . 1267
8.1301.2 Field Documentation . . . 1267
8.1301.2.1 AlphaIDInfo . . . 1267
8.1301.2.2 callID . . . 1267
8.1302 Voice_allCallsDiagInfo Struct Reference . . . 1267
8.1302.1 Detailed Description . . . 1268
8.1302.2 Field Documentation . . . 1268
8.1302.2.1 callID . . . 1268
8.1302.2.2 DiagInfo . . . 1268

8.1303 voice_allCallsUUSInfo Struct Reference . . . 1268
8.1303.1 Detailed Description . . . 1268
8.1303.2 Field Documentation . . . 1268
8.1303.2.1 callID . . . 1268
8.1303.2.2 uusInfo . . . 1268
8.1304 voice_alphaIDInfo Struct Reference . . . 1268
8.1304.1 Detailed Description . . . 1269
8.1304.2 Field Documentation . . . 1269
8.1304.2.1 alphaDcs . . . 1269
8.1304.2.2 alphaLen . . . 1269
8.1304.2.3 alphaText . . . 1269
8.1305 voice_arrAlertingPattern Struct Reference . . . 1269
8.1305.1 Detailed Description . . . 1269
8.1305.2 Field Documentation . . . 1270
8.1305.2.1 alertingPattern . . . 1270
8.1305.2.2 callID . . . 1270
8.1305.2.3 numInstances . . . 1270
8.1306 voice_arrAlertingType Struct Reference . . . 1270
8.1306.1 Detailed Description . . . 1270
8.1306.2 Field Documentation . . . 1271
8.1306.2.1 AlertingType . . . 1271
8.1306.2.2 callID . . . 1271
8.1306.2.3 numInstances . . . 1271
8.1307 voice_arrAlphaID Struct Reference . . . 1271
8.1307.1 Detailed Description . . . 1271
8.1307.2 Field Documentation . . . 1271
8.1307.2.1 allCallsAlphaIDInfoArr . . . 1271
8.1307.2.2 numInstances . . . 1271
8.1308 voice_arrCalledPartyNum Struct Reference . . . 1271
8.1308.1 Detailed Description . . . 1271
8.1308.2 Field Documentation . . . 1272
8.1308.2.1 CalledPartyNum . . . 1272
8.1308.2.2 numInstances . . . 1272
8.1309 voice_arrCallEndReason Struct Reference . . . 1272
8.1309.1 Detailed Description . . . 1272
8.1309.2 Field Documentation . . . 1272
8.1309.2.1 callEndReason . . . 1272
8.1309.2.2 callID . . . 1272
8.1309.2.3 numInstances . . . 1272
8.1310 voice_arrCallInfo Struct Reference . . . 1272

8.1310.1 Detailed Description . . . 1273
8.1310.2 Field Documentation . . . 1273
8.1310.2.1 getAllCallInfo . . . 1273
8.1310.2.2 numInstances . . . 1273
8.1311 voice_arrConnectPartyNum Struct Reference . . . 1273
8.1311.1 Detailed Description . . . 1273
8.1311.2 Field Documentation . . . 1273
8.1311.2.1 ConnectedPartyNum . . . 1273
8.1311.2.2 numInstances . . . 1273
8.1312 voice_arrDiagInfo Struct Reference . . . 1274
8.1312.1 Detailed Description . . . 1274
8.1312.2 Field Documentation . . . 1274
8.1312.2.1 DiagInfo . . . 1274
8.1312.2.2 numInstances . . . 1274
8.1313 voice_arrRedirPartyNum Struct Reference . . . 1274
8.1313.1 Detailed Description . . . 1274
8.1313.2 Field Documentation . . . 1274
8.1313.2.1 numInstances . . . 1275
8.1313.2.2 RedirPartyNum . . . 1275
8.1314 voice_arrRemotePartyName Struct Reference . . . 1275
8.1314.1 Detailed Description . . . 1275
8.1314.2 Field Documentation . . . 1275
8.1314.2.1 GetAllCallRmtPtyName . . . 1275
8.1314.2.2 numInstances . . . 1275
8.1315 voice_arrRemotePartyNum Struct Reference . . . 1275
8.1315.1 Detailed Description . . . 1275
8.1315.2 Field Documentation . . . 1276
8.1315.2.1 numInstances . . . 1276
8.1315.2.2 RmtPtyNum . . . 1276
8.1316 voice_arrSvcOption Struct Reference . . . 1276
8.1316.1 Detailed Description . . . 1276
8.1316.2 Field Documentation . . . 1276
8.1316.2.1 callID . . . 1276
8.1316.2.2 numInstances . . . 1276
8.1316.2.3 srvOption . . . 1276
8.1317 voice_arrUUSInfo Struct Reference . . . 1276
8.1317.1 Detailed Description . . . 1276
8.1317.2 Field Documentation . . . 1277
8.1317.2.1 AllCallsUUSInfo . . . 1277
8.1317.2.2 numInstances . . . 1277

8.1318 voice_burstDTMFInfo Struct Reference . . . 1277
8.1318.1 Detailed Description . . . 1277
8.1318.2 Field Documentation . . . 1277
8.1318.2.1 digitCnt . . . 1277
8.1318.2.2 pCallID . . . 1277
8.1318.2.3 pDigitBuff . . . 1277
8.1319 voice_calledPartySubAdd Struct Reference . . . 1278
8.1319.1 Detailed Description . . . 1278
8.1319.2 Field Documentation . . . 1278
8.1319.2.1 extBit . . . 1278
8.1319.2.2 oddEvenInd . . . 1278
8.1319.2.3 subAddr . . . 1278
8.1319.2.4 subAddrLen . . . 1278
8.1319.2.5 subAddrType . . . 1278
8.1320 voice_callFwdTypeAndPlan Struct Reference . . . 1278
8.1320.1 Detailed Description . . . 1279
8.1320.2 Field Documentation . . . 1279
8.1320.2.1 numberPlan . . . 1279
8.1320.2.2 numberType . . . 1279
8.1321 voice_callFWExtInfo Struct Reference . . . 1279
8.1321.1 Detailed Description . . . 1280
8.1321.2 Field Documentation . . . 1281
8.1321.2.1 noReplyTimer . . . 1281
8.1321.2.2 number . . . 1281
8.1321.2.3 numLen . . . 1281
8.1321.2.4 numPlan . . . 1281
8.1321.2.5 numType . . . 1281
8.1321.2.6 PI . . . 1281
8.1321.2.7 SI . . . 1281
8.1321.2.8 SvcClass . . . 1281
8.1321.2.9 SvcStatus . . . 1281
8.1322 voice_callFWInfo Struct Reference . . . 1281
8.1322.1 Detailed Description . . . 1282
8.1322.2 Field Documentation . . . 1282
8.1322.2.1 noReplyTimer . . . 1282
8.1322.2.2 number . . . 1282
8.1322.2.3 numLen . . . 1282
8.1322.2.4 SvcClass . . . 1282
8.1322.2.5 SvcStatus . . . 1282
8.1323 voice_callInfo Struct Reference . . . 1282

8.1323.1 Detailed Description . . . 1282
8.1323.2 Field Documentation . . . 1283
8.1323.2.1 callID . . . 1283
8.1323.2.2 callState . . . 1284
8.1323.2.3 callType . . . 1284
8.1323.2.4 direction . . . 1284
8.1323.2.5 mode . . . 1284
8.1324 voice_ccSUPSType Struct Reference . . . 1284
8.1324.1 Detailed Description . . . 1284
8.1324.2 Field Documentation . . . 1284
8.1324.2.1 reason . . . 1284
8.1324.2.2 svcType . . . 1284
8.1325 voice_CLIPResp Struct Reference . . . 1284
8.1325.1 Detailed Description . . . 1285
8.1325.2 Field Documentation . . . 1285
8.1325.2.1 ActiveStatus . . . 1285
8.1325.2.2 ProvisionStatus . . . 1285
8.1326 voice_CLIRResp Struct Reference . . . 1285
8.1326.1 Detailed Description . . . 1285
8.1326.2 Field Documentation . . . 1286
8.1326.2.1 ActiveStatus . . . 1286
8.1326.2.2 ProvisionStatus . . . 1286
8.1327 voice_CNAPResp Struct Reference . . . 1286
8.1327.1 Detailed Description . . . 1286
8.1327.2 Field Documentation . . . 1286
8.1327.2.1 ActiveStatus . . . 1286
8.1327.2.2 ProvisionStatus . . . 1286
8.1328 voice_COLPResp Struct Reference . . . 1287
8.1328.1 Detailed Description . . . 1287
8.1328.2 Field Documentation . . . 1287
8.1328.2.1 ActiveStatus . . . 1287
8.1328.2.2 ProvisionStatus . . . 1287
8.1329 voice_COLRResp Struct Reference . . . 1287
8.1329.1 Detailed Description . . . 1287
8.1329.2 Field Documentation . . . 1288
8.1329.2.1 ActiveStatus . . . 1288
8.1329.2.2 ProvisionStatus . . . 1288
8.1330 voice_connectNumInfo Struct Reference . . . 1288
8.1330.1 Detailed Description . . . 1288
8.1330.2 Field Documentation . . . 1289

8.1330.2.1 callerID . . . 1289
8.1330.2.2 callerIDLen . . . 1289
8.1330.2.3 numPlan . . . 1289
8.1330.2.4 numPresInd . . . 1290
8.1330.2.5 numType . . . 1290
8.1330.2.6 screeningInd . . . 1290
8.1331 voice_CUGInfo Struct Reference . . . 1290
8.1331.1 Detailed Description . . . 1290
8.1331.2 Field Documentation . . . 1290
8.1331.2.1 CUGIndex . . . 1290
8.1331.2.2 SuppOA . . . 1290
8.1331.2.3 SuppPrefCUG . . . 1290
8.1332 voice_curAMRConfig Struct Reference . . . 1290
8.1332.1 Detailed Description . . . 1290
8.1332.2 Field Documentation . . . 1291
8.1332.2.1 gsmAmrStat . . . 1291
8.1332.2.2 wcdmaAmrStat . . . 1291
8.1333 voice_diagInfo Struct Reference . . . 1291
8.1333.1 Detailed Description . . . 1291
8.1333.2 Field Documentation . . . 1291
8.1333.2.1 diagInfoLen . . . 1291
8.1333.2.2 diagnosticInfo . . . 1291
8.1334 voice_DTMFLengths Struct Reference . . . 1292
8.1334.1 Detailed Description . . . 1292
8.1334.2 Field Documentation . . . 1292
8.1334.2.1 DTMFInterdigitInterval . . . 1292
8.1334.2.2 DTMFPulseWidth . . . 1292
8.1335 voice_getAllCallInformation Struct Reference . . . 1292
8.1335.1 Detailed Description . . . 1292
8.1335.2 Field Documentation . . . 1293
8.1335.2.1 ALS . . . 1293
8.1335.2.2 CallInfo . . . 1293
8.1335.2.3 isEmpty . . . 1293
8.1336 voice_getAllCallRmtPtyName Struct Reference . . . 1293
8.1336.1 Detailed Description . . . 1293
8.1336.2 Field Documentation . . . 1293
8.1336.2.1 callID . . . 1293
8.1336.2.2 RemotePartyName . . . 1293
8.1337 voice_getAllCallRmtPtyNum Struct Reference . . . 1293
8.1337.1 Detailed Description . . . 1294

8.1337.2 Field Documentation . . . 1294
8.1337.2.1 callID . . . 1294
8.1337.2.2 RemotePartyNum . . . 1294
8.1338 voice_getCallFWExtInfo Struct Reference . . . 1294
8.1338.1 Detailed Description . . . 1294
8.1338.2 Field Documentation . . . 1294
8.1338.2.1 CallFWExtInfo . . . 1294
8.1338.2.2 numInstances . . . 1294
8.1339 voice_getCallFWInfo Struct Reference . . . 1295
8.1339.1 Detailed Description . . . 1295
8.1339.2 Field Documentation . . . 1295
8.1339.2.1 CallFWInfo . . . 1295
8.1339.2.2 numInstances . . . 1295
8.1340 voice_peerNumberInfo Struct Reference . . . 1295
8.1340.1 Detailed Description . . . 1295
8.1340.2 Field Documentation . . . 1297
8.1340.2.1 callID . . . 1297
8.1340.2.2 number . . . 1297
8.1340.2.3 numLen . . . 1297
8.1340.2.4 numPI . . . 1297
8.1340.2.5 numPlan . . . 1297
8.1340.2.6 numSI . . . 1297
8.1340.2.7 numType . . . 1297
8.1341 voice_prefVoiceSO Struct Reference . . . 1297
8.1341.1 Detailed Description . . . 1297
8.1341.2 Field Documentation . . . 1299
8.1341.2.1 evrcCapability . . . 1299
8.1341.2.2 homeOrigVoiceSO . . . 1299
8.1341.2.3 homePageVoiceSO . . . 1299
8.1341.2.4 namID . . . 1299
8.1341.2.5 roamOrigVoiceSO . . . 1299
8.1342 voice_remotePartyName Struct Reference . . . 1299
8.1342.1 Detailed Description . . . 1299
8.1342.2 Field Documentation . . . 1299
8.1342.2.1 callerName . . . 1300
8.1342.2.2 codingScheme . . . 1300
8.1342.2.3 nameLen . . . 1300
8.1342.2.4 namePI . . . 1300
8.1343 voice_remotePartyNum Struct Reference . . . 1300
8.1343.1 Detailed Description . . . 1300

8.1343.2 Field Documentation . . . 1300
8.1343.2.1 numLen . . . 1300
8.1343.2.2 presentationInd . . . 1300
8.1343.2.3 remPartyNumber . . . 1300
8.1344 voice_roamTimer Struct Reference . . . 1300
8.1344.1 Detailed Description . . . 1301
8.1344.2 Field Documentation . . . 1301
8.1344.2.1 namID . . . 1301
8.1344.2.2 roamTimerValue . . . 1301
8.1345 voice_USSInfo Struct Reference . . . 1301
8.1345.1 Detailed Description . . . 1301
8.1345.2 Field Documentation . . . 1301
8.1345.2.1 ussData . . . 1302
8.1345.2.2 ussDCS . . . 1302
8.1345.2.3 ussLen . . . 1302
8.1346 voice_UUSInfo Struct Reference . . . 1302
8.1346.1 Detailed Description . . . 1302
8.1346.2 Field Documentation . . . 1303
8.1346.2.1 UUSData . . . 1303
8.1346.2.2 UUSDatalen . . . 1303
8.1346.2.3 UUSDcs . . . 1303
8.1346.2.4 UUSType . . . 1303
8.1347 voiceALSSelectLineInfo Struct Reference . . . 1303
8.1347.1 Detailed Description . . . 1303
8.1347.2 Field Documentation . . . 1303
8.1347.2.1 lineValue . . . 1303
8.1348 voiceALSSetLineSwitchInfo Struct Reference . . . 1303
8.1348.1 Detailed Description . . . 1303
8.1348.2 Field Documentation . . . 1303
8.1348.2.1 switchOption . . . 1304
8.1349 voiceAnswerCall Struct Reference . . . 1304
8.1349.1 Detailed Description . . . 1304
8.1349.2 Field Documentation . . . 1304
8.1349.2.1 pCallId . . . 1304
8.1350 voiceBindSubscriptionInfo Struct Reference . . . 1304
8.1350.1 Detailed Description . . . 1304
8.1350.2 Field Documentation . . . 1304
8.1350.2.1 subsType . . . 1304
8.1351 voiceBurstDTMFInfo Struct Reference . . . 1304
8.1351.1 Detailed Description . . . 1305

8.1351.2 Field Documentation . . . 1305
8.1351.2.1 BurstDTMFInfo . . . 1305
8.1351.2.2 pBurstDTMFLengths . . . 1305
8.1352 VoiceCallInfoReq Struct Reference . . . 1305
8.1352.1 Detailed Description . . . 1305
8.1352.2 Field Documentation . . . 1305
8.1352.2.1 callID . . . 1305
8.1353 VoiceCallInfoResp Struct Reference . . . 1305
8.1353.1 Detailed Description . . . 1306
8.1353.2 Field Documentation . . . 1308
8.1353.2.1 pAlertingPattern . . . 1308
8.1353.2.2 pAlertType . . . 1308
8.1353.2.3 pAlphaIDInfo . . . 1308
8.1353.2.4 pCallInfo . . . 1308
8.1353.2.5 pConnectNumInfo . . . 1308
8.1353.2.6 pDiagInfo . . . 1308
8.1353.2.7 pOTASPStatus . . . 1308
8.1353.2.8 pRemotePartyName . . . 1308
8.1353.2.9 pRemotePartyNum . . . 1308
8.1353.2.10 pSrvOpt . . . 1308
8.1353.2.11 pUUSInfo . . . 1308
8.1353.2.12 pVoicePrivacy . . . 1308
8.1354 VoiceCallRequestParams Struct Reference . . . 1308
8.1354.1 Detailed Description . . . 1309
8.1354.2 Field Documentation . . . 1310
8.1354.2.1 callNumber . . . 1310
8.1354.2.2 pCallPartySubAdd . . . 1310
8.1354.2.3 pCallType . . . 1310
8.1354.2.4 pCLIRType . . . 1310
8.1354.2.5 pCUGInfo . . . 1310
8.1354.2.6 pEmergencyCategory . . . 1310
8.1354.2.7 pSvcType . . . 1310
8.1354.2.8 pUUSInFo . . . 1310
8.1355 VoiceCallResponseParams Struct Reference . . . 1310
8.1355.1 Detailed Description . . . 1310
8.1355.2 Field Documentation . . . 1311
8.1355.2.1 pAlphaIDInfo . . . 1311
8.1355.2.2 pCallID . . . 1311
8.1355.2.3 pCCResultType . . . 1311
8.1355.2.4 pCCSUPSType . . . 1311

8.1356 voiceContDTMFinfo Struct Reference . . . 1311
8.1356.1 Detailed Description . . . 1311
8.1356.2 Field Documentation . . . 1311
8.1356.2.1 DTMFdigit . . . 1311
8.1356.2.2 pCallID . . . 1312
8.1357 voiceDTMFEventInfo Struct Reference . . . 1312
8.1357.1 Detailed Description . . . 1312
8.1357.2 Field Documentation . . . 1312
8.1357.2.1 DTMFInformation . . . 1312
8.1357.2.2 pOffLength . . . 1312
8.1357.2.3 pOnLength . . . 1312
8.1358 voiceFlashInfo Struct Reference . . . 1312
8.1358.1 Detailed Description . . . 1313
8.1358.2 Field Documentation . . . 1313
8.1358.2.1 pCallID . . . 1313
8.1358.2.2 pFlashPayLd . . . 1313
8.1358.2.3 pFlashType . . . 1313
8.1359 voiceGetAllCallInfo Struct Reference . . . 1313
8.1359.1 Detailed Description . . . 1314
8.1359.2 Field Documentation . . . 1315
8.1359.2.1 pArrAlertingPattern . . . 1315
8.1359.2.2 pArrAlertingType . . . 1315
8.1359.2.3 pArrAlphaID . . . 1315
8.1359.2.4 pArrCalledPartyNum . . . 1315
8.1359.2.5 pArrCallEndReason . . . 1315
8.1359.2.6 pArrCallInfo . . . 1315
8.1359.2.7 pArrConnectPartyNum . . . 1315
8.1359.2.8 pArrDiagInfo . . . 1315
8.1359.2.9 pArrRedirPartyNum . . . 1315
8.1359.2.10 pArrRemotePartyName . . . 1315
8.1359.2.11 pArrRemotePartyNum . . . 1315
8.1359.2.12 pArrSvcOption . . . 1316
8.1359.2.13 pArrUUSInfo . . . 1316
8.1359.2.14 pOTASPStatus . . . 1316
8.1359.2.15 pVoicePrivacy . . . 1316
8.1360 voiceGetCallBarringReq Struct Reference . . . 1316
8.1360.1 Detailed Description . . . 1316
8.1360.2 Field Documentation . . . 1316
8.1360.2.1 pSvcClass . . . 1316
8.1360.2.2 reason . . . 1316

8.1361 VoiceGetCallBarringResp Struct Reference . . . 1316
8.1361.1 Detailed Description . . . 1317
8.1361.2 Field Documentation . . . 1317
8.1361.2.1 pAlphaIDInfo . . . 1318
8.1361.2.2 pCallID . . . 1318
8.1361.2.3 pCCResType . . . 1318
8.1361.2.4 pCCSUPSType . . . 1318
8.1361.2.5 pFailCause . . . 1318
8.1361.2.6 pSvcClass . . . 1318
8.1362 VoiceGetCallFWReq Struct Reference . . . 1318
8.1362.1 Detailed Description . . . 1318
8.1362.2 Field Documentation . . . 1318
8.1362.2.1 pSvcClass . . . 1318
8.1362.2.2 Reason . . . 1318
8.1363 VoiceGetCallFWResp Struct Reference . . . 1318
8.1363.1 Detailed Description . . . 1319
8.1363.2 Field Documentation . . . 1320
8.1363.2.1 pAlphaIDInfo . . . 1320
8.1363.2.2 pCallID . . . 1320
8.1363.2.3 pCCResType . . . 1320
8.1363.2.4 pCCSUPSType . . . 1320
8.1363.2.5 pFailCause . . . 1320
8.1363.2.6 pGetCallFWExtInfo . . . 1320
8.1363.2.7 pGetCallFWInfo . . . 1320
8.1364 VoiceGetCallWaitInfo Struct Reference . . . 1320
8.1364.1 Detailed Description . . . 1320
8.1364.2 Field Documentation . . . 1321
8.1364.2.1 pAlphaIDInfo . . . 1321
8.1364.2.2 pCallID . . . 1321
8.1364.2.3 pCCResType . . . 1321
8.1364.2.4 pCCSUPSType . . . 1321
8.1364.2.5 pFailCause . . . 1321
8.1364.2.6 pSvcClass . . . 1321
8.1365 VoiceGetCLIPResp Struct Reference . . . 1321
8.1365.1 Detailed Description . . . 1321
8.1365.2 Field Documentation . . . 1322
8.1365.2.1 pAlphaIDInfo . . . 1322
8.1365.2.2 pCallID . . . 1322
8.1365.2.3 pCCResType . . . 1322
8.1365.2.4 pCCSUPSType . . . 1322

8.1365.2.5 pCLIPResp . . . 1322
8.1365.2.6 pFailCause . . . 1322
8.1366 VoiceGetCLIRResp Struct Reference . . . 1322
8.1366.1 Detailed Description . . . 1323
8.1366.2 Field Documentation . . . 1323
8.1366.2.1 pAlphaIDInfo . . . 1323
8.1366.2.2 pCallID . . . 1324
8.1366.2.3 pCCResType . . . 1324
8.1366.2.4 pCCSUPSType . . . 1324
8.1366.2.5 pCLIRResp . . . 1324
8.1366.2.6 pFailCause . . . 1324
8.1367 VoiceGetCNAPResp Struct Reference . . . 1324
8.1367.1 Detailed Description . . . 1324
8.1367.2 Field Documentation . . . 1325
8.1367.2.1 pAlphaIDInfo . . . 1325
8.1367.2.2 pCallID . . . 1325
8.1367.2.3 pCCResType . . . 1325
8.1367.2.4 pCCSUPSType . . . 1325
8.1367.2.5 pCNAPResp . . . 1325
8.1367.2.6 pFailCause . . . 1325
8.1368 VoiceGetCOLPResp Struct Reference . . . 1325
8.1368.1 Detailed Description . . . 1325
8.1368.2 Field Documentation . . . 1326
8.1368.2.1 pAlphaIDInfo . . . 1326
8.1368.2.2 pCallID . . . 1326
8.1368.2.3 pCCResType . . . 1326
8.1368.2.4 pCCSUPSType . . . 1326
8.1368.2.5 pCOLPResp . . . 1326
8.1368.2.6 pFailCause . . . 1326
8.1369 VoiceGetCOLRResp Struct Reference . . . 1326
8.1369.1 Detailed Description . . . 1326
8.1369.2 Field Documentation . . . 1327
8.1369.2.1 pAlphaIDInfo . . . 1327
8.1369.2.2 pCallID . . . 1327
8.1369.2.3 pCCResType . . . 1327
8.1369.2.4 pCCSUPSType . . . 1327
8.1369.2.5 pCOLRResp . . . 1327
8.1369.2.6 pFailCause . . . 1327
8.1370 VoiceGetConfigReq Struct Reference . . . 1327
8.1370.1 Detailed Description . . . 1328

8.1370.2 Field Documentation . . . 1329
8.1370.2.1 pAirTimer . . . 1329
8.1370.2.2 pAMRStatus . . . 1329
8.1370.2.3 pAutoAnswer . . . 1329
8.1370.2.4 pNamID . . . 1329
8.1370.2.5 pPrefVoicePrivacy . . . 1329
8.1370.2.6 pPrefVoiceSO . . . 1329
8.1370.2.7 pRoamTimer . . . 1329
8.1370.2.8 pTTYMode . . . 1329
8.1370.2.9 pVoiceDomainPref . . . 1329
8.1371 VoiceGetConfigResp Struct Reference . . . 1329
8.1371.1 Detailed Description . . . 1330
8.1371.2 Field Documentation . . . 1331
8.1371.2.1 pAirTimerCnt . . . 1331
8.1371.2.2 pAutoAnswerStat . . . 1331
8.1371.2.3 pCurAMRConfig . . . 1331
8.1371.2.4 pCurPrefVoiceSO . . . 1331
8.1371.2.5 pCurrTTYMode . . . 1331
8.1371.2.6 pCurVoiceDomainPref . . . 1331
8.1371.2.7 pCurVoicePrivacyPref . . . 1331
8.1371.2.8 pRoamTimerCnt . . . 1331
8.1372 VoiceIndicationRegisterInfo Struct Reference . . . 1331
8.1372.1 Detailed Description . . . 1331
8.1372.2 Field Documentation . . . 1332
8.1372.2.1 pRegDTMFEvents . . . 1332
8.1372.2.2 pRegVoicePrivacyEvents . . . 1332
8.1372.2.3 pSuppsNotifEvents . . . 1332
8.1373 VoiceInfoRec Struct Reference . . . 1332
8.1373.1 Detailed Description . . . 1332
8.1373.2 Field Documentation . . . 1334
8.1373.2.1 callID . . . 1334
8.1373.2.2 pCalledPartyInfo . . . 1334
8.1373.2.3 pCallerIDInfo . . . 1334
8.1373.2.4 pCallerNameInfo . . . 1334
8.1373.2.5 pCallingPartyInfo . . . 1334
8.1373.2.6 pCallWaitInd . . . 1334
8.1373.2.7 pCLIRCause . . . 1334
8.1373.2.8 pConnectNumInfo . . . 1334
8.1373.2.9 pDispInfo . . . 1334
8.1373.2.10 pExtDispInfo . . . 1334

8.1373.2.1 pExtDispRecInfo . . . 1334
8.1373.2.2 pLineCtrlInfo . . . 1334
8.1373.2.3 pNSSAudioCtrl . . . 1334
8.1373.2.4 pNSSRelease . . . 1334
8.1373.2.5 pRedirNumInfo . . . 1334
8.1373.2.6 pSignalInfo . . . 1334
8.1374 VoiceManageCallsReq Struct Reference . . . 1334
8.1374.1 Detailed Description . . . 1334
8.1374.2 Field Documentation . . . 1335
8.1374.2.1 pCallID . . . 1335
8.1374.2.2 SUPSType . . . 1335
8.1375 VoiceManageCallsResp Struct Reference . . . 1335
8.1375.1 Detailed Description . . . 1335
8.1375.2 Field Documentation . . . 1336
8.1375.2.1 pFailCause . . . 1336
8.1376 VoiceOrigUSSDNoWaitInfo Struct Reference . . . 1336
8.1376.1 Detailed Description . . . 1336
8.1376.2 Field Documentation . . . 1336
8.1376.2.1 USSInformation . . . 1336
8.1377 VoiceOTASPStatusInfo Struct Reference . . . 1336
8.1377.1 Detailed Description . . . 1336
8.1377.2 Field Documentation . . . 1337
8.1377.2.1 callID . . . 1337
8.1377.2.2 OTASPStatus . . . 1337
8.1378 VoicePrivacyInfo Struct Reference . . . 1337
8.1378.1 Detailed Description . . . 1337
8.1378.2 Field Documentation . . . 1338
8.1378.2.1 callID . . . 1338
8.1378.2.2 voicePrivacy . . . 1338
8.1379 VoiceSetAllCallStatusCbkInfo Struct Reference . . . 1338
8.1379.1 Detailed Description . . . 1338
8.1379.2 Field Documentation . . . 1340
8.1379.2.1 arrCallInfomation . . . 1340
8.1379.2.2 pArrAlertingPattern . . . 1340
8.1379.2.3 pArrAlertingType . . . 1340
8.1379.2.4 pArrAlphaID . . . 1340
8.1379.2.5 pArrCalledPartyNum . . . 1340
8.1379.2.6 pArrCallEndReason . . . 1340
8.1379.2.7 pArrConnectPartyNum . . . 1340
8.1379.2.8 pArrDiagInfo . . . 1340

8.1379.2.9 pArrRedirPartyNum . . . 1340
8.1379.2.10 pArrRemotePartyName . . . 1340
8.1379.2.11 pArrRemotePartyNum . . . 1340
8.1379.2.12 pArrSvcOption . . . 1340
8.1380 VoiceSetCallBarringPwdInfo Struct Reference . . . 1340
8.1380.1 Detailed Description . . . 1340
8.1380.2 Field Documentation . . . 1341
8.1380.2.1 newPasswd . . . 1341
8.1380.2.2 newPasswdAgain . . . 1341
8.1380.2.3 oldPasswd . . . 1341
8.1380.2.4 Reason . . . 1341
8.1381 VoiceSetCallBarringPwdResp Struct Reference . . . 1341
8.1381.1 Detailed Description . . . 1342
8.1381.2 Field Documentation . . . 1342
8.1381.2.1 pAlphaIDInfo . . . 1342
8.1381.2.2 pCallID . . . 1342
8.1381.2.3 pCCResType . . . 1342
8.1381.2.4 pCCSUPSType . . . 1342
8.1381.2.5 pFailCause . . . 1342
8.1382 VoiceSetConfigReq Struct Reference . . . 1342
8.1382.1 Detailed Description . . . 1343
8.1382.2 Field Documentation . . . 1343
8.1382.2.1 pAirTimerConfig . . . 1344
8.1382.2.2 pAutoAnswer . . . 1344
8.1382.2.3 pPrefVoiceDomain . . . 1344
8.1382.2.4 pPrefVoiceSO . . . 1344
8.1382.2.5 pRoamTimerConfig . . . 1344
8.1382.2.6 pTTYMode . . . 1344
8.1383 VoiceSetConfigResp Struct Reference . . . 1344
8.1383.1 Detailed Description . . . 1344
8.1383.2 Field Documentation . . . 1345
8.1383.2.1 pAirTimerStatus . . . 1345
8.1383.2.2 pAutoAnsStatus . . . 1345
8.1383.2.3 pPrefVoiceSOStatus . . . 1345
8.1383.2.4 pRoamTimerStatus . . . 1345
8.1383.2.5 pTTYConfigStatus . . . 1345
8.1383.2.6 pVoiceDomainPrefStatus . . . 1345
8.1384 VoiceSetPrefPrivacy Struct Reference . . . 1345
8.1384.1 Detailed Description . . . 1345
8.1384.2 Field Documentation . . . 1346

8.1384.2.1 privacyPref . . . 1346
8.1385 voiceSetSUPSServiceReq Struct Reference . . . 1346
8.1385.1 Detailed Description . . . 1346
8.1385.2 Field Documentation . . . 1348
8.1385.2.1 pCallBarringPasswd . . . 1348
8.1385.2.2 pCallForwardingNumber . . . 1348
8.1385.2.3 pCallFwdTypeAndPlan . . . 1348
8.1385.2.4 pServiceClass . . . 1348
8.1385.2.5 pTimerVal . . . 1348
8.1385.2.6 reason . . . 1348
8.1385.2.7 voiceSvc . . . 1348
8.1386 voiceSetSUPSServiceResp Struct Reference . . . 1348
8.1386.1 Detailed Description . . . 1348
8.1386.2 Field Documentation . . . 1349
8.1386.2.1 pAlphaIDInfo . . . 1349
8.1386.2.2 pCallID . . . 1349
8.1386.2.3 pCCResultType . . . 1349
8.1386.2.4 pCCSUPSType . . . 1349
8.1386.2.5 pFailCause . . . 1349
8.1387 voiceStopContDTMFinfo Struct Reference . . . 1349
8.1387.1 Detailed Description . . . 1349
8.1387.2 Field Documentation . . . 1350
8.1387.2.1 callID . . . 1350
8.1388 voiceSUPSInfo Struct Reference . . . 1350
8.1388.1 Detailed Description . . . 1350
8.1388.2 Field Documentation . . . 1352
8.1388.2.1 pAlphaIDInfo . . . 1352
8.1388.2.2 pCallBarPasswd . . . 1352
8.1388.2.3 pCallFwdInfo . . . 1352
8.1388.2.4 pCallFWNum . . . 1352
8.1388.2.5 pCallFWTimerVal . . . 1352
8.1388.2.6 pCallID . . . 1352
8.1388.2.7 pCLIPstatus . . . 1352
8.1388.2.8 pCLIRstatus . . . 1352
8.1388.2.9 pCNAPstatus . . . 1352
8.1388.2.10 pCOLPstatus . . . 1352
8.1388.2.11 pCOLRstatus . . . 1352
8.1388.2.12 pDataSrc . . . 1352
8.1388.2.13 pFailCause . . . 1352
8.1388.2.14 pNewPwdData . . . 1352

8.1388.2.15 pReason . . . 1352
8.1388.2.16 pSvcClass . . . 1352
8.1388.2.17 pUSSInfo . . . 1352
8.1388.2.18 SUPSInformation . . . 1352
8.1389 VoiceSUPSNotification Struct Reference . . . 1352
8.1389.1 Detailed Description . . . 1353
8.1389.2 Field Documentation . . . 1354
8.1389.2.1 callID . . . 1354
8.1389.2.2 notifType . . . 1354
8.1389.2.3 pCUGIndex . . . 1354
8.1389.2.4 pECTNum . . . 1354
8.1390 WcdmaCellInfo Struct Reference . . . 1354
8.1390.1 Detailed Description . . . 1354
8.1390.2 Field Documentation . . . 1354
8.1390.2.1 cpich_ecno . . . 1354
8.1390.2.2 cpich_rscp . . . 1354
8.1390.2.3 psc . . . 1355
8.1390.2.4 srxlev . . . 1355
8.1391 WCDMAECIOThresh Struct Reference . . . 1355
8.1391.1 Detailed Description . . . 1355
8.1391.2 Field Documentation . . . 1355
8.1391.2.1 pWCDMAECIOThreshList . . . 1355
8.1391.2.2 WCDMAECIOThreshListLen . . . 1355
8.1392 WCDMAInfoLTENeighborCell Struct Reference . . . 1355
8.1392.1 Detailed Description . . . 1355
8.1392.2 Field Documentation . . . 1356
8.1392.2.1 UMTSLTENbrCell . . . 1356
8.1392.2.2 umtsLTENbrCellLen . . . 1356
8.1392.2.3 wcdmaRRCState . . . 1356
8.1393 WcdmaLongMsgDecodingParams Struct Reference . . . 1356
8.1393.1 Detailed Description . . . 1356
8.1393.2 Field Documentation . . . 1357
8.1393.2.1 Date . . . 1357
8.1393.2.2 pIsUDHPresent . . . 1357
8.1393.2.3 pMessage . . . 1358
8.1393.2.4 pPartNum . . . 1358
8.1393.2.5 pReferenceNum . . . 1358
8.1393.2.6 pScAddr . . . 1358
8.1393.2.7 pScAddrLength . . . 1358
8.1393.2.8 pSenderAddr . . . 1358

8.1393.2.9 pSenderAddrLength . . . 1358
8.1393.2.10 pTextMsg . . . 1358
8.1393.2.11 pTextMsgLength . . . 1358
8.1393.2.12 pTotalNum . . . 1358
8.1393.2.13 Time . . . 1358
8.1394 WcdmaMsgDecodingParams Struct Reference . . . 1358
8.1394.1 Detailed Description . . . 1358
8.1394.2 Field Documentation . . . 1359
8.1394.2.1 Date . . . 1359
8.1394.2.2 pMessage . . . 1359
8.1394.2.3 pScAddr . . . 1359
8.1394.2.4 pScAddrLength . . . 1359
8.1394.2.5 pSenderAddr . . . 1359
8.1394.2.6 pSenderAddrLength . . . 1359
8.1394.2.7 pTextMsg . . . 1359
8.1394.2.8 pTextMsgLength . . . 1359
8.1394.2.9 Time . . . 1359
8.1395 WcdmaMsgEncodingParams Struct Reference . . . 1359
8.1395.1 Detailed Description . . . 1360
8.1395.2 Field Documentation . . . 1360
8.1395.2.1 alphabet . . . 1360
8.1395.2.2 messageSize . . . 1360
8.1395.2.3 pDestAddr . . . 1360
8.1395.2.4 pPDUMessage . . . 1360
8.1395.2.5 pTextMsg . . . 1360
8.1396 WCDMARSSIThresh Struct Reference . . . 1360
8.1396.1 Detailed Description . . . 1360
8.1396.2 Field Documentation . . . 1361
8.1396.2.1 pWCDMARSSIThreshList . . . 1361
8.1396.2.2 WCDMARSSIThreshListLen . . . 1361
8.1397 WCDMASysInfo Struct Reference . . . 1361
8.1397.1 Detailed Description . . . 1361
8.1397.2 Field Documentation . . . 1363
8.1397.2.1 cellId . . . 1364
8.1397.2.2 cellIdValid . . . 1364
8.1397.2.3 hsCallStatus . . . 1364
8.1397.2.4 hsCallStatusValid . . . 1364
8.1397.2.5 hsInd . . . 1364
8.1397.2.6 hsIndValid . . . 1364
8.1397.2.7 lac . . . 1364

8.1397.2.8 lacValid . . . 1364
8.1397.2.9 MCC . . . 1364
8.1397.2.10 MNC . . . 1364
8.1397.2.11 networkIdValid . . . 1364
8.1397.2.12 psc . . . 1364
8.1397.2.13 pscValid . . . 1364
8.1397.2.14 regRejectInfoValid . . . 1364
8.1397.2.15 rejCause . . . 1364
8.1397.2.16 rejectSrvDomain . . . 1364
8.1397.2.17 sysInfoWCDMA . . . 1364
8.1398 wcdmaUARFCN Struct Reference . . . 1364
8.1398.1 Detailed Description . . . 1364
8.1398.2 Field Documentation . . . 1365
8.1398.2.1 status . . . 1365
8.1398.2.2 uarfcn . . . 1365
8.1399 wds_currNetworkInfo Struct Reference . . . 1365
8.1399.1 Detailed Description . . . 1365
8.1399.2 Field Documentation . . . 1366
8.1399.2.1 NetworkType . . . 1366
8.1399.2.2 RATMask . . . 1366
8.1399.2.3 SOMask . . . 1366
8.1400 wds_DataULongLongTlv Struct Reference . . . 1366
8.1400.1 Detailed Description . . . 1366
8.1400.2 Field Documentation . . . 1366
8.1400.2.1 TlvPresent . . . 1366
8.1400.2.2 ullData . . . 1367
8.1401 wds_DataULongTlv Struct Reference . . . 1367
8.1401.1 Detailed Description . . . 1367
8.1401.2 Field Documentation . . . 1367
8.1401.2.1 TlvPresent . . . 1367
8.1401.2.2 ulData . . . 1367
8.1402 wds_DHCPLeaseOptTlv Struct Reference . . . 1367
8.1402.1 Detailed Description . . . 1367
8.1402.2 Field Documentation . . . 1367
8.1402.2.1 numOpt . . . 1367
8.1402.2.2 optList . . . 1367
8.1402.2.3 optListData . . . 1367
8.1402.2.4 TlvPresent . . . 1367
8.1403 wds_DHCPLeaseStateTlv Struct Reference . . . 1368
8.1403.1 Detailed Description . . . 1368

8.1403.2 Field Documentation . . . 1368
8.1403.2.1 leaseState . . . 1368
8.1403.2.2 TlvPresent . . . 1368
8.1404 wds_DHCPOpt Struct Reference . . . 1368
8.1404.1 Detailed Description . . . 1368
8.1404.2 Field Documentation . . . 1369
8.1404.2.1 optCode . . . 1369
8.1404.2.2 optValLen . . . 1369
8.1404.2.3 pOptVal . . . 1369
8.1405 wds_DHCPProfileIdTlv Struct Reference . . . 1369
8.1405.1 Detailed Description . . . 1369
8.1405.2 Field Documentation . . . 1369
8.1405.2.1 profileId . . . 1369
8.1405.2.2 profileType . . . 1369
8.1405.2.3 TlvPresent . . . 1369
8.1406 wds_DHCPv4HWConfig Struct Reference . . . 1369
8.1406.1 Detailed Description . . . 1369
8.1406.2 Field Documentation . . . 1370
8.1406.2.1 chaddr . . . 1370
8.1406.2.2 chaddrLen . . . 1370
8.1406.2.3 hwType . . . 1370
8.1407 wds_DHCPv4Option Struct Reference . . . 1370
8.1407.1 Detailed Description . . . 1370
8.1407.2 Field Documentation . . . 1370
8.1407.2.1 optCode . . . 1370
8.1407.2.2 optVal . . . 1370
8.1407.2.3 optValLen . . . 1370
8.1408 wds_DHCPv4OptionList Struct Reference . . . 1371
8.1408.1 Detailed Description . . . 1371
8.1408.2 Field Documentation . . . 1371
8.1408.2.1 numOpt . . . 1371
8.1408.2.2 pOptList . . . 1371
8.1409 wds_DHCPv4ProfileId Struct Reference . . . 1371
8.1409.1 Detailed Description . . . 1371
8.1409.2 Field Documentation . . . 1371
8.1409.2.1 profileId . . . 1371
8.1409.2.2 profileType . . . 1371
8.1410 wds_Domain Struct Reference . . . 1372
8.1410.1 Detailed Description . . . 1372
8.1410.2 Field Documentation . . . 1372

8.1410.2.1 domainLen . . . 1372
8.1410.2.2 domainName . . . 1372
8.1411 wds_DomainNameList Struct Reference . . . 1372
8.1411.1 Detailed Description . . . 1372
8.1411.2 Field Documentation . . . 1372
8.1411.2.1 domain . . . 1372
8.1411.2.2 numInstances . . . 1372
8.1412 wds_GPRSQoS Struct Reference . . . 1373
8.1412.1 Detailed Description . . . 1373
8.1412.2 Field Documentation . . . 1373
8.1412.2.1 delayClass . . . 1373
8.1412.2.2 meanThroughputClass . . . 1373
8.1412.2.3 peakThroughputClass . . . 1373
8.1412.2.4 precedenceClass . . . 1373
8.1412.2.5 reliabilityClass . . . 1373
8.1413 wds_IPv4AdTlv Struct Reference . . . 1373
8.1413.1 Detailed Description . . . 1374
8.1413.2 Field Documentation . . . 1374
8.1413.2.1 IPv4Addr . . . 1374
8.1413.2.2 TlvPresent . . . 1374
8.1414 wds_IPV6AddressInfo Struct Reference . . . 1374
8.1414.1 Detailed Description . . . 1374
8.1414.2 Field Documentation . . . 1374
8.1414.2.1 IPAddressV6 . . . 1374
8.1414.2.2 IPV6PrefixLen . . . 1374
8.1415 wds_IPV6GWAddressInfo Struct Reference . . . 1374
8.1415.1 Detailed Description . . . 1374
8.1415.2 Field Documentation . . . 1375
8.1415.2.1 gwAddressV6 . . . 1375
8.1415.2.2 gwV6PrefixLen . . . 1375
8.1416 wds_PCSCFFQDNAddress Struct Reference . . . 1375
8.1416.1 Detailed Description . . . 1375
8.1416.2 Field Documentation . . . 1375
8.1416.2.1 fqdnAddr . . . 1375
8.1416.2.2 fqdnLen . . . 1375
8.1417 wds_PCSCFFQDNAddressList Struct Reference . . . 1375
8.1417.1 Detailed Description . . . 1375
8.1417.2 Field Documentation . . . 1376
8.1417.2.1 numInstances . . . 1376
8.1417.2.2 pcsfFQDNAddress . . . 1376

8.1418 wds_PCSCFIPv4ServerAddressList Struct Reference . . . 1376
8.1418.1 Detailed Description . . . 1376
8.1418.2 Field Documentation . . . 1376
8.1418.2.1 numInstances . . . 1376
8.1418.2.2 pscsfIPv4Addr . . . 1376
8.1419 wds_ProfileIdentifier Struct Reference . . . 1376
8.1419.1 Detailed Description . . . 1376
8.1419.2 Field Documentation . . . 1377
8.1419.2.1 profileIndex . . . 1377
8.1419.2.2 profileType . . . 1377
8.1420 wds_profileInfo Union Reference . . . 1377
8.1420.1 Detailed Description . . . 1377
8.1420.2 Field Documentation . . . 1377
8.1420.2.1 SlqsProfile3GPP . . . 1377
8.1420.2.2 SlqsProfile3GPP2 . . . 1377
8.1421 wds_TrStatInd Struct Reference . . . 1377
8.1421.1 Detailed Description . . . 1377
8.1421.2 Field Documentation . . . 1378
8.1421.2.1 statsMask . . . 1378
8.1421.2.2 statsPeriod . . . 1378
8.1422 wds_UMTSMinQoS Struct Reference . . . 1378
8.1422.1 Detailed Description . . . 1378
8.1422.2 Field Documentation . . . 1380
8.1422.2.1 deliveryErrSDU . . . 1380
8.1422.2.2 grntDownlinkBitrate . . . 1380
8.1422.2.3 grntUplinkBitrate . . . 1380
8.1422.2.4 maxDownlinkBitrate . . . 1380
8.1422.2.5 maxSDUSize . . . 1380
8.1422.2.6 maxUplinkBitrate . . . 1380
8.1422.2.7 qosDeliveryOrder . . . 1380
8.1422.2.8 resBerRatio . . . 1380
8.1422.2.9 sduErrorRatio . . . 1380
8.1422.2.10 trafficClass . . . 1380
8.1422.2.11 trafficPriority . . . 1380
8.1422.2.12 transferDelay . . . 1380
8.1423 WdsByteTotals Struct Reference . . . 1380
8.1423.1 Detailed Description . . . 1381
8.1423.2 Field Documentation . . . 1381
8.1423.2.1 ByteTotalsElmntsV4 . . . 1381
8.1423.2.2 ByteTotalsElmntsV6 . . . 1381

8.1423.2.3 pV4sessionId . . . 1381
8.1423.2.4 pV6sessionId . . . 1381
8.1424 WdsByteTotalsElmnts Struct Reference . . . 1381
8.1424.1 Detailed Description . . . 1381
8.1424.2 Field Documentation . . . 1382
8.1424.2.1 pRXTotalBytes . . . 1382
8.1424.2.2 pTXTotalBytes . . . 1382
8.1425 WdsClientLeaseChange Struct Reference . . . 1382
8.1425.1 Detailed Description . . . 1382
8.1425.2 Field Documentation . . . 1382
8.1425.2.1 pEnableNotification . . . 1382
8.1426 WdsConnectionRate Struct Reference . . . 1382
8.1426.1 Detailed Description . . . 1382
8.1426.2 Field Documentation . . . 1383
8.1426.2.1 ConnRateElmntsV4 . . . 1383
8.1426.2.2 ConnRateElmntsV6 . . . 1383
8.1426.2.3 pV4sessionId . . . 1383
8.1426.2.4 pV6sessionId . . . 1383
8.1427 WdsConnectionRateElmnts Struct Reference . . . 1383
8.1427.1 Detailed Description . . . 1383
8.1427.2 Field Documentation . . . 1384
8.1427.2.1 pCurrentChannelRXRate . . . 1384
8.1427.2.2 pCurrentChannelTXRate . . . 1384
8.1427.2.3 pMaxChannelRXRate . . . 1384
8.1427.2.4 pMaxChannelTXRate . . . 1384
8.1428 WdsDHCPv4ClientLeaseInd Struct Reference . . . 1384
8.1428.1 Detailed Description . . . 1384
8.1428.2 Field Documentation . . . 1384
8.1428.2.1 pIPv4Addr . . . 1384
8.1428.2.2 pLeaseState . . . 1384
8.1428.2.3 pOptList . . . 1385
8.1428.2.4 pProfileId . . . 1385
8.1429 WdsDHCPv4Config Struct Reference . . . 1385
8.1429.1 Detailed Description . . . 1385
8.1429.2 Field Documentation . . . 1385
8.1429.2.1 pHwConfig . . . 1385
8.1429.2.2 pProfileId . . . 1385
8.1429.2.3 pRequestOptionList . . . 1385
8.1430 WdsDhcpv4HwConfig Struct Reference . . . 1385
8.1430.1 Detailed Description . . . 1385

8.1430.2 Field Documentation . . . 1386
8.1430.2.1 chaddr . . . 1386
8.1430.2.2 chaddrLen . . . 1386
8.1430.2.3 hwType . . . 1386
8.1431 WdsDHCPv4HWConfig Struct Reference . . . 1386
8.1431.1 Detailed Description . . . 1386
8.1431.2 Field Documentation . . . 1386
8.1431.2.1 chaddr . . . 1386
8.1431.2.2 chaddrLen . . . 1386
8.1431.2.3 hwType . . . 1386
8.1432 WdsDHCPv4Option Struct Reference . . . 1386
8.1432.1 Detailed Description . . . 1387
8.1432.2 Field Documentation . . . 1387
8.1432.2.1 optCode . . . 1387
8.1432.2.2 optVal . . . 1387
8.1432.2.3 optValLen . . . 1387
8.1433 WdsDhcpv4Option Struct Reference . . . 1387
8.1433.1 Detailed Description . . . 1387
8.1433.2 Field Documentation . . . 1387
8.1433.2.1 optCode . . . 1387
8.1433.2.2 optVal . . . 1387
8.1433.2.3 optValLen . . . 1387
8.1434 WdsDhcpv4OptionList Struct Reference . . . 1388
8.1434.1 Detailed Description . . . 1388
8.1434.2 Field Documentation . . . 1388
8.1434.2.1 numOpt . . . 1388
8.1434.2.2 pOptList . . . 1388
8.1435 WdsDHCPv4OptionList Struct Reference . . . 1388
8.1435.1 Detailed Description . . . 1388
8.1435.2 Field Documentation . . . 1388
8.1435.2.1 numOpt . . . 1388
8.1435.2.2 pOptList . . . 1388
8.1436 WdsDhcpv4ProfileId Struct Reference . . . 1388
8.1436.1 Detailed Description . . . 1389
8.1436.2 Field Documentation . . . 1389
8.1436.2.1 profileId . . . 1389
8.1436.2.2 profileType . . . 1389
8.1437 WdsDHCPv4ProfileId Struct Reference . . . 1389
8.1437.1 Detailed Description . . . 1389
8.1437.2 Field Documentation . . . 1389

8.1437.2.1 profileId . . . 1389
8.1437.2.2 profileType . . . 1389
8.1438 WDSGetLoopbackData Struct Reference . . . 1389
8.1438.1 Detailed Description . . . 1390
8.1438.2 Field Documentation . . . 1390
8.1438.2.1 ByteLoopbackMode . . . 1390
8.1438.2.2 ByteLoopbackMultiplier . . . 1390
8.1439 WdsIpAddressInfoReq Struct Reference . . . 1390
8.1439.1 Detailed Description . . . 1390
8.1439.2 Field Documentation . . . 1391
8.1439.2.1 ip . . . 1391
8.1439.2.2 pv4sessionId . . . 1391
8.1439.2.3 pv6sessionId . . . 1391
8.1440 WdsPktStatisticsElmnts Struct Reference . . . 1391
8.1440.1 Detailed Description . . . 1391
8.1440.2 Field Documentation . . . 1392
8.1440.2.1 pRXDroppedCount . . . 1392
8.1440.2.2 pRXOkBytesCount . . . 1392
8.1440.2.3 pRXOKBytesLastCall . . . 1392
8.1440.2.4 pRXPacketErrors . . . 1392
8.1440.2.5 pRXPacketOverflows . . . 1392
8.1440.2.6 pRXPacketSuccesses . . . 1392
8.1440.2.7 pTXDroppedCount . . . 1392
8.1440.2.8 pTXOkBytesCount . . . 1392
8.1440.2.9 pTXOKBytesLastCall . . . 1392
8.1440.2.10 pTXPacketErrors . . . 1392
8.1440.2.11 pTXPacketOverflows . . . 1392
8.1440.2.12 pTXPacketSuccesses . . . 1393
8.1441 WdsPktStatisticsReq Struct Reference . . . 1393
8.1441.1 Detailed Description . . . 1393
8.1441.2 Field Documentation . . . 1393
8.1441.2.1 pStatMask . . . 1393
8.1442 WdsPktStatisticsResp Struct Reference . . . 1393
8.1442.1 Detailed Description . . . 1393
8.1442.2 Field Documentation . . . 1394
8.1442.2.1 PktStatElmntsV4 . . . 1394
8.1442.2.2 PktStatElmntsV6 . . . 1394
8.1442.2.3 pV4sessionId . . . 1394
8.1442.2.4 pV6sessionId . . . 1394
8.1443 WdsProfileParam Union Reference . . . 1394

8.1443.1 Detailed Description . . . 1394
8.1443.2 Field Documentation . . . 1394
8.1443.2.1 SlqsProfile3GPP . . . 1394
8.1443.2.2 SlqsProfile3GPP2 . . . 1394
8.1444 WdsRunTimeSettings Struct Reference . . . 1394
8.1444.1 Detailed Description . . . 1394
8.1444.2 Field Documentation . . . 1395
8.1444.2.1 rts . . . 1395
8.1444.2.2 v4sessionId . . . 1395
8.1444.2.3 v6sessionId . . . 1395
8.1445 WdsSetEventReportReq Struct Reference . . . 1395
8.1445.1 Detailed Description . . . 1395
8.1445.2 Field Documentation . . . 1396
8.1445.2.1 pCurrChannelRateInd . . . 1396
8.1445.2.2 pCurrDataBearerTechInd . . . 1396
8.1445.2.3 pCurrPrefDataSysInd . . . 1396
8.1445.2.4 pDataBearerTechInd . . . 1397
8.1445.2.5 pDataCallStatusChangeInd . . . 1397
8.1445.2.6 pDataSystemStatusChangeInd . . . 1397
8.1445.2.7 pDormancyStatusInd . . . 1397
8.1445.2.8 pEVDOPageMonPerChangeInd . . . 1397
8.1445.2.9 pMIPStatusInd . . . 1397
8.1445.2.10 pTransferStatInd . . . 1397
8.1446 WDSSetLoopbackData Struct Reference . . . 1397
8.1446.1 Detailed Description . . . 1397
8.1446.2 Field Documentation . . . 1397
8.1446.2.1 pLoopbackMode . . . 1397
8.1446.2.2 pLoopbackMultiplier . . . 1397
8.1447 WDSSWICurrentChannelRates Struct Reference . . . 1397
8.1447.1 Detailed Description . . . 1398
8.1447.2 Field Documentation . . . 1398
8.1447.2.1 current_channel_rx_rate . . . 1398
8.1447.2.2 current_channel_tx_rate . . . 1398
8.1447.2.3 max_channel_rx_rate . . . 1398
8.1447.2.4 max_channel_tx_rate . . . 1398

9 File Documentation **1399**

9.1 apdoxypages.c File Reference . . . 1399
9.1.1 Detailed Description . . . 1399
9.2 common.h File Reference . . . 1399

9.2.1 Macro Definition Documentation . . . 1401
9.2.1.1 DEAULT_LOC_TIMEOUT_IN_SEC . . . 1401
9.2.1.2 MINREQBKLEN . . . 1401
9.2.1.3 MSGID_AND_LEN . . . 1401
9.2.1.4 MSGID_DONT_CARE . . . 1401
9.2.1.5 SDK_VALIDATE_INPUT_PACK_PARAM . . . 1401
9.2.1.6 SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID . . . 1401
9.2.1.7 SDU_HDR_LEN . . . 1401
9.2.1.8 UNUSEDPARAM . . . 1401
9.2.2 Typedef Documentation . . . 1401
9.2.2.1 logger . . . 1401
9.2.3 Enumeration Type Documentation . . . 1401
9.2.3.1 eLOG_LEVEL . . . 1401
9.2.3.2 eQMI_SVC . . . 1402
9.2.3.3 eTimeout . . . 1402
9.2.3.4 msgtype . . . 1403
9.2.4 Function Documentation . . . 1403
9.2.4.1 fill_pack_ctx . . . 1403
9.2.4.2 fill_sdu_hdr . . . 1403
9.2.4.3 get_version . . . 1403
9.2.4.4 helper_get_resp_ctx . . . 1403
9.2.4.5 helper_get_xid . . . 1403
9.2.4.6 helper_set_log_func . . . 1403
9.2.4.7 helper_set_log_lvl . . . 1403
9.2.4.8 liteqmi_GetVersion . . . 1403
9.2.4.9 liteqmi_helper_decode7bitAsciiEncString . . . 1404
9.2.4.10 liteqmi_log . . . 1404
9.2.4.11 unpack_result_code_only . . . 1404
9.2.5 Variable Documentation . . . 1404
9.2.5.1 glog . . . 1404
9.2.5.2 gloglvl . . . 1404
9.3 dms.h File Reference . . . 1404
9.3.1 Macro Definition Documentation . . . 1410
9.3.1.1 ACT_CODE_MAX_SIZE . . . 1410
9.3.1.2 CK_MAX_SIZE . . . 1410
9.3.1.3 DMS_IMGDETAILS_LEN . . . 1410
9.3.1.4 DMS_MAX_CUST_ID_LEN . . . 1410
9.3.1.5 DMS_MAX_CUST_VALUE_LEN . . . 1410
9.3.1.6 DMS_MAX_FWUPDATE_LOG_STR_SZ . . . 1410
9.3.1.7 DMS_MAX_FWUPDATE_REF_STR_SZ . . . 1410

9.3.1.8 DMS_PM_FACTORY . . . 1411
9.3.1.9 DMS_PM_LOW . . . 1411
9.3.1.10 DMS_PM_OFFLINE . . . 1411
9.3.1.11 DMS_PM_ONLINE . . . 1411
9.3.1.12 DMS_PM_PERSISTENT_LOW . . . 1411
9.3.1.13 DMS_PM_RESET . . . 1411
9.3.1.14 DMS_PM_SHUT_DOWN . . . 1411
9.3.1.15 DMS_SET_REPORT_DISABLE . . . 1411
9.3.1.16 DMS_SET_REPORT_ENABLE . . . 1411
9.3.1.17 DMS_SLQSFWINFO_APPVERSION_SZ . . . 1411
9.3.1.18 DMS_SLQSFWINFO_BOOTVERSION_SZ . . . 1411
9.3.1.19 DMS_SLQSFWINFO_CARRIER_SZ . . . 1411
9.3.1.20 DMS_SLQSFWINFO_CUR_CARR_NAME . . . 1411
9.3.1.21 DMS_SLQSFWINFO_CUR_CARR_REV . . . 1411
9.3.1.22 DMS_SLQSFWINFO_MODELID_SZ . . . 1411
9.3.1.23 DMS_SLQSFWINFO_PACKAGEID_SZ . . . 1411
9.3.1.24 DMS_SLQSFWINFO_PRIVERSION_SZ . . . 1411
9.3.1.25 DMS_SLQSFWINFO_SKU_SZ . . . 1411
9.3.1.26 DMS_SWI_SET_IND_DISABLE . . . 1411
9.3.1.27 DMS_SWI_SET_IND_ENABLE . . . 1411
9.3.1.28 DMS_UINT8_MAX_STRING_SZ . . . 1411
9.3.1.29 ERI_DATA_MAX_SIZE . . . 1411
9.3.1.30 MAX_BUILD_ID_LEN . . . 1411
9.3.1.31 MEID_MAX_SIZE . . . 1411
9.3.1.32 SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1411
9.3.1.33 SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1411
9.3.1.34 SPC_SIZE . . . 1411
9.3.1.35 UNIQUE_ID_LEN . . . 1411
9.3.2 Function Documentation . . . 1412
9.3.2.1 pack_dms_ActivateAutomatic . . . 1412
9.3.2.2 pack_dms_GetActivationState . . . 1412
9.3.2.3 pack_dms_GetBandCapability . . . 1412
9.3.2.4 pack_dms_GetCrashAction . . . 1413
9.3.2.5 pack_dms_GetCustFeature . . . 1413
9.3.2.6 pack_dms_GetCustFeaturesV2 . . . 1413
9.3.2.7 pack_dms_GetDeviceCap . . . 1414
9.3.2.8 pack_dms_GetDeviceCapabilities . . . 1414
9.3.2.9 pack_dms_GetDeviceHardwareRev . . . 1415
9.3.2.10 pack_dms_GetDeviceMfr . . . 1415
9.3.2.11 pack_dms_GetDeviceSerialNumbers . . . 1415

9.3.2.12 pack_dms_GetFirmwareInfo . . . 1416
9.3.2.13 pack_dms_GetFirmwareRevision . . . 1416
9.3.2.14 pack_dms_GetFirmwareRevisions . . . 1416
9.3.2.15 pack_dms_GetFSN . . . 1417
9.3.2.16 pack_dms_GetHardwareRevision . . . 1417
9.3.2.17 pack_dms_GetIMSI . . . 1417
9.3.2.18 pack_dms_GetManufacturer . . . 1418
9.3.2.19 pack_dms_GetModelID . . . 1418
9.3.2.20 pack_dms_GetNetworkTime . . . 1418
9.3.2.21 pack_dms_GetOfflineReason . . . 1419
9.3.2.22 pack_dms_GetPower . . . 1419
9.3.2.23 pack_dms_GetPRLVersion . . . 1419
9.3.2.24 pack_dms_GetSerialNumbers . . . 1420
9.3.2.25 pack_dms_GetUSBComp . . . 1420
9.3.2.26 pack_dms_GetVoiceNumber . . . 1421
9.3.2.27 pack_dms_ResetToFactoryDefaults . . . 1421
9.3.2.28 pack_dms_SetActivationStatusCallback . . . 1421
9.3.2.29 pack_dms_SetCrashAction . . . 1422
9.3.2.30 pack_dms_SetCustFeature . . . 1422
9.3.2.31 pack_dms_SetCustFeaturesV2 . . . 1422
9.3.2.32 pack_dms_SetEventReport . . . 1423
9.3.2.33 pack_dms_SetFirmwarePreference . . . 1423
9.3.2.34 pack_dms_SetPower . . . 1423
9.3.2.35 pack_dms_SetUSBComp . . . 1424
9.3.2.36 pack_dms_SLQSDmsSwiGetResetInfo . . . 1424
9.3.2.37 pack_dms_SLQSDmsSwiIndicationRegister . . . 1425
9.3.2.38 pack_dms_SLQSGetBandCapability . . . 1425
9.3.2.39 pack_dms_SLQSGetERIFile . . . 1425
9.3.2.40 pack_dms_SLQSSwiClearDyingGaspStatistics . . . 1426
9.3.2.41 pack_dms_SLQSSwiGetCrashInfo . . . 1426
9.3.2.42 pack_dms_SLQSSwiGetDyingGaspCfg . . . 1426
9.3.2.43 pack_dms_SLQSSwiGetDyingGaspStatistics . . . 1427
9.3.2.44 pack_dms_SLQSSwiGetFirmwareCurr . . . 1427
9.3.2.45 pack_dms_SLQSSwiGetFwUpdateStatus . . . 1427
9.3.2.46 pack_dms_SLQSSwiGetHostDevInfo . . . 1428
9.3.2.47 pack_dms_SLQSSwiGetOSInfo . . . 1428
9.3.2.48 pack_dms_SLQSSwiGetSerialNoExt . . . 1428
9.3.2.49 pack_dms_SLQSSwiSetDyingGaspCfg . . . 1429
9.3.2.50 pack_dms_SLQSSwiSetHostDevInfo . . . 1429
9.3.2.51 pack_dms_SLQSSwiSetOSInfo . . . 1429

9.3.2.52 pack_dms_SLQSUIMGetState . . . 1430
9.3.2.53 pack_dms_UIMChangePIN . . . 1430
9.3.2.54 pack_dms_UIMGetControlKeyStatus . . . 1431
9.3.2.55 pack_dms_UIMGetICCID . . . 1431
9.3.2.56 pack_dms_UIMGetPINStatus . . . 1431
9.3.2.57 pack_dms_UIMSetControlKeyProtection . . . 1432
9.3.2.58 pack_dms_UIMSetPINProtection . . . 1432
9.3.2.59 pack_dms_UIMUnblockControlKey . . . 1432
9.3.2.60 pack_dms_UIMUnblockPIN . . . 1433
9.3.2.61 pack_dms_UIMVerifyPIN . . . 1433
9.3.2.62 pack_dms_ValidateSPC . . . 1434
9.3.2.63 unpack_dms_ActivateAutomatic . . . 1434
9.3.2.64 unpack_dms_GetActivationState . . . 1434
9.3.2.65 unpack_dms_GetBandCapability . . . 1435
9.3.2.66 unpack_dms_GetCrashAction . . . 1435
9.3.2.67 unpack_dms_GetCustFeature . . . 1435
9.3.2.68 unpack_dms_GetCustFeaturesV2 . . . 1436
9.3.2.69 unpack_dms_GetDeviceCap . . . 1436
9.3.2.70 unpack_dms_GetDeviceCapabilities . . . 1436
9.3.2.71 unpack_dms_GetDeviceHardwareRev . . . 1437
9.3.2.72 unpack_dms_GetDeviceMfr . . . 1437
9.3.2.73 unpack_dms_GetDeviceSerialNumbers . . . 1437
9.3.2.74 unpack_dms_GetFirmwareInfo . . . 1438
9.3.2.75 unpack_dms_GetFirmwareRevision . . . 1438
9.3.2.76 unpack_dms_GetFirmwareRevisions . . . 1438
9.3.2.77 unpack_dms_GetFSN . . . 1439
9.3.2.78 unpack_dms_GetHardwareRevision . . . 1439
9.3.2.79 unpack_dms_GetIMSI . . . 1439
9.3.2.80 unpack_dms_GetManufacturer . . . 1440
9.3.2.81 unpack_dms_GetModelID . . . 1440
9.3.2.82 unpack_dms_GetNetworkTime . . . 1440
9.3.2.83 unpack_dms_GetOfflineReason . . . 1441
9.3.2.84 unpack_dms_GetPower . . . 1441
9.3.2.85 unpack_dms_GetPRLVersion . . . 1441
9.3.2.86 unpack_dms_GetSerialNumbers . . . 1442
9.3.2.87 unpack_dms_GetUSBComp . . . 1442
9.3.2.88 unpack_dms_GetVoiceNumber . . . 1442
9.3.2.89 unpack_dms_ResetToFactoryDefaults . . . 1443
9.3.2.90 unpack_dms_SetActivationStatusCallback . . . 1443
9.3.2.91 unpack_dms_SetCrashAction . . . 1443

9.3.2.92 unpack_dms_SetCustFeature . . . 1444
9.3.2.93 unpack_dms_SetCustFeaturesV2 . . . 1444
9.3.2.94 unpack_dms_SetEventReport . . . 1444
9.3.2.95 unpack_dms_SetEventReport_ind . . . 1445
9.3.2.96 unpack_dms_SetFirmwarePreference . . . 1445
9.3.2.97 unpack_dms_SetPower . . . 1445
9.3.2.98 unpack_dms_SetUSBComp . . . 1446
9.3.2.99 unpack_dms_SLQSDmsSwiGetResetInfo . . . 1446
9.3.2.100 unpack_dms_SLQSDmsSwiGetResetInfo_Ind . . . 1447
9.3.2.101 unpack_dms_SLQSDmsSwiIndicationRegister . . . 1447
9.3.2.102 unpack_dms_SLQSGetBandCapability . . . 1447
9.3.2.103 unpack_dms_SLQSGetERIFile . . . 1448
9.3.2.104 unpack_dms_SLQSSwiClearDyingGaspStatistics . . . 1448
9.3.2.105 unpack_dms_SLQSSwiGetCrashInfo . . . 1448
9.3.2.106 unpack_dms_SLQSSwiGetDyingGaspCfg . . . 1449
9.3.2.107 unpack_dms_SLQSSwiGetDyingGaspStatistics . . . 1449
9.3.2.108 unpack_dms_SLQSSwiGetFirmwareCurr . . . 1449
9.3.2.109 unpack_dms_SLQSSwiGetFwUpdateStatus . . . 1450
9.3.2.110 unpack_dms_SLQSSwiGetHostDevInfo . . . 1450
9.3.2.111 unpack_dms_SLQSSwiGetOSInfo . . . 1450
9.3.2.112 unpack_dms_SLQSSwiGetSerialNoExt . . . 1451
9.3.2.113 unpack_dms_SLQSSwiSetDyingGaspCfg . . . 1451
9.3.2.114 unpack_dms_SLQSSwiSetHostDevInfo . . . 1451
9.3.2.115 unpack_dms_SLQSSwiSetOSInfo . . . 1452
9.3.2.116 unpack_dms_SLQSUIMGetState . . . 1452
9.3.2.117 unpack_dms_UIMChangePIN . . . 1452
9.3.2.118 unpack_dms_UIMGetControlKeyStatus . . . 1453
9.3.2.119 unpack_dms_UIMGetICCID . . . 1453
9.3.2.120 unpack_dms_UIMGetPINStatus . . . 1453
9.3.2.121 unpack_dms_UIMSetControlKeyProtection . . . 1454
9.3.2.122 unpack_dms_UIMSetPINProtection . . . 1454
9.3.2.123 unpack_dms_UIMUnblockControlKey . . . 1455
9.3.2.124 unpack_dms_UIMUnblockPIN . . . 1455
9.3.2.125 unpack_dms_UIMVerifyPIN . . . 1455
9.3.2.126 unpack_dms_ValidateSPC . . . 1456
9.4 fms.h File Reference . . . 1456
9.4.1 Macro Definition Documentation . . . 1457
9.4.1.1 FMS_FW_PRI_BUILD_MATCH_LEN . . . 1457
9.4.1.2 FMS_GOBI_LISTENTRIES_MAX . . . 1457
9.4.1.3 FMS_GOBI_MBN_BUILD_ID_STR_LEN . . . 1457

9.4.1.4 FMS_GOBI_MBN_IMG_ID_STR_LEN . . . 1457
9.4.1.5 FMS_IMAGE_ID_BUILD_ID_LEN . . . 1457
9.4.1.6 FMS_IMAGE_ID_IMG_ID_LEN . . . 1457
9.4.1.7 FMS_IMAGE_ID_MAX_ENTRIES . . . 1457
9.4.1.8 FMS_IMAGE_ID_PRI_IMGTYPE . . . 1457
9.4.1.9 FMS_MAX_IMAGE_ID_ELEMENT . . . 1457
9.4.1.10 FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE . . . 1457
9.4.2 Function Documentation . . . 1457
9.4.2.1 GetValidFwPriCombinations . . . 1457
9.4.2.2 pack_fms_GetImagesPreference . . . 1458
9.4.2.3 pack_fms_GetStoredImages . . . 1458
9.4.2.4 pack_fms_SetImagesPreference . . . 1458
9.4.2.5 unpack_fms_GetImagesPreference . . . 1459
9.4.2.6 unpack_fms_GetStoredImages . . . 1459
9.4.2.7 unpack_fms_SetImagesPreference . . . 1459
9.5 loc.h File Reference . . . 1459
9.5.1 Macro Definition Documentation . . . 1462
9.5.1.1 LOC_UINT8_MAX_STRING_SZ . . . 1462
9.5.1.2 LOCEVENTMASKBATCHFULLNOTIFICATION . . . 1462
9.5.1.3 LOCEVENTMASKENGINESTATE . . . 1462
9.5.1.4 LOCEVENTMASKFIXSESSIONSTATE . . . 1462
9.5.1.5 LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION . . . 1462
9.5.1.6 LOCEVENTMASKGEOFENCEBREACHNOTIFICATION . . . 1462
9.5.1.7 LOCEVENTMASKGEOFENCEGENALERT . . . 1462
9.5.1.8 LOCEVENTMASKGNSSMEASUREMENTREPORT . . . 1463
9.5.1.9 LOCEVENTMASKGNSSSVINFO . . . 1463
9.5.1.10 LOCEVENTMASKINJECTPOSITIONREQ . . . 1463
9.5.1.11 LOCEVENTMASKINJECTPREDICTEDORBITSREQ . . . 1463
9.5.1.12 LOCEVENTMASKINJECTTIMEREQ . . . 1463
9.5.1.13 LOCEVENTMASKINJECTWIFIAPDATAREQ . . . 1463
9.5.1.14 LOCEVENTMASKINVALIDVALUE . . . 1463
9.5.1.15 LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT . . . 1463
9.5.1.16 LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ . . . 1463
9.5.1.17 LOCEVENTMASKMOTIONDATACONTROL . . . 1463
9.5.1.18 LOCEVENTMASKNIGEOFENCENOTIFICATION . . . 1463
9.5.1.19 LOCEVENTMASKNINOTIFYVERIFYREQ . . . 1464
9.5.1.20 LOCEVENTMASKNMEA . . . 1464
9.5.1.21 LOCEVENTMASKPEDOMETERCONTROL . . . 1464
9.5.1.22 LOCEVENTMASKPOSITIONREPORT . . . 1464
9.5.1.23 LOCEVENTMASKSENSORSTREAMINGREADYSTATUS . . . 1464

9.5.1.24 LOCEVENTMASKSETSPISTREAMINGREPORT . . . 1464
9.5.1.25 LOCEVENTMASKTIMESYNCREQ . . . 1464
9.5.1.26 LOCEVENTMASKVEHICLEDATAREADYSTATUS . . . 1464
9.5.1.27 LOCEVENTMASKWIFIREQ . . . 1464
9.5.1.28 MAX_LOC_NMEA_STR_LEN . . . 1464
9.5.1.29 MAX_SENSOR_DATA_LEN . . . 1464
9.5.1.30 MAX_TEMP_DATA_LEN . . . 1464
9.5.2 Enumeration Type Documentation . . . 1464
9.5.2.1 anonymous enum . . . 1465
9.5.3 Function Documentation . . . 1465
9.5.3.1 pack_loc_DeleteAssistData . . . 1465
9.5.3.2 pack_loc_EventRegister . . . 1465
9.5.3.3 pack_loc_SetExtPowerState . . . 1465
9.5.3.4 pack_loc_SetOperationMode . . . 1466
9.5.3.5 pack_loc_SLQSLOCGetBestAvailPos . . . 1466
9.5.3.6 pack_loc_SLQSLOCGetServer . . . 1467
9.5.3.7 pack_loc_SLQSLOCInjectPosition . . . 1467
9.5.3.8 pack_loc_SLQSLOCInjectSensorData . . . 1467
9.5.3.9 pack_loc_SLQSLOCInjectUTCTime . . . 1468
9.5.3.10 pack_loc_SLQSLOCSetCradleMountConfig . . . 1468
9.5.3.11 pack_loc_Start . . . 1468
9.5.3.12 pack_loc_Stop . . . 1469
9.5.3.13 unpack_loc_BestAvailPos_Ind . . . 1469
9.5.3.14 unpack_loc_DeleteAssistData . . . 1469
9.5.3.15 unpack_loc_DeleteAssistData_Ind . . . 1470
9.5.3.16 unpack_loc_EngineState_Ind . . . 1470
9.5.3.17 unpack_loc_EventNMEA_Ind . . . 1470
9.5.3.18 unpack_loc_EventRegister . . . 1471
9.5.3.19 unpack_loc_GetServer_Ind . . . 1471
9.5.3.20 unpack_loc_GnssSvInfo_Ind . . . 1471
9.5.3.21 unpack_loc_PositionRpt_Ind . . . 1472
9.5.3.22 unpack_loc_SetExtPowerConfig_Ind . . . 1472
9.5.3.23 unpack_loc_SetExtPowerState . . . 1472
9.5.3.24 unpack_loc_SetOperationMode . . . 1473
9.5.3.25 unpack_loc_SetOperationMode_Ind . . . 1473
9.5.3.26 unpack_loc_SLQSLOCGetBestAvailPos . . . 1473
9.5.3.27 unpack_loc_SLQSLOCGetServer . . . 1474
9.5.3.28 unpack_loc_SLQSLOCInjectPosition . . . 1474
9.5.3.29 unpack_loc_SLQSLOCInjectSensorData . . . 1474
9.5.3.30 unpack_loc_SLQSLOCInjectUTCTime . . . 1475

9.5.3.31 unpack_loc_SLQSLOCSetCradleMountConfig . . . 1475
9.5.3.32 unpack_loc_Start . . . 1475
9.5.3.33 unpack_loc_Stop . . . 1475
9.6 nas.h File Reference . . . 1476
9.6.1 Macro Definition Documentation . . . 1484
9.6.1.1 NAS_IMSI_M_S1_LENGTH . . . 1484
9.6.1.2 NAS_IMSI_M_S2_LENGTH . . . 1484
9.6.1.3 NAS_MAX_DESCRIPTION_LENGTH . . . 1484
9.6.1.4 NAS_MAX_NUM_NETWORKS . . . 1484
9.6.1.5 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1484
9.6.1.6 NAS_NAM_NAME_LENGTH . . . 1484
9.6.1.7 NAS_OTA_MESSAGE_MAX_BUF_SIZE . . . 1484
9.6.1.8 NAS_PLMN_LENGTH . . . 1484
9.6.1.9 NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST . . . 1484
9.6.2 Enumeration Type Documentation . . . 1484
9.6.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB . . . 1484
9.6.2.2 LITEQMI_NAS_LTE_CPHY_SCELL_STATE . . . 1484
9.6.2.3 NAS_LTE_CPHY_CA_BW_NRB_LITE . . . 1484
9.6.2.4 NAS_LTE_CPHY_SCELL_STATE_LITE . . . 1485
9.6.3 Function Documentation . . . 1485
9.6.3.1 pack_nas_GetACCOLC . . . 1485
9.6.3.2 pack_nas_GetANAAAAuthenticationStatus . . . 1485
9.6.3.3 pack_nas_GetCDMANetworkParameters . . . 1485
9.6.3.4 pack_nas_GetHomeNetwork . . . 1486
9.6.3.5 pack_nas_GetHomeNetwork3GPP2 . . . 1486
9.6.3.6 pack_nas_GetNetworkPreference . . . 1486
9.6.3.7 pack_nas_GetRFInfo . . . 1487
9.6.3.8 pack_nas_GetServingNetwork . . . 1487
9.6.3.9 pack_nas_GetServingNetworkCapabilities . . . 1487
9.6.3.10 pack_nas_GetSignalStrengths . . . 1487
9.6.3.11 pack_nas_InitiateDomainAttach . . . 1488
9.6.3.12 pack_nas_PerformNetworkScan . . . 1488
9.6.3.13 pack_nas_SetACCOLC . . . 1488
9.6.3.14 pack_nas_SetCDMANetworkParameters . . . 1489
9.6.3.15 pack_nas_SetLURejectCallback . . . 1489
9.6.3.16 pack_nas_SetNetworkPreference . . . 1490
9.6.3.17 pack_nas_SetRFInfoCallback . . . 1490
9.6.3.18 pack_nas_SLQSConfigSigInfo . . . 1490
9.6.3.19 pack_nas_SLQSGetErrorRate . . . 1490
9.6.3.20 pack_nas_SlqsGetLTECphyCAInfo . . . 1491

9.6.3.21 pack_nas_SLQSGetNetworkTime . . . 1491
9.6.3.22 pack_nas_SLQSGetOperatorNameData . . . 1491
9.6.3.23 pack_nas_SLQSGetPLMNName . . . 1492
9.6.3.24 pack_nas_SLQSGetServingSystem . . . 1492
9.6.3.25 pack_nas_SLQSGetSignalStrength . . . 1492
9.6.3.26 pack_nas_SLQSGetSysInfo . . . 1493
9.6.3.27 pack_nas_SLQSGetSysSelectionPref . . . 1493
9.6.3.28 pack_nas_SLQSInitiateNetworkRegistration . . . 1493
9.6.3.29 pack_nas_SLQSNasConfigSigInfo2 . . . 1494
9.6.3.30 pack_nas_SLQSNasGet3GPP2Subscription . . . 1494
9.6.3.31 pack_nas_SLQSNasGetCellLocationInfo . . . 1494
9.6.3.32 pack_nas_SLQSNasGetHDRColorCode . . . 1495
9.6.3.33 pack_nas_SLQSNasGetSigInfo . . . 1495
9.6.3.34 pack_nas_SLQSNasGetTxRxInfo . . . 1495
9.6.3.35 pack_nas_SLQSNasIndicationRegisterExt . . . 1496
9.6.3.36 pack_nas_SLQSNASSwiGetChannelLock . . . 1496
9.6.3.37 pack_nas_SLQSNasSwiIndicationRegister . . . 1497
9.6.3.38 pack_nas_SLQSNasSwiModemStatus . . . 1497
9.6.3.39 pack_nas_SLQSNASSwiSetChannelLock . . . 1497
9.6.3.40 pack_nas_SLQSSetBandPreference . . . 1498
9.6.3.41 pack_nas_SLQSSetSignalStrengthsCallback . . . 1498
9.6.3.42 pack_nas_SLQSSetSysSelectionPref . . . 1498
9.6.3.43 pack_nas_SLQSSwiGetHDRPersonality . . . 1499
9.6.3.44 pack_nas_SLQSSwiGetHDRProtSubtype . . . 1499
9.6.3.45 pack_nas_SLQSSwiGetHRPDStats . . . 1499
9.6.3.46 pack_nas_SLQSSwiGetLteCQI . . . 1500
9.6.3.47 pack_nas_SLQSSwiGetLteSccRxInfo . . . 1500
9.6.3.48 pack_nas_SLQSSwiNetworkDebug . . . 1500
9.6.3.49 pack_nas_SLQSSwiPSDetach . . . 1501
9.6.3.50 unpack_nas_GetACCOLC . . . 1501
9.6.3.51 unpack_nas_GetANAAAAuthenticationStatus . . . 1501
9.6.3.52 unpack_nas_GetCDMANetworkParameters . . . 1502
9.6.3.53 unpack_nas_GetHomeNetwork . . . 1502
9.6.3.54 unpack_nas_GetHomeNetwork3GPP2 . . . 1502
9.6.3.55 unpack_nas_GetNetworkPreference . . . 1503
9.6.3.56 unpack_nas_GetRFInfo . . . 1503
9.6.3.57 unpack_nas_GetServingNetwork . . . 1503
9.6.3.58 unpack_nas_GetServingNetworkCapabilities . . . 1504
9.6.3.59 unpack_nas_GetSignalStrengths . . . 1504
9.6.3.60 unpack_nas_InitiateDomainAttach . . . 1504

9.6.3.61 unpack_nas_PerformNetworkScan . . . 1505
9.6.3.62 unpack_nas_SetACCOLC . . . 1505
9.6.3.63 unpack_nas_SetCDMANetworkParameters . . . 1505
9.6.3.64 unpack_nas_SetDataCapabilitiesCallback_ind . . . 1506
9.6.3.65 unpack_nas_SetEventReportInd . . . 1506
9.6.3.66 unpack_nas_SetLURejectCallback . . . 1506
9.6.3.67 unpack_nas_SetNasLTECphyCaIndCallback_ind . . . 1506
9.6.3.68 unpack_nas_SetNetworkPreference . . . 1507
9.6.3.69 unpack_nas_SetRFInfoCallback . . . 1507
9.6.3.70 unpack_nas_SetRoamingIndicatorCallback_ind . . . 1507
9.6.3.71 unpack_nas_SetServingSystemCallback_ind . . . 1508
9.6.3.72 unpack_nas_SLQSConfigSigInfo . . . 1508
9.6.3.73 unpack_nas_SLQSGetErrorRate . . . 1508
9.6.3.74 unpack_nas_SlqsGetLTECphyCAInfo . . . 1509
9.6.3.75 unpack_nas_SLQSGetNetworkTime . . . 1509
9.6.3.76 unpack_nas_SLQSGetOperatorNameData . . . 1509
9.6.3.77 unpack_nas_SLQSGetPLMNName . . . 1510
9.6.3.78 unpack_nas_SLQSGetServingSystem . . . 1510
9.6.3.79 unpack_nas_SLQSGetSignalStrength . . . 1510
9.6.3.80 unpack_nas_SLQSGetSysInfo . . . 1511
9.6.3.81 unpack_nas_SLQSGetSysSelectionPref . . . 1511
9.6.3.82 unpack_nas_SLQSInitiateNetworkRegistration . . . 1511
9.6.3.83 unpack_nas_SLQSNasConfigSigInfo2 . . . 1512
9.6.3.84 unpack_nas_SLQSNasGet3GPP2Subscription . . . 1512
9.6.3.85 unpack_nas_SLQSNasGetCellLocationInfo . . . 1512
9.6.3.86 unpack_nas_SLQSNasGetHDRColorCode . . . 1513
9.6.3.87 unpack_nas_SLQSNasGetSigInfo . . . 1513
9.6.3.88 unpack_nas_SLQSNasGetTxRxInfo . . . 1513
9.6.3.89 unpack_nas_SLQSNasIndicationRegisterExt . . . 1514
9.6.3.90 unpack_nas_SLQSNasNetworkTimeCallBack_ind . . . 1514
9.6.3.91 unpack_nas_SLQSNasSigInfoCallback_ind . . . 1514
9.6.3.92 unpack_nas_SLQSNASSwiGetChannelLock . . . 1515
9.6.3.93 unpack_nas_SLQSNasSwiIndicationRegister . . . 1515
9.6.3.94 unpack_nas_SLQSNasSwiModemStatus . . . 1515
9.6.3.95 unpack_nas_SLQSNasSwiOTAMessageCallback_ind . . . 1516
9.6.3.96 unpack_nas_SLQSNASSwiSetChannelLock . . . 1516
9.6.3.97 unpack_nas_SLQSNasSysInfoCallback_ind . . . 1516
9.6.3.98 unpack_nas_SLQSNasTimerCallback_ind . . . 1517
9.6.3.99 unpack_nas_SLQSSetBandPreference . . . 1517
9.6.3.100 unpack_nas_SLQSSetSignalStrengthsCallback . . . 1517

9.6.3.101 unpack_nas_SLQSSetSysSelectionPref . . . 1517
9.6.3.102 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind . . . 1518
9.6.3.103 unpack_nas_SLQSSwiGetHDRPersonality . . . 1518
9.6.3.104 unpack_nas_SLQSSwiGetHDRProtSubtype . . . 1518
9.6.3.105 unpack_nas_SLQSSwiGetHRPDStats . . . 1519
9.6.3.106 unpack_nas_SLQSSwiGetLteCQI . . . 1519
9.6.3.107 unpack_nas_SLQSSwiGetLteSccRxInfo . . . 1519
9.6.3.108 unpack_nas_SLQSSwiNetworkDebug . . . 1520
9.6.3.109 unpack_nas_SLQSSwiPSDetach . . . 1520
9.7 pds.h File Reference . . . 1520
9.7.1 Function Documentation . . . 1522
9.7.1.1 pack_pds_ForceXTRADownload . . . 1522
9.7.1.2 pack_pds_GetPDSDefaults . . . 1523
9.7.1.3 pack_pds_GetPDSState . . . 1523
9.7.1.4 pack_pds_GetPortAutomaticTracking . . . 1523
9.7.1.5 pack_pds_GetServiceAutomaticTracking . . . 1524
9.7.1.6 pack_pds_GetXTRAAutomaticDownload . . . 1524
9.7.1.7 pack_pds_GetXTRANetwork . . . 1524
9.7.1.8 pack_pds_GetXTRAValidity . . . 1525
9.7.1.9 pack_pds_PDSInjectTimeReference . . . 1525
9.7.1.10 pack_pds_ResetPDSData . . . 1526
9.7.1.11 pack_pds_SetEventReportCallback . . . 1526
9.7.1.12 pack_pds_SetPDSDefaults . . . 1526
9.7.1.13 pack_pds_SetPDSState . . . 1527
9.7.1.14 pack_pds_SetPortAutomaticTracking . . . 1527
9.7.1.15 pack_pds_SetServiceAutomaticTracking . . . 1527
9.7.1.16 pack_pds_SetXTRAAutomaticDownload . . . 1528
9.7.1.17 pack_pds_SetXTRANetwork . . . 1528
9.7.1.18 pack_pds_SLQSGetAGPSConfig . . . 1529
9.7.1.19 pack_pds_SLQSGetGPSStateInfo . . . 1529
9.7.1.20 pack_pds_SLQSPDSDeterminePosition . . . 1529
9.7.1.21 pack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1530
9.7.1.22 pack_pds_SLQSPDSInjectPositionData . . . 1530
9.7.1.23 pack_pds_SLQSSetAGPSConfig . . . 1530
9.7.1.24 pack_pds_SLQSSetPositionMethodState . . . 1531
9.7.1.25 pack_pds_StartPDSTrackingSessionExt . . . 1531
9.7.1.26 pack_pds_StopPDSTrackingSession . . . 1531
9.7.1.27 unpack_pds_ForceXTRADownload . . . 1532
9.7.1.28 unpack_pds_GetPDSDefaults . . . 1532
9.7.1.29 unpack_pds_GetPDSState . . . 1532

9.7.1.30 unpack_pds_GetPortAutomaticTracking . . . 1533
9.7.1.31 unpack_pds_GetServiceAutomaticTracking . . . 1533
9.7.1.32 unpack_pds_GetXTRAAutomaticDownload . . . 1534
9.7.1.33 unpack_pds_GetXTRANetwork . . . 1534
9.7.1.34 unpack_pds_GetXTRAValidity . . . 1534
9.7.1.35 unpack_pds_PDSInjectTimeReference . . . 1535
9.7.1.36 unpack_pds_ResetPDSData . . . 1535
9.7.1.37 unpack_pds_SetEventReport_Ind . . . 1535
9.7.1.38 unpack_pds_SetEventReportCallback . . . 1536
9.7.1.39 unpack_pds_SetPDSDefaults . . . 1536
9.7.1.40 unpack_pds_SetPDSState . . . 1536
9.7.1.41 unpack_pds_SetPdsState_Ind . . . 1537
9.7.1.42 unpack_pds_SetPortAutomaticTracking . . . 1537
9.7.1.43 unpack_pds_SetServiceAutomaticTracking . . . 1537
9.7.1.44 unpack_pds_SetXTRAAutomaticDownload . . . 1538
9.7.1.45 unpack_pds_SetXTRANetwork . . . 1538
9.7.1.46 unpack_pds_SLQSGetAGPSConfig . . . 1538
9.7.1.47 unpack_pds_SLQSGetGPSStateInfo . . . 1539
9.7.1.48 unpack_pds_SLQSPDSDeterminePosition . . . 1539
9.7.1.49 unpack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1539
9.7.1.50 unpack_pds_SLQSPDSInjectPositionData . . . 1540
9.7.1.51 unpack_pds_SLQSSetAGPSConfig . . . 1540
9.7.1.52 unpack_pds_SLQSSetPositionMethodState . . . 1540
9.7.1.53 unpack_pds_StartPDSTrackingSessionExt . . . 1540
9.7.1.54 unpack_pds_StopPDSTrackingSession . . . 1541
9.8 qaCbkCatEventReportInd.h File Reference . . . 1541
9.8.1 Macro Definition Documentation . . . 1542
9.8.1.1 QMI_CAN_COMMON_EVENT_TLV_NUMBER . . . 1542
9.8.1.2 QMI_MAX_CAT_EVENT_DATA_LENGTH . . . 1542
9.8.2 Enumeration Type Documentation . . . 1542
9.8.2.1 eQMI_CAT_EVENT_REPORT_IND_TLV . . . 1542
9.8.2.2 eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH . . . 1542
9.8.3 Function Documentation . . . 1543
9.8.3.1 UpkQmiCbkCatEventReportInd . . . 1543
9.9 qaCbkSwiOmaDmEventReportInd.h File Reference . . . 1543
9.9.1 Macro Definition Documentation . . . 1543
9.9.1.1 QMI_SWIOMA_DM_CONFIG . . . 1543
9.9.1.2 QMI_SWIOMA_DM_FOTA . . . 1543
9.9.1.3 QMI_SWIOMA_DM_NOT . . . 1543
9.9.2 Enumeration Type Documentation . . . 1543

9.9.2.1 eQMI_SWIOMA_DM_EVENT_REPORT_IND . . . 1543
9.9.3 Function Documentation . . . 1544
9.9.3.1 UpkQmiCbkSwiOmaDmEventReportInd . . . 1544
9.9.3.2 UpkQmiCbkSwiOmaDmEventReportIndExt . . . 1544
9.10 qaGobiApiAudio.h File Reference . . . 1544
9.10.1 Detailed Description . . . 1544
9.10.2 Function Documentation . . . 1544
9.10.2.1 SLQSGetAudioPathConfig . . . 1544
9.10.2.2 SLQSGetAudioProfile . . . 1545
9.10.2.3 SLQSGetAudioVolTLBConfig . . . 1545
9.10.2.4 SLQSSetAudioPathConfig . . . 1546
9.10.2.5 SLQSSetAudioProfile . . . 1546
9.10.2.6 SLQSSetAudioVolTLBConfig . . . 1547
9.11 qaGobiApiCat.h File Reference . . . 1547
9.11.1 Detailed Description . . . 1547
9.11.2 Function Documentation . . . 1548
9.11.2.1 CATSendEnvelopeCommand . . . 1548
9.11.2.2 CATSendTerminalResponse . . . 1548
9.12 qaGobiApiCbk.h File Reference . . . 1549
9.12.1 Detailed Description . . . 1557
9.12.2 Macro Definition Documentation . . . 1557
9.12.2.1 CBK_DISABLE_EVENT . . . 1557
9.12.2.2 CBK_ENABLE_EVENT . . . 1557
9.12.2.3 CBK_NOCHANGE . . . 1557
9.12.2.4 DEREGISTER_EVENT . . . 1557
9.12.2.5 DEREGISTER_SRV . . . 1557
9.12.2.6 DHCP_MAX_NUM_OPTIONS . . . 1557
9.12.2.7 DHCP_OPTION_DATA_BUF_SIZE . . . 1557
9.12.2.8 EVENT_MASK_CARD . . . 1557
9.12.2.9 EVENT_MASK_DEREGISTER_ALL . . . 1557
9.12.2.10 EVENT_MASK_PHY_SLOT_STATUS . . . 1557
9.12.2.11 FIRST_INSTANCE . . . 1558
9.12.2.12 INVALID_INSTACNE . . . 1558
9.12.2.13 IPV4 . . . 1558
9.12.2.14 IPV4V6 . . . 1558
9.12.2.15 IPV6 . . . 1558
9.12.2.16 LOC_EVENT_MASK_ENG_STATE . . . 1558
9.12.2.17 LOC_EVENT_MASK_GNSS_SV_INFO . . . 1558
9.12.2.18 LOC_EVENT_MASK_INJECT_TIME . . . 1558
9.12.2.19 LOC_EVENT_MASK_NMEA . . . 1558

9.12.2.20 LOC_EVENT_MASK_SENSOR_STREAM . . . 1558
9.12.2.21 LOC_EVENT_MASK_TIME_SYNC . . . 1558
9.12.2.22 LOC_EVENT_POSITION_REPORT . . . 1558
9.12.2.23 MAX_MITIGATION_DEV_ID_LEN . . . 1558
9.12.2.24 MAX_NMEA_STR_LEN . . . 1558
9.12.2.25 MAX_NO_OF_APPLICATIONS . . . 1558
9.12.2.26 MAX_NO_OF_CALLS . . . 1558
9.12.2.27 MAX_NO_OF_FILES . . . 1558
9.12.2.28 MAX_NO_OF_SLOTS . . . 1558
9.12.2.29 MAX_NO_OF_UUSINFO . . . 1558
9.12.2.30 MAX_PATH_LENGTH . . . 1558
9.12.2.31 MAX_RADIO_INTERFACE_LIST . . . 1558
9.12.2.32 MAX_URL_ADDR_STR_LEN . . . 1558
9.12.2.33 MAXUSSDLENGTH . . . 1558
9.12.2.34 NAS_SRV . . . 1558
9.12.2.35 NUM_OF_SET . . . 1558
9.12.2.36 PDS_SRV . . . 1558
9.12.2.37 QMI_ETWS_MAX_PAYLOAD_LENGTH . . . 1558
9.12.2.38 QMI_MAX_VOICE_NUMBER_LENGTH . . . 1558
9.12.2.39 QMI_WMS_MAX_PAYLOAD_LENGTH . . . 1559
9.12.2.40 REGISTER_EVENT . . . 1559
9.12.2.41 REGISTER_SRV . . . 1559
9.12.2.42 SECOND_INSTANCE . . . 1559
9.12.2.43 SIGSTRENGTH_THRESHOLD_ARR_SZ . . . 1559
9.12.2.44 THIRD_INSTANCE . . . 1559
9.12.2.45 USSD_DCS_8BIT . . . 1559
9.12.2.46 USSD_DCS_ASCII . . . 1559
9.12.2.47 USSD_DCS_UCS2 . . . 1559
9.12.2.48 VOICE_SRV . . . 1559
9.12.2.49 WDS_SRV . . . 1559
9.12.3 Typedef Documentation . . . 1559
9.12.3.1 accelAcceptReady . . . 1559
9.12.3.2 accelTempAcceptReady . . . 1559
9.12.3.3 eDevState . . . 1560
9.12.3.4 eSMSEventType . . . 1560
9.12.3.5 gpsTime . . . 1560
9.12.3.6 gyroAcceptReady . . . 1560
9.12.3.7 gyroTempAcceptReady . . . 1561
9.12.3.8 LteNasReleaseInfo . . . 1561
9.12.3.9 modemTempNotification . . . 1562

9.12.3.10 packetSrvStatus . . . 1562
9.12.3.11 precisionDilution . . . 1563
9.12.3.12 ResetInfoNotification . . . 1564
9.12.3.13 sensorDataUsage . . . 1564
9.12.3.14 sessionInformation . . . 1565
9.12.3.15 sessionInformationExt . . . 1565
9.12.3.16 SMSAsyncRawSend . . . 1565
9.12.3.17 SMSCAddressInfo . . . 1566
9.12.3.18 SMSEtwsMessageInfo . . . 1566
9.12.3.19 SMSEtwsPlmnInfo . . . 1567
9.12.3.20 SMSEventInfo . . . 1567
9.12.3.21 SMSMessageModeInfo . . . 1568
9.12.3.22 SMSMTMessageInfo . . . 1568
9.12.3.23 SMSOnIMSInfo . . . 1568
9.12.3.24 SMSTransferRouteMTMessageInfo . . . 1568
9.12.3.25 svUsedforFix . . . 1569
9.12.3.26 SwiAvmsEventInd . . . 1569
9.12.3.27 SwiOTAMsg . . . 1569
9.12.3.28 tFNActivationStatus . . . 1570
9.12.3.29 tFNAllCallStatus . . . 1570
9.12.3.30 tFNASwiLTECphyCaInfo . . . 1570
9.12.3.31 tFNASwiOTAMsg . . . 1571
9.12.3.32 tFNAsyncRawSend . . . 1571
9.12.3.33 tFNBandPreference . . . 1571
9.12.3.34 tFNBestAvailPos . . . 1573
9.12.3.35 tFNCATEvent . . . 1573
9.12.3.36 tFNCbkUimSlotStatusChangeInd . . . 1573
9.12.3.37 tFNDataCapabilities . . . 1574
9.12.3.38 tFNDataSysStatus . . . 1574
9.12.3.39 tFNDelAssistData . . . 1574
9.12.3.40 tFNDeviceStateChange . . . 1574
9.12.3.41 tFNDHCPv4ClientLeaseStatus . . . 1575
9.12.3.42 tFNDTMFEvent . . . 1575
9.12.3.43 tFNDUNCallInfo . . . 1575
9.12.3.44 tFNEventNmea . . . 1575
9.12.3.45 tFNEventPosition . . . 1576
9.12.3.46 tFNFwDldCompletion . . . 1576
9.12.3.47 tFNGetServer . . . 1576
9.12.3.48 tFNGnssSvInfo . . . 1576
9.12.3.49 tFNHDRPersonaity . . . 1577

9.12.3.50 tFNImsaPdpStatus . . . 1577
9.12.3.51 tFNImsaRatStatus . . . 1577
9.12.3.52 tFNImsaRegStatus . . . 1577
9.12.3.53 tFNImsaSvcStatus . . . 1577
9.12.3.54 tFNImsRegMgrConfig . . . 1578
9.12.3.55 tFNImsSIPConfig . . . 1578
9.12.3.56 tFNImsSMSConfig . . . 1578
9.12.3.57 tFNImsUserConfig . . . 1578
9.12.3.58 tFNImsVoIPConfig . . . 1578
9.12.3.59 tFNInfoRec . . . 1579
9.12.3.60 tFNInjectPosition . . . 1579
9.12.3.61 tFNInjectSensorData . . . 1579
9.12.3.62 tFNInjectTimeStatus . . . 1579
9.12.3.63 tFNInjectUTCTime . . . 1579
9.12.3.64 tFNLUReject . . . 1580
9.12.3.65 tFNMemoryFull . . . 1582
9.12.3.66 tFNMessageWaiting . . . 1582
9.12.3.67 tFNMitiLvlRpt . . . 1582
9.12.3.68 tFNMobileIPStatus . . . 1582
9.12.3.69 tFNModemTempInfo . . . 1582
9.12.3.70 tFNNasTimer . . . 1583
9.12.3.71 tFNNet . . . 1583
9.12.3.72 tFNNetworkReject . . . 1583
9.12.3.73 tFNNetworkTime . . . 1583
9.12.3.74 tFNNewGPS . . . 1584
9.12.3.75 tFNNewNMEA . . . 1584
9.12.3.76 tFNNewRMTransferStatistics . . . 1585
9.12.3.77 tFNNewSMS . . . 1585
9.12.3.78 tFNOMADMState . . . 1585
9.12.3.79 tFNOpMode . . . 1586
9.12.3.80 tFNOTASPStatus . . . 1587
9.12.3.81 tFNPacketSrvState . . . 1587
9.12.3.82 tFNPDSState . . . 1587
9.12.3.83 tFNPower . . . 1588
9.12.3.84 tFNPrivacyChange . . . 1588
9.12.3.85 tFNQosNWStatus . . . 1588
9.12.3.86 tFNQosPriEvent . . . 1589
9.12.3.87 tFNQosStatus . . . 1589
9.12.3.88 tFNRankIndicator . . . 1590
9.12.3.89 tFNResetInfo . . . 1590

9.12.3.90 tFNRFInfo . . . 1590
9.12.3.91 tFNRoamingIndicator . . . 1591
9.12.3.92 tFNSDKTerminated . . . 1591
9.12.3.93 tFNSensorStreaming . . . 1591
9.12.3.94 tFNServingSystem . . . 1592
9.12.3.95 tFNSetCradleMount . . . 1592
9.12.3.96 tFNSetEngineState . . . 1592
9.12.3.97 tFNSetEventTimeSync . . . 1592
9.12.3.98 tFNSetExtPowerConfig . . . 1592
9.12.3.99 tFNSigInfo . . . 1593
9.12.3.100 tFNSignalStrength . . . 1593
9.12.3.101 tFNSLQSOMADMAlert . . . 1593
9.12.3.102 tFNSLQSQOSEvent . . . 1593
9.12.3.103 tFNSLQSSessionState . . . 1594
9.12.3.104 tFNSLQSSignalStrengths . . . 1594
9.12.3.105 tFNSLQSWDSEvent . . . 1594
9.12.3.106 tFNSMSEvents . . . 1594
9.12.3.107 tFNSUPSInfo . . . 1594
9.12.3.108 tFNSUPSNotification . . . 1595
9.12.3.109 tFNSwiAvmEventReport . . . 1595
9.12.3.110 tFNSysInfo . . . 1595
9.12.3.111 tFNSysSelectionPref . . . 1595
9.12.3.112 tFNtransLayerInfo . . . 1595
9.12.3.113 tFNtransNWRegInfo . . . 1596
9.12.3.114 tFNUIMRefresh . . . 1596
9.12.3.115 tFNUIMStatusChangeInfo . . . 1596
9.12.3.116 tFNUSSDNotification . . . 1596
9.12.3.117 tFNUSSDNoWaitIndication . . . 1597
9.12.3.118 tFNUSSDRelease . . . 1597
9.12.3.119 transLayerNotification . . . 1597
9.12.3.120 transNWRegInfoNotification . . . 1597
9.12.4 Enumeration Type Documentation . . . 1598
9.12.4.1 device_state_enum . . . 1598
9.12.4.2 eQaQMIService . . . 1598
9.12.4.3 SMSEventType . . . 1598
9.12.4.4 TYPE_OF_SERVICE_DOMAIN . . . 1598
9.12.5 Function Documentation . . . 1599
9.12.5.1 iSetCATEventCallback . . . 1599
9.12.5.2 iSetSignalStrengthCallback . . . 1599
9.12.5.3 iSLQSSetDUNCallInfoCallback . . . 1599

9.12.5.4 iSLQSSetSignalStrengthsCallback . . . 1599
9.12.5.5 iSLQSSetWdsFirstInstEventCallback . . . 1599
9.12.5.6 iSLQSSetWdsSecondInstEventCallback . . . 1599
9.12.5.7 iSLQSSetWdsThirdInstEventCallback . . . 1599
9.12.5.8 iSLQSSetWdsXferStatsFirstInstCallback . . . 1599
9.12.5.9 iSLQSSetWdsXferStatsSecondInstCallback . . . 1599
9.12.5.10 SetActivationStatusCallback . . . 1599
9.12.5.11 SetCATEventCallback . . . 1599
9.12.5.12 SetDataCapabilitiesCallback . . . 1600
9.12.5.13 SetDeviceStateChangeCbk . . . 1601
9.12.5.14 SetFwDldCompletionCbk . . . 1601
9.12.5.15 SetGPSCallback . . . 1602
9.12.5.16 SetLocBestAvailPosCallback . . . 1602
9.12.5.17 SetLocCradleMountCallback . . . 1602
9.12.5.18 SetLocDeleteAssistDataCallback . . . 1603
9.12.5.19 SetLocEngineStateCallback . . . 1603
9.12.5.20 SetLocEventMaskNMEACallback . . . 1603
9.12.5.21 SetLocEventPositionCallback . . . 1603
9.12.5.22 SetLocEventTimeSyncCallback . . . 1604
9.12.5.23 SetLocGetServerCallback . . . 1604
9.12.5.24 SetLocGnssSvInfoCallback . . . 1604
9.12.5.25 SetLocInjectSensorDataCallback . . . 1605
9.12.5.26 SetLocInjectTimeCallback . . . 1605
9.12.5.27 SetLocOpModeCallback . . . 1605
9.12.5.28 SetLocSensorStreamingCallback . . . 1605
9.12.5.29 SetLocSetExtPowerConfigCallback . . . 1606
9.12.5.30 SetLURejectCallback . . . 1606
9.12.5.31 SetMobileIPStatusCallback . . . 1606
9.12.5.32 SetNasLTECphyCaIndCallback . . . 1607
9.12.5.33 SetNetChangeCbk . . . 1607
9.12.5.34 SetNewSMSCallback . . . 1608
9.12.5.35 SetNMEACallback . . . 1608
9.12.5.36 SetOMADMStateCallback . . . 1609
9.12.5.37 SetPDSStateCallback . . . 1609
9.12.5.38 SetPowerCallback . . . 1609
9.12.5.39 SetRankIndicatorCallback . . . 1610
9.12.5.40 SetRFInfoCallback . . . 1610
9.12.5.41 SetRMTransferStatisticsCallback . . . 1610
9.12.5.42 SetRoamingIndicatorCallback . . . 1611
9.12.5.43 SetSignalStrengthCallback . . . 1611

9.12.5.44 SetSLQSOMADMAlertCallback . . . 1612
9.12.5.45 SetSLQSOMADMAlertCallbackExt . . . 1612
9.12.5.46 SetUimSlotStatusChangeCallback . . . 1612
9.12.5.47 SetUSSDNotificationCallback . . . 1613
9.12.5.48 SetUSSDNoWaitIndicationCallback . . . 1613
9.12.5.49 SetUSSDReleaseCallback . . . 1613
9.12.5.50 SLQSNasNetworkRejectCallback . . . 1614
9.12.5.51 SLQSNasNetworkTimeCallBack . . . 1614
9.12.5.52 SLQSNasSigInfo2CallBack . . . 1614
9.12.5.53 SLQSNasSigInfoCallBack . . . 1615
9.12.5.54 SLQSNasSwiOTAMessageCallback . . . 1615
9.12.5.55 SLQSNasSysInfoCallBack . . . 1616
9.12.5.56 SLQSNasTimerCallback . . . 1616
9.12.5.57 SLQSSetBandPreferenceCbk . . . 1617
9.12.5.58 SLQSSetDataSystemStatusCallback . . . 1617
9.12.5.59 SLQSSetDHCPv4ClientLeaseStatusCallback . . . 1617
9.12.5.60 SLQSSetDUNCallInfoCallback . . . 1618
9.12.5.61 SLQSSetIMSAPdpStatusCallback . . . 1618
9.12.5.62 SLQSSetIMSARatStatusCallback . . . 1619
9.12.5.63 SLQSSetIMSARegStatusCallback . . . 1619
9.12.5.64 SLQSSetIMSASvcStatusCallback . . . 1619
9.12.5.65 SLQSSetIMSSMSConfigCallback . . . 1620
9.12.5.66 SLQSSetIMSUserConfigCallback . . . 1620
9.12.5.67 SLQSSetIMSVoIPConfigCallback . . . 1621
9.12.5.68 SLQSSetLocInjectPositionCallback . . . 1621
9.12.5.69 SLQSSetLocInjectUTCTimeCallback . . . 1621
9.12.5.70 SLQSSetModemTempCallback . . . 1621
9.12.5.71 SLQSSetPacketSrvStatusCallback . . . 1622
9.12.5.72 SLQSSetQosEventCallback . . . 1622
9.12.5.73 SLQSSetQosNWStatusCallback . . . 1623
9.12.5.74 SLQSSetQosPriEventCallback . . . 1623
9.12.5.75 SLQSSetQosStatusCallback . . . 1624
9.12.5.76 SLQSSetRegMgrConfigCallback . . . 1624
9.12.5.77 SLQSSetSDKTerminatedCallback . . . 1624
9.12.5.78 SLQSSetServingSystemCallback . . . 1625
9.12.5.79 SLQSSetSessionStateCallback . . . 1625
9.12.5.80 SLQSSetSignalStrengthsCallback . . . 1626
9.12.5.81 SLQSSetSIPConfigCallback . . . 1626
9.12.5.82 SLQSSetSMSEventCallback . . . 1627
9.12.5.83 SLQSSetSwiGetResetInfoCallback . . . 1627

9.12.5.84 SLQSSetSwiHDRPersCallback . . . 1627
9.12.5.85 SLQSSetSysSelectionPrefCallBack . . . 1628
9.12.5.86 SLQSSetTransLayerInfoCallback . . . 1628
9.12.5.87 SLQSSetTransNWRegInfoCallback . . . 1629
9.12.5.88 SLQSSetWdsEventCallback . . . 1629
9.12.5.89 SLQSSetWdsTransferStatisticCallback . . . 1630
9.12.5.90 SLQSSwiAVMSEventReportCallback . . . 1630
9.12.5.91 SLQSTmdMitigationLvlRptCallback . . . 1630
9.12.5.92 SLQSUIMSetRefreshCallBack . . . 1631
9.12.5.93 SLQSUIMSetStatusChangeCallBack . . . 1631
9.12.5.94 SLQSVoiceInfoRecCallback . . . 1631
9.12.5.95 SLQSVoiceSetAllCallStatusCallBack . . . 1632
9.12.5.96 SLQSVoiceSetDTMFEventCallBack . . . 1632
9.12.5.97 SLQSVoiceSetOTASPStatusCallBack . . . 1633
9.12.5.98 SLQSVoiceSetPrivacyChangeCallBack . . . 1633
9.12.5.99 SLQSVoiceSetSUPSCallBack . . . 1634
9.12.5.100 SLQSVoiceSetSUPSNotificationCallback . . . 1634
9.12.5.101 SLQSWmsAsyncRawSendCallBack . . . 1635
9.12.5.102 SLQSWmsMemoryFullCallBack . . . 1635
9.12.5.103 SLQSWmsMessageWaitingCallBack . . . 1635
9.13 qaGobiApiDcs.h File Reference . . . 1636
9.13.1 Detailed Description . . . 1636
9.13.2 Macro Definition Documentation . . . 1637
9.13.2.1 LEN . . . 1637
9.13.2.2 PORTNAM_LEN . . . 1637
9.13.3 Function Documentation . . . 1637
9.13.3.1 QCWWAN2kConnect . . . 1637
9.13.3.2 QCWWAN2kEnumerateDevices . . . 1637
9.13.3.3 QCWWAN2kGetConnectedDeviceID . . . 1638
9.13.3.4 QCWWANConnect . . . 1638
9.13.3.5 QCWWANDisconnect . . . 1639
9.13.3.6 QCWWANEnumerateDevices . . . 1639
9.13.3.7 SetSDKImagePath . . . 1640
9.13.3.8 SLQSGetDeviceMode . . . 1640
9.13.3.9 SLQSGetNetStatistic . . . 1640
9.13.3.10 SLQSGetUsbPortNames . . . 1641
9.13.3.11 SLQSKillSDKProcess . . . 1641
9.13.3.12 SLQSSetLoggingMask . . . 1642
9.13.3.13 SLQSStart . . . 1642
9.13.3.14 SLQSStart_AVAgent . . . 1643

9.13.3.15 SLQSStartSrv . . . 1643
9.14 qaGobiApiDms.h File Reference . . . 1644
9.14.1 Detailed Description . . . 1646
9.14.2 Macro Definition Documentation . . . 1646
9.14.2.1 MAX_CUST_ID_LEN . . . 1646
9.14.2.2 MAX_CUST_VALUE_LEN . . . 1646
9.14.2.3 MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1646
9.14.2.4 MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1646
9.14.2.5 MAX_FSN_LENGTH . . . 1646
9.14.3 Typedef Documentation . . . 1646
9.14.3.1 custFeaturesInfo . . . 1646
9.14.3.2 custFeaturesSetting . . . 1648
9.14.3.3 dmsCurrentPRLInfo . . . 1650
9.14.3.4 ERIFileparams . . . 1650
9.14.3.5 serialNumbersInfo . . . 1650
9.14.3.6 SLQSSwiGetHostDevInfoParams . . . 1651
9.14.3.7 SLQSSwiGetOSInfoParams . . . 1651
9.14.3.8 SLQSSwiGetSerialNoExtParams . . . 1652
9.14.3.9 SLQSSwiSetHostDevInfoParams . . . 1652
9.14.3.10 SLQSSwiSetOSInfoParams . . . 1653
9.14.4 Function Documentation . . . 1653
9.14.4.1 ActivateAutomatic . . . 1653
9.14.4.2 GetActivationState . . . 1654
9.14.4.3 GetDeviceCapabilities . . . 1654
9.14.4.4 GetFirmwareRevision . . . 1655
9.14.4.5 GetFirmwareRevisions . . . 1656
9.14.4.6 GetHardwareRevision . . . 1656
9.14.4.7 GetIMSI . . . 1657
9.14.4.8 GetManufacturer . . . 1657
9.14.4.9 GetModelID . . . 1658
9.14.4.10 GetNetworkTime . . . 1658
9.14.4.11 GetOfflineReason . . . 1659
9.14.4.12 GetPower . . . 1660
9.14.4.13 GetPRLVersion . . . 1660
9.14.4.14 GetSerialNumbers . . . 1661
9.14.4.15 GetVoiceNumber . . . 1661
9.14.4.16 ResetToFactoryDefaults . . . 1662
9.14.4.17 SetPower . . . 1662
9.14.4.18 SLQSDmsSwiGetResetInfo . . . 1663
9.14.4.19 SLQSDmsSwiIndicationRegister . . . 1663

9.14.4.20 SLQSGetBandCapabilities . . . 1663
9.14.4.21 SLQSGetBandCapability . . . 1664
9.14.4.22 SLQSGetCurrentPRLInfo . . . 1665
9.14.4.23 SLQSGetCustFeatures . . . 1665
9.14.4.24 SLQSGetCustFeaturesV2 . . . 1666
9.14.4.25 SLQSGetERIFile . . . 1666
9.14.4.26 SLQSGetSerialNumbers . . . 1667
9.14.4.27 SLQSSetCustFeatures . . . 1667
9.14.4.28 SLQSSetCustFeaturesV2 . . . 1667
9.14.4.29 SLQSSwiClearDyingGaspStatistics . . . 1668
9.14.4.30 SLQSSwiGetCrashAction . . . 1668
9.14.4.31 SLQSSwiGetCrashInfo . . . 1668
9.14.4.32 SLQSSwiGetDyingGaspCfg . . . 1669
9.14.4.33 SLQSSwiGetDyingGaspStatistics . . . 1669
9.14.4.34 SLQSSwiGetFSN . . . 1669
9.14.4.35 SLQSSwiGetFwUpdateStatus . . . 1670
9.14.4.36 SLQSSwiGetHostDevInfo . . . 1670
9.14.4.37 SLQSSwiGetOSInfo . . . 1671
9.14.4.38 SLQSSwiGetSerialNoExt . . . 1671
9.14.4.39 SLQSSwiGetUSBComp . . . 1672
9.14.4.40 SLQSSwiSetCrashAction . . . 1672
9.14.4.41 SLQSSwiSetDyingGaspCfg . . . 1673
9.14.4.42 SLQSSwiSetHostDevInfo . . . 1673
9.14.4.43 SLQSSwiSetOSInfo . . . 1673
9.14.4.44 SLQSSwiSetUSBComp . . . 1674
9.14.4.45 SLQSUIMGetState . . . 1674
9.14.4.46 UIMChangePIN . . . 1675
9.14.4.47 UIMGetControlKeyStatus . . . 1675
9.14.4.48 UIMGetICCID . . . 1676
9.14.4.49 UIMGetPINStatus . . . 1677
9.14.4.50 UIMSetControlKeyProtection . . . 1678
9.14.4.51 UIMSetPINProtection . . . 1678
9.14.4.52 UIMUnblockControlKey . . . 1679
9.14.4.53 UIMUnblockPIN . . . 1680
9.14.4.54 UIMVerifyPIN . . . 1681
9.14.4.55 ValidateSPC . . . 1682
9.15 qaGobiApiFms.h File Reference . . . 1682
9.15.1 Detailed Description . . . 1685
9.15.2 Macro Definition Documentation . . . 1686
9.15.2.1 BUILD_ID_LEN . . . 1686

9.15.2.2 BUILD_ID_MAX_LEN . . . 1686
9.15.2.3 DEVICE_OFFLINE . . . 1686
9.15.2.4 DEVICE_RESET . . . 1686
9.15.2.5 DEVICE_SHUTDOWN . . . 1686
9.15.2.6 FIRMWARE_UPDATE_FAIL . . . 1686
9.15.2.7 FIRMWARE_UPDATE_SUCCESS . . . 1686
9.15.2.8 FIRMWARE_UPGRADE_SUCCESS . . . 1686
9.15.2.9 G3K_FIRMWARE_DOWNLOAD . . . 1686
9.15.2.10 GOBI_LISTENTRIES_MAX . . . 1686
9.15.2.11 GOBI_MBN_BUILD_ID_STR_LEN . . . 1686
9.15.2.12 GOBI_MBN_IMG_ID_STR_LEN . . . 1686
9.15.2.13 GOBI_SET_IMG_PREF_RSPLEN . . . 1686
9.15.2.14 IMG_ID_LEN . . . 1686
9.15.2.15 IMGDETAILS_LEN . . . 1686
9.15.2.16 MAX_IMAGE_IDE_ELEMENTS . . . 1686
9.15.2.17 PRI_UPDATE_FAIL . . . 1686
9.15.2.18 SLQSFWINFO_APPVERSION_SZ . . . 1686
9.15.2.19 SLQSFWINFO_BOOTVERSION_SZ . . . 1686
9.15.2.20 SLQSFWINFO_CARRIER_SZ . . . 1686
9.15.2.21 SLQSFWINFO_CUR_CARR_NAME . . . 1686
9.15.2.22 SLQSFWINFO_CUR_CARR_REV . . . 1686
9.15.2.23 SLQSFWINFO_MODELID_SZ . . . 1686
9.15.2.24 SLQSFWINFO_PACKAGEID_SZ . . . 1686
9.15.2.25 SLQSFWINFO_PRIVERSION_SZ . . . 1686
9.15.2.26 SLQSFWINFO_SKU_SZ . . . 1686
9.15.2.27 SPKG_FIRMWARE_DOWNLOAD . . . 1686
9.15.2.28 UNIQUE_ID_LEN . . . 1687
9.15.3 Enumeration Type Documentation . . . 1687
9.15.3.1 eGobiDeviceSeries . . . 1687
9.15.3.2 eGobiImageCarrier . . . 1687
9.15.3.3 eGobiImageGPS . . . 1688
9.15.3.4 eGobiImageRegion . . . 1688
9.15.3.5 eGobiImageTech . . . 1689
9.15.4 Function Documentation . . . 1689
9.15.4.1 DeleteStoredImage . . . 1689
9.15.4.2 DownloadToSlot . . . 1689
9.15.4.3 eGetDeviceSeries . . . 1690
9.15.4.4 GetImagesPreference . . . 1690
9.15.4.5 GetImageStore . . . 1691
9.15.4.6 GetStoredImages . . . 1691

9.15.4.7 SetImagesPreference . . . 1692
9.15.4.8 SLQSDownloadFirmwareToSlot . . . 1692
9.15.4.9 SLQSGetBootVersionNumber . . . 1693
9.15.4.10 SLQSGetFirmwareInfo . . . 1694
9.15.4.11 SLQSGetImageInfo . . . 1694
9.15.4.12 SLQSGetImageInfo_9x15 . . . 1695
9.15.4.13 SLQSGetImageInfoMC77xx . . . 1695
9.15.4.14 SLQSGetImageInfoMC83xx . . . 1696
9.15.4.15 SLQSGetValidFwPriCombinations . . . 1697
9.15.4.16 SLQSIsSpkgFormatRequired . . . 1697
9.15.4.17 SLQSSetCrashStateCheckIgnore . . . 1697
9.15.4.18 SLQSSetSIMBasedImageSwitching . . . 1698
9.15.4.19 SLQSSetSpkgFormatRequired . . . 1698
9.15.4.20 SLQSSwiGetAllCarrierImages . . . 1698
9.15.4.21 SLQSSwiGetFirmwareCurr . . . 1699
9.15.4.22 SLQSUpgradeFirmware9x15 . . . 1699
9.15.4.23 upgrade_mc77xx_fw . . . 1700
9.15.4.24 UpgradeFirmware2k . . . 1700
9.16 qaGobiApiIms.h File Reference . . . 1701
9.16.1 Detailed Description . . . 1702
9.16.2 Function Documentation . . . 1702
9.16.2.1 SLQSGetIMSSMSConfig . . . 1702
9.16.2.2 SLQSGetIMSUserConfig . . . 1702
9.16.2.3 SLQSGetIMSVoIPConfig . . . 1703
9.16.2.4 SLQSGetRegMgrConfig . . . 1703
9.16.2.5 SLQSGetSIPConfig . . . 1704
9.16.2.6 SLQSImsConfigIndicationRegister . . . 1704
9.16.2.7 SLQSSetIMSSMSConfig . . . 1705
9.16.2.8 SLQSSetIMSUserConfig . . . 1705
9.16.2.9 SLQSSetIMSVoIPConfig . . . 1706
9.16.2.10 SLQSSetRegMgrConfig . . . 1706
9.16.2.11 SLQSSetSIPConfig . . . 1707
9.17 qaGobiApiImsa.h File Reference . . . 1707
9.17.1 Detailed Description . . . 1708
9.17.2 Function Documentation . . . 1708
9.17.2.1 SLQSGetIMSARegStatus . . . 1708
9.17.2.2 SLQSGetIMSAServiceStatus . . . 1708
9.17.2.3 SLQSGetIMSASupportedFields . . . 1709
9.17.2.4 SLQSGetIMSASupportedMsg . . . 1709
9.17.2.5 SLQSRegisterIMSAIndication . . . 1710

9.18 qaGobiApiLoc.h File Reference . . . 1710
9.18.1 Detailed Description . . . 1711
9.18.2 Macro Definition Documentation . . . 1711
9.18.2.1 MAX_SENSOR_DATA_LEN . . . 1711
9.18.2.2 MAX_TEMP_DATA_LEN . . . 1711
9.18.3 Function Documentation . . . 1711
9.18.3.1 SLQSLOCDelAssData . . . 1712
9.18.3.2 SLQSLOCEventRegister . . . 1712
9.18.3.3 SLQSLOCGetBestAvailPos . . . 1712
9.18.3.4 SLQSLOCGetServer . . . 1713
9.18.3.5 SLQSLOCInjectPosition . . . 1713
9.18.3.6 SLQSLOCInjectSensorData . . . 1714
9.18.3.7 SLQSLOCInjectUTCTime . . . 1714
9.18.3.8 SLQSLOCSetCradleMountConfig . . . 1714
9.18.3.9 SLQSLOCSetExtPowerState . . . 1715
9.18.3.10 SLQSLOCSetOpMode . . . 1715
9.18.3.11 SLQSLOCStart . . . 1716
9.18.3.12 SLQSLOCStop . . . 1716
9.18.3.13 SwiLocGetAutoStart . . . 1717
9.18.3.14 SwiLocSetAutoStart . . . 1717
9.19 qaGobiApiNas.h File Reference . . . 1718
9.19.1 Detailed Description . . . 1723
9.19.2 Macro Definition Documentation . . . 1723
9.19.2.1 IMSI_M_S1_LENGTH . . . 1723
9.19.2.2 IMSI_M_S2_LENGTH . . . 1723
9.19.2.3 MAX_DATA_SRV_CAPABILITIES . . . 1723
9.19.2.4 MAX_DESCRIPTION_LENGTH . . . 1723
9.19.2.5 MAX_PILOT_SETS . . . 1723
9.19.2.6 MAX_SERV_SYSTEM_RADIO_INTERFACES . . . 1723
9.19.2.7 NAM_NAME_LENGTH . . . 1723
9.19.2.8 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1723
9.19.2.9 NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE . . . 1723
9.19.2.10 NAS_SIG_INFO_MIN_dB_FLOAT_VALUE . . . 1723
9.19.2.11 NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE . . . 1723
9.19.2.12 PLMN_LENGTH . . . 1723
9.19.2.13 SLQS_SS_INFO_LIST_MAX_ELEMENTS . . . 1723
9.19.2.14 SLQS_SYSTEM_ID_SIZE . . . 1723
9.19.2.15 UATISIZE . . . 1723
9.19.3 Typedef Documentation . . . 1723
9.19.3.1 SlqsNas3GppNetworkRAT . . . 1723

9.19.3.2 slqsNetworkScanInfo . . . 1724
9.19.3.3 sysSelectPrefInfo . . . 1725
9.19.3.4 sysSelectPrefParams . . . 1729
9.19.4 Enumeration Type Documentation . . . 1735
9.19.4.1 _NAMS_RADIO_IF_TECHNOLOGY_ . . . 1735
9.19.4.2 eSYS_SRV_DOMAIN . . . 1735
9.19.4.3 NAS_LTE_CPHY_CA_BW_NRB . . . 1736
9.19.4.4 NAS_LTE_CPHY_SCELL_STATE . . . 1736
9.19.5 Function Documentation . . . 1736
9.19.5.1 GetACCOLC . . . 1736
9.19.5.2 GetANAAAAuthenticationStatus . . . 1736
9.19.5.3 GetCDMANetworkParameters . . . 1737
9.19.5.4 GetHomeNetwork . . . 1739
9.19.5.5 GetHomeNetwork3GPP2 . . . 1739
9.19.5.6 GetNetworkPreference . . . 1741
9.19.5.7 GetRFInfo . . . 1742
9.19.5.8 GetServingNetwork . . . 1742
9.19.5.9 GetServingNetworkCapabilities . . . 1743
9.19.5.10 GetSignalStrengths . . . 1744
9.19.5.11 InitiateDomainAttach . . . 1745
9.19.5.12 InitiateNetworkRegistration . . . 1745
9.19.5.13 PerformNetworkScan . . . 1746
9.19.5.14 SetACCOLC . . . 1746
9.19.5.15 SetCDMANetworkParameters . . . 1747
9.19.5.16 SetNetworkPreference . . . 1748
9.19.5.17 SLQSConfigSigInfo . . . 1749
9.19.5.18 SLQSGetErrorRate . . . 1750
9.19.5.19 SLQSGetNetworkTime . . . 1750
9.19.5.20 SLQSGetOperatorNameData . . . 1750
9.19.5.21 SLQSGetPLMNName . . . 1751
9.19.5.22 SLQSGetServingSystem . . . 1751
9.19.5.23 SLQSGetSignalStrength . . . 1752
9.19.5.24 SLQSGetSysSelectionPref . . . 1752
9.19.5.25 SLQSInitiateNetworkRegistration . . . 1753
9.19.5.26 SLQSNasConfigSigInfo2 . . . 1753
9.19.5.27 SLQSNasGet3GPP2Subscription . . . 1754
9.19.5.28 SLQSNasGetCellLocationInfo . . . 1754
9.19.5.29 SLQSNasGetHDRColorCode . . . 1755
9.19.5.30 SLQSNASGetLTECPHYCaInfo . . . 1755
9.19.5.31 SLQSNasGetSigInfo . . . 1755

9.19.5.32 SLQSNasGetSysInfo . . . 1756
9.19.5.33 SLQSNasGetTxRxInfo . . . 1756
9.19.5.34 SLQSNasIndicationRegister . . . 1757
9.19.5.35 SLQSNasIndicationRegisterExt . . . 1758
9.19.5.36 SLQSNasIndicationRegisterLTECphyCa . . . 1758
9.19.5.37 SLQSNASSwiGetChannelLock . . . 1758
9.19.5.38 SLQSNasSwiIndicationRegister . . . 1759
9.19.5.39 SLQSNasSwiModemStatus . . . 1759
9.19.5.40 SLQSNASSwiSetChannelLock . . . 1760
9.19.5.41 SLQSPerformNetworkScan . . . 1760
9.19.5.42 SLQSSetBandPreference . . . 1761
9.19.5.43 SLQSSetSysSelectionPref . . . 1762
9.19.5.44 SLQSSwiGetHDRPersonality . . . 1762
9.19.5.45 SLQSSwiGetHDRProtSubtype . . . 1763
9.19.5.46 SLQSSwiGetHRPDStats . . . 1763
9.19.5.47 SLQSSwiGetLteCQI . . . 1763
9.19.5.48 SLQSSwiGetLteSccRxInfo . . . 1764
9.19.5.49 SLQSSwiNetworkDebug . . . 1764
9.19.5.50 SLQSSwiPSDetach . . . 1765
9.20 qaGobiApiOmadm.h File Reference . . . 1765
9.20.1 Detailed Description . . . 1765
9.20.2 Function Documentation . . . 1765
9.20.2.1 OMADMCancelSession . . . 1765
9.20.2.2 OMADMGetPendingNIA . . . 1766
9.20.2.3 OMADMGetSessionInfo . . . 1766
9.20.2.4 OMADMStartSession . . . 1768
9.21 qaGobiApiPds.h File Reference . . . 1768
9.21.1 Detailed Description . . . 1769
9.21.2 Macro Definition Documentation . . . 1769
9.21.2.1 DEFAULTBYTEVALUE . . . 1769
9.21.2.2 DEFAULTLONGVALUE . . . 1769
9.21.2.3 DEFAULTWORDVALUE . . . 1769
9.21.3 Enumeration Type Documentation . . . 1769
9.21.3.1 anonymous enum . . . 1769
9.21.4 Function Documentation . . . 1770
9.21.4.1 ForceXTRADownload . . . 1770
9.21.4.2 GetPDSDefaults . . . 1770
9.21.4.3 GetPDSState . . . 1771
9.21.4.4 GetPortAutomaticTracking . . . 1771
9.21.4.5 GetServiceAutomaticTracking . . . 1772

9.21.4.6 GetXTRAAutomaticDownload . . . 1772
9.21.4.7 GetXTRANetwork . . . 1773
9.21.4.8 GetXTRAValidity . . . 1773
9.21.4.9 PDSInjectTimeReference . . . 1774
9.21.4.10 ResetPDSData . . . 1774
9.21.4.11 SetPDSDefaults . . . 1775
9.21.4.12 SetPDSState . . . 1776
9.21.4.13 SetPortAutomaticTracking . . . 1776
9.21.4.14 SetServiceAutomaticTracking . . . 1777
9.21.4.15 SetXTRAAutomaticDownload . . . 1777
9.21.4.16 SetXTRANetwork . . . 1778
9.21.4.17 SLQSGetAGPSConfig . . . 1778
9.21.4.18 SLQSGetGPSStateInfo . . . 1779
9.21.4.19 SLQSPDSDeterminePosition . . . 1779
9.21.4.20 SLQSPDSInjectAbsoluteTimeReference . . . 1780
9.21.4.21 SLQSPDSInjectPositionData . . . 1780
9.21.4.22 SLQSSetAGPSConfig . . . 1781
9.21.4.23 SLQSSetPositionMethodState . . . 1781
9.21.4.24 StartPDSTrackingSessionExt . . . 1782
9.21.4.25 StopPDSTrackingSession . . . 1783
9.22 qaGobiApiQos.h File Reference . . . 1783
9.22.1 Detailed Description . . . 1784
9.22.2 Macro Definition Documentation . . . 1784
9.22.2.1 MAX_QOS_FILTER_TLV . . . 1784
9.22.2.2 MAX_QOS_SPEC_PER_APN . . . 1784
9.22.3 Function Documentation . . . 1784
9.22.3.1 SLQSQosGetFlowStatus . . . 1784
9.22.3.2 SLQSQosGetGranted . . . 1784
9.22.3.3 SLQSQosGetNetworkStatus . . . 1785
9.22.3.4 SLQSQosGetNWProf . . . 1785
9.22.3.5 SLQSQosModify . . . 1786
9.22.3.6 SLQSQosRel . . . 1786
9.22.3.7 SLQSQosReq . . . 1787
9.22.3.8 SLQSQosReset . . . 1787
9.22.3.9 SLQSQosResume . . . 1788
9.22.3.10 SLQSQosSuspend . . . 1788
9.22.3.11 SLQSQosSwiReadApnExtraParams . . . 1789
9.22.3.12 SLQSQosSwiReadDataStats . . . 1789
9.23 qaGobiApiRms.h File Reference . . . 1789
9.23.1 Detailed Description . . . 1790

9.23.2 Function Documentation . . . 1790
9.23.2.1 GetSMSWake . . . 1790
9.23.2.2 SetSMSWake . . . 1790
9.24 qaGobiApiSar.h File Reference . . . 1791
9.24.1 Detailed Description . . . 1791
9.24.2 Enumeration Type Documentation . . . 1791
9.24.2.1 eQMISARRFState . . . 1791
9.24.3 Function Documentation . . . 1792
9.24.3.1 SLQSGetRfSarState . . . 1792
9.24.3.2 SLQSSetRfSarState . . . 1793
9.25 qaGobiApiSms.h File Reference . . . 1793
9.25.1 Detailed Description . . . 1795
9.25.2 Macro Definition Documentation . . . 1795
9.25.2.1 ABSOLUTE_VALIDITY . . . 1795
9.25.2.2 CONFIG_LEN . . . 1795
9.25.2.3 MAX_SMS_ROUTES . . . 1795
9.25.2.4 NUM_OF_SET . . . 1795
9.25.2.5 TIME_DATE_BUF . . . 1795
9.25.2.6 TIME_STAMP_BUF . . . 1795
9.25.3 Typedef Documentation . . . 1795
9.25.3.1 getIndicationRegResp . . . 1796
9.25.3.2 getTransLayerInfoResp . . . 1796
9.25.3.3 getTransNWRegInfoResp . . . 1797
9.25.3.4 qaQmi3GPP2BroadcastCfgInfo . . . 1797
9.25.3.5 qaQmi3GPPBroadcastCfgInfo . . . 1797
9.25.3.6 setIndicationRegReq . . . 1798
9.25.3.7 transLayerInfo . . . 1798
9.25.4 Function Documentation . . . 1799
9.25.4.1 GetSMSCAddress . . . 1799
9.25.4.2 SaveSMS . . . 1799
9.25.4.3 SendSMS . . . 1800
9.25.4.4 SetSMSCAddress . . . 1801
9.25.4.5 SLQSCDMADecodeMTTextMsg . . . 1802
9.25.4.6 SLQSCDMAEncodeMOTextMsg . . . 1802
9.25.4.7 SLQSDeleteSMS . . . 1802
9.25.4.8 SLQSGetIndicationRegister . . . 1803
9.25.4.9 SLQSGetMessageWaiting . . . 1804
9.25.4.10 SLQSGetSMS . . . 1804
9.25.4.11 SLQSGetSmsBroadcastConfig . . . 1805
9.25.4.12 SLQSGetSMSList . . . 1806

9.25.4.13 SLQSGetTransLayerInfo . . . 1807
9.25.4.14 SLQSGetTransNWRegInfo . . . 1807
9.25.4.15 SLQSModifySMSStatus . . . 1807
9.25.4.16 SLQSSendAsyncSMS . . . 1808
9.25.4.17 SLQSSendLongSMS . . . 1809
9.25.4.18 SLQSSendSMS . . . 1809
9.25.4.19 SLQSSetIndicationRegister . . . 1810
9.25.4.20 SLQSSetSmsBroadcastActivation . . . 1810
9.25.4.21 SLQSSetSmsBroadcastConfig . . . 1811
9.25.4.22 SLQSSetSmsStorage . . . 1811
9.25.4.23 SLQSSmsGetMaxStorageSize . . . 1812
9.25.4.24 SLQSSmsGetMessageProtocol . . . 1812
9.25.4.25 SLQSSmsSetRoutes . . . 1813
9.25.4.26 SLQSSwiGetSMSStorage . . . 1813
9.25.4.27 SLQSWCDMADecodeLongTextMsg . . . 1814
9.25.4.28 SLQSWCDMADecodeMTTextMsg . . . 1814
9.25.4.29 SLQSWCDMAEncodeMOTextMsg . . . 1815
9.26 qaGobiApiSwi.h File Reference . . . 1815
9.26.1 Detailed Description . . . 1815
9.26.2 Function Documentation . . . 1815
9.26.2.1 SLQSGetPidof . . . 1815
9.26.2.2 SLQSGetSdkVersion . . . 1816
9.26.2.3 SLQSSendRawQMI . . . 1816
9.27 qaGobiApiSwiAudio.h File Reference . . . 1816
9.27.1 Detailed Description . . . 1817
9.27.2 Macro Definition Documentation . . . 1817
9.27.2.1 MAX_LEN_IFACE_TABLE . . . 1817
9.27.3 Function Documentation . . . 1817
9.27.3.1 SLQSGetM2MAudioProfile . . . 1817
9.27.3.2 SLQSGetM2MAudioVolume . . . 1818
9.27.3.3 SLQSGetM2MAVMute . . . 1818
9.27.3.4 SLQSGetM2MSpkrGain . . . 1819
9.27.3.5 SLQSSetM2MAudioAVCFG . . . 1819
9.27.3.6 SLQSSetM2MAudioLPBK . . . 1819
9.27.3.7 SLQSSetM2MAudioNVDef . . . 1820
9.27.3.8 SLQSSetM2MAudioProfile . . . 1820
9.27.3.9 SLQSSetM2MAudioVolume . . . 1820
9.27.3.10 SLQSSetM2MAVMute . . . 1821
9.27.3.11 SLQSSetM2MSpkrGain . . . 1821
9.28 qaGobiApiSwiAvms.h File Reference . . . 1822

9.28.1 Detailed Description . . . 1824
9.28.2 Macro Definition Documentation . . . 1824
9.28.2.1 MAX_AVMS_RETRY_TIMER_NUMBER . . . 1824
9.28.2.2 MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH . . . 1824
9.28.2.3 MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH . . . 1824
9.28.2.4 MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH . . . 1824
9.28.2.5 MAX_SWI_AVMS_SET_SETTING_APN_LENGTH . . . 1824
9.28.2.6 MAX_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS . . . 1824
9.28.2.7 MAX_SWI_AVMS_SET_SETTING_PWD_LENGTH . . . 1824
9.28.2.8 MAX_SWI_AVMS_SET_SETTING_UNAME_LENGTH . . . 1824
9.28.2.9 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH . . . 1824
9.28.2.10 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH . . . 1824
9.28.2.11 QMISWIAVMSSESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH . . . 1824
9.28.2.12 QMISWIAVMSSESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH . . . 1824
9.28.3 Typedef Documentation . . . 1824
9.28.3.1 QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp . . . 1824
9.28.3.2 QmiSwiAvmsEventReportConfigResp . . . 1825
9.28.3.3 QmiSwiAvmsEventReportConnectionRequestResp . . . 1825
9.28.3.4 QmiSwiAvmsEventReportDataSessionStatusResp . . . 1826
9.28.3.5 QmiSwiAvmsEventReportHTTPStatusResp . . . 1826
9.28.3.6 QmiSwiAvmsEventReportNotificationResp . . . 1827
9.28.3.7 QmiSwiAvmsEventReportPackageIDResp . . . 1828
9.28.3.8 QmiSwiAvmsEventReportRegStatusResp . . . 1828
9.28.3.9 QmiSwiAvmsEventReportSessionTypeResp . . . 1829
9.28.3.10 QmiSwiAvmsEventReportWAMSParamChangeResp . . . 1829
9.28.3.11 QmiSwiAvmsSetSettingsAPNInfo . . . 1829
9.28.3.12 QmiSwiAvmsSetSettingsConnectionRetryTimers . . . 1829
9.28.3.13 QmiSwiAvmsSetSettingsPeriodInfo . . . 1829
9.28.3.14 SLQSAVMSSetSettingsReq . . . 1829
9.28.3.15 SLQSAVMSSettings . . . 1830
9.28.3.16 SLQSAVMSSettingsAPNInfo . . . 1830
9.28.3.17 SLQSAVMSSettingsConnectionRetryTimers . . . 1830
9.28.3.18 SLQSAVMSSettingsPeriodsInfo . . . 1831
9.28.4 Function Documentation . . . 1831
9.28.4.1 SLQSAVMSGetSessionInfo . . . 1831
9.28.4.2 SLQSAVMSGetSettings . . . 1831
9.28.4.3 SLQSAVMSSelection . . . 1831
9.28.4.4 SLQSAVMSSetEventReport . . . 1832

9.28.4.5 SLQSAVMSSetSettings . . . 1832
9.28.4.6 SLQSAVMSSetSettingsIgnoreAutoReboot . . . 1833
9.28.4.7 SLQSAVMSStartSession . . . 1833
9.28.4.8 SLQSAVMSStopSession . . . 1834
9.29 qaGobiApiSwiOmadms.h File Reference . . . 1834
9.29.1 Detailed Description . . . 1835
9.29.2 Typedef Documentation . . . 1835
9.29.2.1 SLQSOMADMSessionInfo . . . 1835
9.29.2.2 SLQSOMADMSettings . . . 1837
9.29.2.3 SLQSOMADMSettingsReqParams . . . 1838
9.29.2.4 SLQSOMADMSettingsReqParams3 . . . 1839
9.29.3 Function Documentation . . . 1840
9.29.3.1 SLQSOMADMCancelSession . . . 1840
9.29.3.2 SLQSOMADMGetSessionInfo . . . 1840
9.29.3.3 SLQSOMADMGetSettings . . . 1841
9.29.3.4 SLQSOMADMGetSettings2 . . . 1841
9.29.3.5 SLQSOMADMSendSelection . . . 1842
9.29.3.6 SLQSOMADMSendSelection2 . . . 1842
9.29.3.7 SLQSOMADMSetSettings . . . 1843
9.29.3.8 SLQSOMADMSetSettings2 . . . 1844
9.29.3.9 SLQSOMADMSetSettings3 . . . 1844
9.29.3.10 SLQSOMADMStartSession . . . 1844
9.29.3.11 SLQSOMADMStartSession2 . . . 1845
9.30 qaGobiApiTableBandClasses.h File Reference . . . 1845
9.30.1 Detailed Description . . . 1845
9.30.2 Band Classes (Value - Description) . . . 1845
9.30.2.1 LTE Bands . . . 1847
9.31 qaGobiApiTableCallControlReturnReasons.h File Reference . . . 1848
9.31.1 Detailed Description . . . 1848
9.31.2 Call Control Result Reasons (Value - Name - Description) . . . 1848
9.32 qaGobiApiTableCallEndReasons.h File Reference . . . 1849
9.32.1 Detailed Description . . . 1849
9.32.2 Call end reason codes (Code - Reason) . . . 1849
9.32.2.1 Technology-agnostic call end reasons . . . 1849
9.32.2.2 EVDO CDMA 1xEV-DO . . . 1850
9.32.2.3 WCDMA/GSM call end reasons . . . 1850
9.32.2.4 EVDO CDMA 1xEV-DO . . . 1852
9.32.2.5 call end reason type . . . 1853
9.32.2.6 Mobile IP call end reasons (Type=1) . . . 1853
9.32.2.7 Internal call end reasons (Type=2) . . . 1855

9.32.2.8 Call Manager defined call end reasons (Type=3) . . . 1857
9.32.2.9 3GPP specification defined call end reasons (Type=6) . . . 1862
9.32.2.10 PPP call end reasons (Type=7) . . . 1864
9.32.2.11 EHRPD call end reasons (Type=8) . . . 1864
9.32.2.12 IPV6 call end reasons (Type=9) . . . 1865
9.33 qaGobiApiTableCarrierCodes.h File Reference . . . 1865
9.33.1 Detailed Description . . . 1865
9.33.2 Carrier Codes (Number - Carrier) . . . 1866
9.34 qaGobiApiTableCodingScheme.h File Reference . . . 1867
9.34.1 Detailed Description . . . 1867
9.34.2 Call Control Result Reasons (Value - Name - Description) . . . 1868
9.34.2.1 Use of bits 3..0\n\n . . . 1868
9.34.3 Coding Group Bits 7..4(0001) . . . 1868
9.34.3.1 use of bits 3..0 . . . 1868
9.34.4 Coding Group Bits 7..4(0010) . . . 1868
9.34.4.1 use of bits 3..0 . . . 1868
9.34.5 Coding Group Bits 7..4(0011) . . . 1869
9.34.5.1 use of bits 3..0 . . . 1869
9.34.6 Coding Group Bits 7..4(01xx) . . . 1869
9.34.6.1 use of bits 3..0 . . . 1869
9.34.7 Coding Group Bits 7..4(1001) . . . 1869
9.34.7.1 Reserved coding groups . . . 1869
9.34.8 Coding Group Bits 7..4(1010..1101) . . . 1870
9.34.8.1 Reserved coding groups . . . 1870
9.34.9 Coding Group Bits 7..4(1110) . . . 1870
9.34.9.1 Defined by the WAP Forum . . . 1870
9.34.10 Coding Group Bits 7..4 (1111) . . . 1870
9.34.10.1 Data coding / message handling . . . 1870
9.34.11 Macro Definition Documentation . . . 1870
9.34.11.1 __GOBI_API_CODING_SCHEME_H__ . . . 1870
9.35 qaGobiApiTableGpsCapabilityCodes.h File Reference . . . 1870
9.35.1 Detailed Description . . . 1870
9.35.2 GPS capability (Value - Capability) . . . 1870
9.36 qaGobiApiTablePowerModes.h File Reference . . . 1871
9.36.1 Detailed Description . . . 1871
9.36.2 Power Modes (Value - Description) . . . 1871
9.37 qaGobiApiTableRadioInterfaces.h File Reference . . . 1871
9.37.1 Detailed Description . . . 1871
9.37.2 Radio interface . . . 1871
9.37.2.1 Technology (Value - Radio Interface Technology) . . . 1871

9.38 qaGobiApiTableRegionCodes.h File Reference . . . 1872
9.38.1 Detailed Description . . . 1872
9.38.2 Region Codes (Code - Region) . . . 1872
9.39 qaGobiApiTableServiceOptions.h File Reference . . . 1872
9.39.1 Detailed Description . . . 1872
9.39.2 Service Option codes (Code - Reason) . . . 1873
9.39.2.1 Description . . . 1873
9.40 qaGobiApiTableSupServiceInfoClasses.h File Reference . . . 1875
9.40.1 Detailed Description . . . 1875
9.40.2 Supplementary Service Information Classes (Value - Service Class) . . . 1875
9.41 qaGobiApiTableSwiAudio.h File Reference . . . 1875
9.41.1 Detailed Description . . . 1875
9.41.2 ACDB Device (Device ID - description) . . . 1875
9.41.3 Physical Interface (Device ID - description - Interface parameters) . . . 1876
9.42 qaGobiApiTableSwiOMADMSessionStatus.h File Reference . . . 1876
9.42.1 Detailed Description . . . 1876
9.42.2 OMA DM Session Status (Session Status - Meaning - Usage) . . . 1876
9.43 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference . . . 1877
9.43.1 Detailed Description . . . 1877
9.43.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage) . . . 1877
9.44 qaGobiApiTableVoiceCallEndReasons.h File Reference . . . 1878
9.44.1 Detailed Description . . . 1878
9.44.2 Voice Call and supplementary services end reason codes (Code - Reason) . . . 1878
9.44.2.1 General . . . 1878
9.44.2.2 service Errors . . . 1880
9.44.2.3 control cause values . . . 1881
9.44.2.4 reject causes . . . 1883
9.44.2.5 reject causes . . . 1883
9.44.2.6 reject causes . . . 1883
9.44.2.7 stratum reject causes . . . 1884
9.44.2.8 reject causes . . . 1884
9.44.2.9 IP end reasons . . . 1884
9.45 qaGobiApiTmd.h File Reference . . . 1885
9.45.1 Detailed Description . . . 1885
9.45.2 Macro Definition Documentation . . . 1885
9.45.2.1 MAX_MITIGATION_DEV_ID_LEN . . . 1885
9.45.2.2 MAX_MITIGATION_DEV_LIST_LEN . . . 1885
9.45.3 Function Documentation . . . 1885
9.45.3.1 SLQSTmdDeRegNotMitigationLvl . . . 1885
9.45.3.2 SLQSTmdGetMitigationDevList . . . 1886

9.45.3.3 SLQSTmdGetMitigationLvl . . . 1886
9.45.3.4 SLQSTmdRegNotMitigationLvl . . . 1886
9.46 qaGobiApiUim.h File Reference . . . 1887
9.46.1 Detailed Description . . . 1889
9.46.2 Macro Definition Documentation . . . 1889
9.46.2.1 MAX_ACTIVE_PERS_FEATURES . . . 1889
9.46.2.2 MAX_CONTENT_LENGTH . . . 1889
9.46.2.3 MAX_DESCRIPTION_LENGTH . . . 1889
9.46.2.4 MAX_ICCID_LENGTH . . . 1889
9.46.2.5 MAX_NO_OF_APPLICATIONS . . . 1889
9.46.2.6 MAX_NO_OF_SLOTS . . . 1889
9.46.2.7 MAX_PATH_LENGTH . . . 1889
9.46.2.8 MAX_PUK_LENGTH . . . 1889
9.46.2.9 MAX_SLOTS_STATUS . . . 1889
9.46.3 Function Documentation . . . 1889
9.46.3.1 SLQSUIMAuthenticate . . . 1889
9.46.3.2 SLQSUIMChangePin . . . 1890
9.46.3.3 SLQSUIMDepersonalization . . . 1890
9.46.3.4 SLQSUIMEventRegister . . . 1891
9.46.3.5 SLQSUIMGetCardStatus . . . 1891
9.46.3.6 SLQSUIMGetConfiguration . . . 1892
9.46.3.7 SLQSUIMGetFileAttributes . . . 1892
9.46.3.8 SLQSUIMGetSlotsStatus . . . 1893
9.46.3.9 SLQSUIMPowerDown . . . 1893
9.46.3.10 SLQSUIMPowerUp . . . 1894
9.46.3.11 SLQSUIMReadTransparent . . . 1894
9.46.3.12 SLQSUIMRefreshComplete . . . 1895
9.46.3.13 SLQSUIMRefreshGetLastEvent . . . 1895
9.46.3.14 SLQSUIMRefreshOK . . . 1896
9.46.3.15 SLQSUIMRefreshRegister . . . 1896
9.46.3.16 SLQSUIMReset . . . 1897
9.46.3.17 SLQSUIMSetPinProtection . . . 1897
9.46.3.18 SLQSUIMSwitchSlot . . . 1898
9.46.3.19 SLQSUIMUnblockPin . . . 1898
9.46.3.20 SLQSUIMVerifyPin . . . 1899
9.47 qaGobiApiVoice.h File Reference . . . 1899
9.47.1 Detailed Description . . . 1902
9.47.2 Macro Definition Documentation . . . 1902
9.47.2.1 MAX_CALL_NO_LEN . . . 1902
9.47.2.2 MAX_DESCRIPTION_LENGTH . . . 1902

9.47.2.3 MAX_NO_OF_CALLS . . . 1902
9.47.2.4 MAXUSSDLENGTH . . . 1903
9.47.2.5 PASSWORD_LENGTH . . . 1903
9.47.3 Enumeration Type Documentation . . . 1903
9.47.3.1 serviceClassInformation . . . 1903
9.47.4 Function Documentation . . . 1903
9.47.4.1 AnswerUSSD . . . 1903
9.47.4.2 CancelUSSD . . . 1903
9.47.4.3 OriginateUSSD . . . 1904
9.47.4.4 SLQSOriginateUSSD . . . 1904
9.47.4.5 SLQSVoiceALSSelectLine . . . 1905
9.47.4.6 SLQSVoiceALSSetLineSwitching . . . 1905
9.47.4.7 SLQSVoiceAnswerCall . . . 1906
9.47.4.8 SLQSVoiceBindSubscription . . . 1906
9.47.4.9 SLQSVoiceBurstDTMF . . . 1907
9.47.4.10 SLQSVoiceDialCall . . . 1907
9.47.4.11 SLQSVoiceEndCall . . . 1908
9.47.4.12 SLQSVoiceGetAllCallInfo . . . 1908
9.47.4.13 SLQSVoiceGetCallBarring . . . 1909
9.47.4.14 SLQSVoiceGetCallForwardingStatus . . . 1909
9.47.4.15 SLQSVoiceGetCallInfo . . . 1910
9.47.4.16 SLQSVoiceGetCallWaiting . . . 1911
9.47.4.17 SLQSVoiceGetCLIP . . . 1911
9.47.4.18 SLQSVoiceGetCLIR . . . 1912
9.47.4.19 SLQSVoiceGetCNAP . . . 1912
9.47.4.20 SLQSVoiceGetCOLP . . . 1913
9.47.4.21 SLQSVoiceGetCOLR . . . 1913
9.47.4.22 SLQSVoiceGetConfig . . . 1914
9.47.4.23 SLQSVoiceIndicationRegister . . . 1914
9.47.4.24 SLQSVoiceManageCalls . . . 1915
9.47.4.25 SLQSVoiceOrigUSSDNoWait . . . 1915
9.47.4.26 SLQSVoiceSendFlash . . . 1916
9.47.4.27 SLQSVoiceSetCallBarringPassword . . . 1916
9.47.4.28 SLQSVoiceSetConfig . . . 1917
9.47.4.29 SLQSVoiceSetPreferredPrivacy . . . 1918
9.47.4.30 SLQSVoiceSetSUPSService . . . 1918
9.47.4.31 SLQSVoiceStartContDTMF . . . 1919
9.47.4.32 SLQSVoiceStopContDTMF . . . 1919
9.48 qaGobiApiWds.h File Reference . . . 1920
9.48.1 Detailed Description . . . 1923

9.48.2 Macro Definition Documentation . . . 1923
9.48.2.1 IPV6_ADDRESS_ARRAY_SIZE . . . 1923
9.48.3 Typedef Documentation . . . 1923
9.48.3.1 GetProfileSettingIn . . . 1923
9.48.3.2 GetProfileSettingOut . . . 1924
9.48.3.3 QmiProfileInfo . . . 1924
9.48.3.4 QmiWDSDataBearers . . . 1924
9.48.3.5 QmiWDSDataBearerTechnology . . . 1925
9.48.3.6 slqs3GPPConfigItem . . . 1926
9.48.4 Enumeration Type Documentation . . . 1927
9.48.4.1 qmiDataBearerMasks . . . 1927
9.48.5 Function Documentation . . . 1927
9.48.5.1 GetAutoconnect . . . 1927
9.48.5.2 GetByteTotals . . . 1927
9.48.5.3 GetConnectionRate . . . 1928
9.48.5.4 GetDataBearerTechnology . . . 1929
9.48.5.5 GetDefaultProfile . . . 1930
9.48.5.6 GetDefaultProfileLTE . . . 1931
9.48.5.7 GetDefaultProfileNum . . . 1933
9.48.5.8 GetDormancyState . . . 1933
9.48.5.9 GetIPAddressLTE . . . 1934
9.48.5.10 GetLastMobileIPError . . . 1934
9.48.5.11 GetMobileIP . . . 1935
9.48.5.12 GetMobileIPProfile . . . 1935
9.48.5.13 GetPacketStatistics . . . 1937
9.48.5.14 GetPacketStatus . . . 1937
9.48.5.15 GetSessionDuration . . . 1938
9.48.5.16 GetSessionState . . . 1939
9.48.5.17 iGetByteTotals . . . 1939
9.48.5.18 iGetConnectionRate . . . 1939
9.48.5.19 iGetPacketStatistics . . . 1939
9.48.5.20 iSLQSMISetIPFamilyPreference . . . 1939
9.48.5.21 RMSetTransferStatistics . . . 1939
9.48.5.22 SetActiveMobileIPProfile . . . 1940
9.48.5.23 SetAutoconnect . . . 1940
9.48.5.24 SetDefaultProfile . . . 1941
9.48.5.25 SetDefaultProfileLTE . . . 1942
9.48.5.26 SetDefaultProfileLTEV2 . . . 1943
9.48.5.27 SetDefaultProfileNum . . . 1945
9.48.5.28 SetMobileIP . . . 1945

9.48.5.29 SetMobileIPParameters . . . 1946
9.48.5.30 SetMobileIPProfile . . . 1947
9.48.5.31 SLQSAutoConnect . . . 1948
9.48.5.32 SLQSCreateProfile . . . 1949
9.48.5.33 SLQSDeleteProfile . . . 1949
9.48.5.34 SLQSGet3GPPConfigItem . . . 1950
9.48.5.35 SLQSGetByteTotals . . . 1950
9.48.5.36 SLQSGetConnectionRate . . . 1950
9.48.5.37 SLQSGetCurrDataSystemStat . . . 1951
9.48.5.38 SLQSGetCurrentChannelRate . . . 1951
9.48.5.39 SLQSGetDataBearerTechnology . . . 1952
9.48.5.40 SLQSGetDataBearerTechnologyExt . . . 1952
9.48.5.41 SLQSGetDUNCallInfo . . . 1953
9.48.5.42 SLQSGetPacketStatistics . . . 1953
9.48.5.43 SLQSGetProfile . . . 1954
9.48.5.44 SLQSGetProfileSettings . . . 1955
9.48.5.45 SLQSGetRuntimeSettings . . . 1956
9.48.5.46 SLQSGetSessionState . . . 1956
9.48.5.47 SLQSModifyProfile . . . 1957
9.48.5.48 SLQSResetPacketStatics . . . 1957
9.48.5.49 SLQSSet3GPPConfigItem . . . 1958
9.48.5.50 SLQSSetProfile . . . 1958
9.48.5.51 SLQSSGetDHCPv4ClientConfig . . . 1959
9.48.5.52 SLQSSGetLoopback . . . 1960
9.48.5.53 SLQSSSetDHCPv4ClientConfig . . . 1960
9.48.5.54 SLQSSSetLoopback . . . 1960
9.48.5.55 SLQSStartStopDataSession . . . 1961
9.48.5.56 SLQSWdsGoActive . . . 1961
9.48.5.57 SLQSWdsGoDormant . . . 1962
9.48.5.58 SLQSWdsSetEventReport . . . 1962
9.48.5.59 SLQSWdsSwiPDPRuntimeSettings . . . 1963
9.48.5.60 WDS_IsGobiDevice . . . 1963
9.49 qaNasGetRFBandInfo.h File Reference . . . 1963
9.49.1 Enumeration Type Documentation . . . 1963
9.49.1.1 eQMI_NAS_GET_RF_INFO_RESP . . . 1963
9.49.2 Function Documentation . . . 1964
9.49.2.1 PkQmiNasGetRFBandInfo . . . 1964
9.49.2.2 UpkQmiNasGetRFBandInfo . . . 1964
9.50 qaNasPerformNetworkScan.h File Reference . . . 1964
9.50.1 Macro Definition Documentation . . . 1964

9.50.1.1 FORBIDDEN_INDEX . . . 1964
9.50.1.2 INDEX_ZERO . . . 1964
9.50.1.3 MAX_DESCRIPTION_LENGTH . . . 1964
9.50.1.4 PREFERRED_INDEX . . . 1964
9.50.1.5 QMI_NAS_MAX_INSTANCES . . . 1964
9.50.1.6 QMI_NAS_NETSTATUS_MASK . . . 1964
9.50.1.7 ROAMING_INDEX . . . 1964
9.50.2 Enumeration Type Documentation . . . 1964
9.50.2.1 eQMI_NAS_PERFORM_NETWORK_SCAN_RESP . . . 1965
9.50.3 Function Documentation . . . 1965
9.50.3.1 PkQmiNasPerformNetworkScan . . . 1965
9.50.3.2 UpkQmiNasPerformNetworkScan . . . 1965
9.51 qmerrno.h File Reference . . . 1965
9.51.1 Enumeration Type Documentation . . . 1967
9.51.1.1 eQCWWANError . . . 1967
9.51.1.2 qm_wds_ds_profile_extended_err_codes . . . 1972
9.52 qos.h File Reference . . . 1972
9.52.1 Macro Definition Documentation . . . 1974
9.52.1.1 LITEQMI_MAX_QOS_FILTERS . . . 1974
9.52.1.2 LITEQMI_MAX_QOS_FLOW_PER_APN_STATS . . . 1974
9.52.1.3 LITEQMI_MAX_QOS_FLOWS . . . 1974
9.52.2 Function Documentation . . . 1974
9.52.2.1 pack_qos_SLQSQosGetNetworkStatus . . . 1974
9.52.2.2 pack_qos_SLQSQosSwiReadApnExtraParams . . . 1974
9.52.2.3 pack_qos_SLQSQosSwiReadDataStats . . . 1975
9.52.2.4 pack_qos_SLQSSetQosEventCallback . . . 1976
9.52.2.5 unpack_qos_SLQSQosGetNetworkStatus . . . 1976
9.52.2.6 unpack_qos_SLQSQosSwiReadApnExtraParams . . . 1977
9.52.2.7 unpack_qos_SLQSQosSwiReadDataStats . . . 1977
9.52.2.8 unpack_qos_SLQSSetQosEventCallback . . . 1978
9.52.2.9 unpack_qos_SLQSSetQosEventCallback_ind . . . 1978
9.52.2.10 unpack_qos_SLQSSetQosNWStatusCallback_ind . . . 1979
9.52.2.11 unpack_qos_SLQSSetQosPriEventCallback_ind . . . 1979
9.52.2.12 unpack_qos_SLQSSetQosStatusCallback_ind . . . 1980
9.53 rms.h File Reference . . . 1980
9.53.1 Function Documentation . . . 1981
9.53.1.1 pack_rms_GetSMSWake . . . 1981
9.53.1.2 pack_rms_SetSMSWake . . . 1981
9.53.1.3 unpack_rms_GetSMSWake . . . 1981
9.53.1.4 unpack_rms_SetSMSWake . . . 1982

9.54 sar.h File Reference . . . 1982
9.54.1 Function Documentation . . . 1982
9.54.1.1 pack_sar_SLQSGetRfSarState . . . 1982
9.54.1.2 pack_sar_SLQSSetRfSarState . . . 1983
9.54.1.3 unpack_sar_SLQSGetRfSarState . . . 1983
9.54.1.4 unpack_sar_SLQSSetRfSarState . . . 1984
9.55 sms.h File Reference . . . 1984
9.55.1 Macro Definition Documentation . . . 1987
9.55.1.1 MAX_CDMA_ENC_MO_TXT_MSG_SIZE . . . 1987
9.55.1.2 MAX_MS_TRANSFER_ROUTE_MSG . . . 1987
9.55.1.3 MAX_MSC_ADDRESS_SIZE . . . 1987
9.55.1.4 MAX_MSE_TWS_MSG . . . 1987
9.55.1.5 MAX_SMS_LIST_SIZE . . . 1987
9.55.1.6 MAX_SMS_MESSAGE_SIZE . . . 1987
9.55.1.7 SMS_CONFIG_LEN . . . 1987
9.55.1.8 SMS_MAX_SMS_ROUTES . . . 1987
9.55.1.9 SMS_NUM_OF_SET . . . 1987
9.55.1.10 SMSC_TYPE_LEN . . . 1987
9.55.2 Typedef Documentation . . . 1988
9.55.2.1 sMSCAddressInfo . . . 1988
9.55.2.2 sMSEtwsMessageInfo . . . 1988
9.55.2.3 sMSEtwsPlmnInfo . . . 1988
9.55.2.4 sMSMessageModeInfo . . . 1988
9.55.2.5 sMSMTMessageInfo . . . 1988
9.55.2.6 sMSOnIMSInfo . . . 1988
9.55.2.7 sMSTransferRouteMTMessageInfo . . . 1989
9.55.3 Enumeration Type Documentation . . . 1989
9.55.3.1 eqmiCbkSetStatus . . . 1989
9.55.4 Function Documentation . . . 1989
9.55.4.1 pack_sms_GetSMSCAddress . . . 1989
9.55.4.2 pack_sms_SaveSMS . . . 1990
9.55.4.3 pack_sms_SendSMS . . . 1990
9.55.4.4 pack_sms_SetNewSMSCallback . . . 1990
9.55.4.5 pack_sms_SetSMSCAddress . . . 1991
9.55.4.6 pack_sms_SLQSDeleteSMS . . . 1991
9.55.4.7 pack_sms_SLQSGetIndicationRegister . . . 1991
9.55.4.8 pack_sms_SLQSGetMessageWaiting . . . 1992
9.55.4.9 pack_sms_SLQSGetSMS . . . 1992
9.55.4.10 pack_sms_SLQSGetSmsBroadcastConfig . . . 1993
9.55.4.11 pack_sms_SLQSGetSMSList . . . 1993

9.55.4.12 pack_sms_SLQSGetTransLayerInfo . . . 1993
9.55.4.13 pack_sms_SLQSGetTransNWRegInfo . . . 1994
9.55.4.14 pack_sms_SLQSModifySMSStatus . . . 1994
9.55.4.15 pack_sms_SLQSSendAsyncSMS . . . 1994
9.55.4.16 pack_sms_SLQSSetIndicationRegister . . . 1995
9.55.4.17 pack_sms_SLQSSetSmsBroadcastActivation . . . 1995
9.55.4.18 pack_sms_SLQSSetSmsBroadcastConfig . . . 1995
9.55.4.19 pack_sms_SLQSSetSmsStorage . . . 1996
9.55.4.20 pack_sms_SLQSSmsGetMaxStorageSize . . . 1996
9.55.4.21 pack_sms_SLQSSmsGetMessageProtocol . . . 1997
9.55.4.22 pack_sms_SLQSSmsSetRoutes . . . 1997
9.55.4.23 pack_sms_SLQSSwiGetSMSStorage . . . 1997
9.55.4.24 unpack_sms_GetSMSCAddress . . . 1998
9.55.4.25 unpack_sms_SaveSMS . . . 1998
9.55.4.26 unpack_sms_SendSMS . . . 1998
9.55.4.27 unpack_sms_SetNewSMSCallback . . . 1999
9.55.4.28 unpack_sms_SetNewSMSCallback_ind . . . 1999
9.55.4.29 unpack_sms_SetSMSCAddress . . . 1999
9.55.4.30 unpack_sms_SLQSDeleteSMS . . . 2000
9.55.4.31 unpack_sms_SLQSGetIndicationRegister . . . 2000
9.55.4.32 unpack_sms_SLQSGetMessageWaiting . . . 2000
9.55.4.33 unpack_sms_SLQSGetSMS . . . 2001
9.55.4.34 unpack_sms_SLQSGetSmsBroadcastConfig . . . 2001
9.55.4.35 unpack_sms_SLQSGetSMSList . . . 2001
9.55.4.36 unpack_sms_SLQSGetTransLayerInfo . . . 2002
9.55.4.37 unpack_sms_SLQSGetTransNWRegInfo . . . 2002
9.55.4.38 unpack_sms_SLQSModifySMSStatus . . . 2002
9.55.4.39 unpack_sms_SLQSSendAsyncSMS . . . 2003
9.55.4.40 unpack_sms_SLQSSetIndicationRegister . . . 2003
9.55.4.41 unpack_sms_SLQSSetSmsBroadcastActivation . . . 2003
9.55.4.42 unpack_sms_SLQSSetSmsBroadcastConfig . . . 2004
9.55.4.43 unpack_sms_SLQSSetSmsStorage . . . 2004
9.55.4.44 unpack_sms_SLQSSmsGetMaxStorageSize . . . 2004
9.55.4.45 unpack_sms_SLQSSmsGetMessageProtocol . . . 2005
9.55.4.46 unpack_sms_SLQSSmsSetRoutes . . . 2005
9.55.4.47 unpack_sms_SLQSSwiGetSMSStorage . . . 2005
9.55.4.48 unpack_sms_SLQSWmsMemoryFullCallBack_ind . . . 2006
9.56 SwiDataTypes.h File Reference . . . 2006
9.56.1 Detailed Description . . . 2007
9.56.2 Macro Definition Documentation . . . 2007

9.56.2.1 QMI_NO_LTE_FW_SUPPORT . . . 2007
9.56.2.2 QMI_TLV_PLACEHOLDER . . . 2007
9.56.2.3 SWI_API . . . 2007
9.56.2.4 UNUSEDPARAM . . . 2007
9.56.3 Typedef Documentation . . . 2007
9.56.3.1 BOOL . . . 2007
9.56.3.2 BYTE . . . 2007
9.56.3.3 CHAR . . . 2007
9.56.3.4 FLOAT . . . 2007
9.56.3.5 INT32 . . . 2007
9.56.3.6 INT8 . . . 2007
9.56.3.7 LPCSTR . . . 2007
9.56.3.8 SHORT . . . 2007
9.56.3.9 ULONG . . . 2007
9.56.3.10 ULONGLONG . . . 2007
9.56.3.11 USHORT . . . 2007
9.56.3.12 WORD . . . 2007
9.57 swiloc.h File Reference . . . 2007
9.57.1 Function Documentation . . . 2008
9.57.1.1 pack_swiloc_SwiLocGetAutoStart . . . 2008
9.57.1.2 pack_swiloc_SwiLocSetAutoStart . . . 2008
9.57.1.3 unpack_swiloc_SwiLocGetAutoStart . . . 2008
9.57.1.4 unpack_swiloc_SwiLocSetAutoStart . . . 2009
9.58 swioma.h File Reference . . . 2009
9.58.1 Macro Definition Documentation . . . 2010
9.58.1.1 LITEQMI_MAX_SWIOMA_STR_LEN . . . 2010
9.58.2 Function Documentation . . . 2010
9.58.2.1 pack_swioma_SLQSOMADMAlertCallback . . . 2010
9.58.2.2 pack_swioma_SLQSOMADMCancelSession . . . 2011
9.58.2.3 pack_swioma_SLQSOMADMGetSessionInfo . . . 2011
9.58.2.4 pack_swioma_SLQSOMADMGetSettings . . . 2012
9.58.2.5 pack_swioma_SLQSOMADMSendSelection . . . 2013
9.58.2.6 pack_swioma_SLQSOMADMSetSettings . . . 2013
9.58.2.7 pack_swioma_SLQSOMADMStartSession . . . 2014
9.58.2.8 unpack_swioma_SLQSOMADMAlertCallback . . . 2014
9.58.2.9 unpack_swioma_SLQSOMADMAlertCallback_ind . . . 2015
9.58.2.10 unpack_swioma_SLQSOMADMCancelSession . . . 2015
9.58.2.11 unpack_swioma_SLQSOMADMGetSessionInfo . . . 2016
9.58.2.12 unpack_swioma_SLQSOMADMGetSettings . . . 2016
9.58.2.13 unpack_swioma_SLQSOMADMSendSelection . . . 2017

9.58.2.14 unpack_swioma_SLQSOMADMSetSettings . . . 2017
9.58.2.15 unpack_swioma_SLQSOMADMStartSession . . . 2018
9.59 SWIWWANCMAPI.h File Reference . . . 2018
9.60 uim.h File Reference . . . 2018
9.60.1 Macro Definition Documentation . . . 2021
9.60.1.1 MAX_DESCRIPTION_LENGTH . . . 2021
9.60.1.2 MAX_ICCID_LENGTH . . . 2021
9.60.1.3 MAX_NO_OF_APPLICATIONS . . . 2021
9.60.1.4 MAX_NO_OF_SLOTS . . . 2021
9.60.1.5 MAX_SLOTS_STATUS . . . 2021
9.60.1.6 UIM_MAX_ACTIVE_PERS_FEATURES . . . 2021
9.60.1.7 UIM_MAX_CONTENT_LENGTH . . . 2021
9.60.1.8 UIM_MAX_DESCRIPTION_LENGTH . . . 2021
9.60.1.9 UIM_MAX_NO_OF_APPLICATIONS . . . 2021
9.60.1.10 UIM_MAX_NO_OF_SLOTS . . . 2021
9.60.1.11 UIM_UINT8_MAX_STRING_SZ . . . 2021
9.60.2 Function Documentation . . . 2021
9.60.2.1 pack_uim_ChangePin . . . 2021
9.60.2.2 pack_uim_GetCardStatus . . . 2022
9.60.2.3 pack_uim_ReadTransparent . . . 2022
9.60.2.4 pack_uim_SetPinProtection . . . 2022
9.60.2.5 pack_uim_SLQSUIMAuthenticate . . . 2023
9.60.2.6 pack_uim_SLQSUIMDepersonalization . . . 2023
9.60.2.7 pack_uim_SLQSUIMEventRegister . . . 2024
9.60.2.8 pack_uim_SLQSUIMGetConfiguration . . . 2024
9.60.2.9 pack_uim_SLQSUIMGetFileAttributes . . . 2024
9.60.2.10 pack_uim_SLQSUIMGetSlotsStatus . . . 2025
9.60.2.11 pack_uim_SLQSUIMPowerDown . . . 2025
9.60.2.12 pack_uim_SLQSUIMPowerUp . . . 2025
9.60.2.13 pack_uim_SLQSUIMRefreshComplete . . . 2026
9.60.2.14 pack_uim_SLQSUIMRefreshGetLastEvent . . . 2026
9.60.2.15 pack_uim_SLQSUIMRefreshOK . . . 2026
9.60.2.16 pack_uim_SLQSUIMRefreshRegister . . . 2027
9.60.2.17 pack_uim_SLQSUIMReset . . . 2027
9.60.2.18 pack_uim_SLQSUIMSwitchSlot . . . 2028
9.60.2.19 pack_uim_UnblockPin . . . 2028
9.60.2.20 pack_uim_VerifyPin . . . 2028
9.60.2.21 unpack_uim_ChangePin . . . 2029
9.60.2.22 unpack_uim_GetCardStatus . . . 2029
9.60.2.23 unpack_uim_ReadTransparent . . . 2029

9.60.2.24 unpack_uim_SetPinProtection . . . 2030
9.60.2.25 unpack_uim_SetUimSlotStatusChangeCallback_ind . . . 2030
9.60.2.26 unpack_uim_SLQSUIMAuthenticate . . . 2030
9.60.2.27 unpack_uim_SLQSUIMDepersonalization . . . 2031
9.60.2.28 unpack_uim_SLQSUIMEventRegister . . . 2031
9.60.2.29 unpack_uim_SLQSUIMGetConfiguration . . . 2032
9.60.2.30 unpack_uim_SLQSUIMGetFileAttributes . . . 2032
9.60.2.31 unpack_uim_SLQSUIMGetSlotsStatus . . . 2032
9.60.2.32 unpack_uim_SLQSUIMPowerDown . . . 2033
9.60.2.33 unpack_uim_SLQSUIMPowerUp . . . 2033
9.60.2.34 unpack_uim_SLQSUIMRefreshComplete . . . 2033
9.60.2.35 unpack_uim_SLQSUIMRefreshGetLastEvent . . . 2034
9.60.2.36 unpack_uim_SLQSUIMRefreshOK . . . 2034
9.60.2.37 unpack_uim_SLQSUIMRefreshRegister . . . 2034
9.60.2.38 unpack_uim_SLQSUIMReset . . . 2035
9.60.2.39 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind . . . 2035
9.60.2.40 unpack_uim_SLQSUIMSwitchSlot . . . 2035
9.60.2.41 unpack_uim_UnblockPin . . . 2036
9.60.2.42 unpack_uim_VerifyPin . . . 2036
9.61 voice.h File Reference . . . 2036
9.61.1 Macro Definition Documentation . . . 2041
9.61.1.1 BARRING_PASSWORD_LENGTH . . . 2041
9.61.1.2 MAX_VOICE_CALL_NO_LEN . . . 2041
9.61.1.3 MAX_VOICE_DESCRIPTION_LENGTH . . . 2041
9.61.1.4 MAXVOICEUSSDLENGTH . . . 2041
9.61.1.5 VOICE_MAX_NO_OF_CALLS . . . 2041
9.61.2 Enumeration Type Documentation . . . 2041
9.61.2.1 liteServiceClassInformation . . . 2041
9.61.3 Function Documentation . . . 2041
9.61.3.1 pack_voice_AnswerUSSD . . . 2041
9.61.3.2 pack_voice_CancelUSSD . . . 2041
9.61.3.3 pack_voice_OriginateUSSD . . . 2042
9.61.3.4 pack_voice_SLQSOriginateUSSD . . . 2042
9.61.3.5 pack_voice_SLQSVoiceALSSelectLine . . . 2043
9.61.3.6 pack_voice_SLQSVoiceALSSetLineSwitching . . . 2043
9.61.3.7 pack_voice_SLQSVoiceAnswerCall . . . 2043
9.61.3.8 pack_voice_SLQSVoiceBindSubscription . . . 2044
9.61.3.9 pack_voice_SLQSVoiceBurstDTMF . . . 2044
9.61.3.10 pack_voice_SLQSVoiceDialCall . . . 2044
9.61.3.11 pack_voice_SLQSVoiceEndCall . . . 2045

9.61.3.12 pack_voice_SLQSVoiceGetAllCallInfo . . . 2045
9.61.3.13 pack_voice_SLQSVoiceGetCallBarring . . . 2046
9.61.3.14 pack_voice_SLQSVoiceGetCallForwardingStatus . . . 2046
9.61.3.15 pack_voice_SLQSVoiceGetCallInfo . . . 2046
9.61.3.16 pack_voice_SLQSVoiceGetCallWaiting . . . 2047
9.61.3.17 pack_voice_SLQSVoiceGetCLIP . . . 2047
9.61.3.18 pack_voice_SLQSVoiceGetCLIR . . . 2047
9.61.3.19 pack_voice_SLQSVoiceGetCNAP . . . 2048
9.61.3.20 pack_voice_SLQSVoiceGetCOLP . . . 2048
9.61.3.21 pack_voice_SLQSVoiceGetCOLR . . . 2048
9.61.3.22 pack_voice_SLQSVoiceGetConfig . . . 2049
9.61.3.23 pack_voice_SLQSVoiceIndicationRegister . . . 2049
9.61.3.24 pack_voice_SLQSVoiceManageCalls . . . 2049
9.61.3.25 pack_voice_SLQSVoiceOrigUSSDNoWait . . . 2050
9.61.3.26 pack_voice_SLQSVoiceSendFlash . . . 2050
9.61.3.27 pack_voice_SLQSVoiceSetCallBarringPassword . . . 2051
9.61.3.28 pack_voice_SLQSVoiceSetConfig . . . 2051
9.61.3.29 pack_voice_SLQSVoiceSetPreferredPrivacy . . . 2051
9.61.3.30 pack_voice_SLQSVoiceSetSUPSService . . . 2052
9.61.3.31 pack_voice_SLQSVoiceStartContDTMF . . . 2052
9.61.3.32 pack_voice_SLQSVoiceStopContDTMF . . . 2052
9.61.3.33 unpack_voice_AnswerUSSD . . . 2053
9.61.3.34 unpack_voice_CancelUSSD . . . 2053
9.61.3.35 unpack_voice_OriginateUSSD . . . 2053
9.61.3.36 unpack_voice_SLQSOriginateUSSD . . . 2054
9.61.3.37 unpack_voice_SLQSVoiceALSSelectLine . . . 2054
9.61.3.38 unpack_voice_SLQSVoiceALSSetLineSwitching . . . 2054
9.61.3.39 unpack_voice_SLQSVoiceAnswerCall . . . 2055
9.61.3.40 unpack_voice_SLQSVoiceBindSubscription . . . 2055
9.61.3.41 unpack_voice_SLQSVoiceBurstDTMF . . . 2055
9.61.3.42 unpack_voice_SLQSVoiceDialCall . . . 2056
9.61.3.43 unpack_voice_SLQSVoiceEndCall . . . 2056
9.61.3.44 unpack_voice_SLQSVoiceGetAllCallInfo . . . 2057
9.61.3.45 unpack_voice_SLQSVoiceGetCallBarring . . . 2057
9.61.3.46 unpack_voice_SLQSVoiceGetCallForwardingStatus . . . 2057
9.61.3.47 unpack_voice_SLQSVoiceGetCallInfo . . . 2058
9.61.3.48 unpack_voice_SLQSVoiceGetCallWaiting . . . 2058
9.61.3.49 unpack_voice_SLQSVoiceGetCLIP . . . 2058
9.61.3.50 unpack_voice_SLQSVoiceGetCLIR . . . 2059
9.61.3.51 unpack_voice_SLQSVoiceGetCNAP . . . 2059

9.61.3.52 unpack_voice_SLQSVoiceGetCOLP . . . 2059
9.61.3.53 unpack_voice_SLQSVoiceGetCOLR . . . 2060
9.61.3.54 unpack_voice_SLQSVoiceGetConfig . . . 2060
9.61.3.55 unpack_voice_SLQSVoiceIndicationRegister . . . 2060
9.61.3.56 unpack_voice_SLQSVoiceManageCalls . . . 2061
9.61.3.57 unpack_voice_SLQSVoiceOrigUSSDNoWait . . . 2061
9.61.3.58 unpack_voice_SLQSVoiceSendFlash . . . 2061
9.61.3.59 unpack_voice_SLQSVoiceSetCallBarringPassword . . . 2062
9.61.3.60 unpack_voice_SLQSVoiceSetConfig . . . 2062
9.61.3.61 unpack_voice_SLQSVoiceSetPreferredPrivacy . . . 2062
9.61.3.62 unpack_voice_SLQSVoiceSetSUPSService . . . 2063
9.61.3.63 unpack_voice_SLQSVoiceStartContDTMF . . . 2063
9.61.3.64 unpack_voice_SLQSVoiceStopContDTMF . . . 2063
9.62 wds.h File Reference . . . 2064
9.62.1 Macro Definition Documentation . . . 2069
9.62.1.1 BYT_STAT_STAT_MASK . . . 2069
9.62.1.2 IPV6_ADDRESS_ARRAY_SIZE . . . 2069
9.62.1.3 MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE . . . 2069
9.62.1.4 PACK_WDS_IPV4 . . . 2069
9.62.1.5 PACK_WDS_IPV6 . . . 2069
9.62.1.6 WDS_DHCP_MAX_NUM_OPTIONS . . . 2070
9.62.1.7 WDS_DHCP_OPTION_DATA_BUF_SIZE . . . 2070
9.62.2 Typedef Documentation . . . 2070
9.62.2.1 UnpackQmiProfileInfo . . . 2070
9.62.3 Enumeration Type Documentation . . . 2070
9.62.3.1 liteQmiDataBearerMasks . . . 2070
9.62.4 Function Documentation . . . 2070
9.62.4.1 pack_wds_DHCPv4ClientLeaseChange . . . 2070
9.62.4.2 pack_wds_GetAutoconnect . . . 2071
9.62.4.3 pack_wds_GetByteTotals . . . 2071
9.62.4.4 pack_wds_GetConnectionRate . . . 2071
9.62.4.5 pack_wds_GetDataBearerTechnology . . . 2072
9.62.4.6 pack_wds_GetDefaultProfile . . . 2072
9.62.4.7 pack_wds_GetDefaultProfileNum . . . 2073
9.62.4.8 pack_wds_GetDormancyState . . . 2073
9.62.4.9 pack_wds_GetLastMobileIPError . . . 2073
9.62.4.10 pack_wds_GetMobileIP . . . 2074
9.62.4.11 pack_wds_GetMobileIPProfile . . . 2074
9.62.4.12 pack_wds_GetPacketStatistics . . . 2075
9.62.4.13 pack_wds_GetPacketStatus . . . 2075

9.62.4.14 pack_wds_GetSessionDuration . . . 2076
9.62.4.15 pack_wds_GetSessionState . . . 2076
9.62.4.16 pack_wds_RMSetTransferStatistics . . . 2076
9.62.4.17 pack_wds_SetAutoconnect . . . 2077
9.62.4.18 pack_wds_SetDefaultProfile . . . 2077
9.62.4.19 pack_wds_SetDefaultProfileNum . . . 2078
9.62.4.20 pack_wds_SetMobileIP . . . 2078
9.62.4.21 pack_wds_SetMobileIPParameters . . . 2078
9.62.4.22 pack_wds_SetMobileIPProfile . . . 2079
9.62.4.23 pack_wds_SLQSCreateProfile . . . 2079
9.62.4.24 pack_wds_SLQSDeleteProfile . . . 2080
9.62.4.25 pack_wds_SLQSGet3GPPConfigItem . . . 2080
9.62.4.26 pack_wds_SLQSGetCurrDataSystemStat . . . 2080
9.62.4.27 pack_wds_SLQSGetCurrentChannelRate . . . 2081
9.62.4.28 pack_wds_SLQSGetDataBearerTechnology . . . 2081
9.62.4.29 pack_wds_SLQSGetDUNCallInfo . . . 2082
9.62.4.30 pack_wds_SLQSGetProfileSettings . . . 2082
9.62.4.31 pack_wds_SLQSGetRuntimeSettings . . . 2083
9.62.4.32 pack_wds_SLQSModifyProfile . . . 2083
9.62.4.33 pack_wds_SLQSResetPacketStatics . . . 2083
9.62.4.34 pack_wds_SLQSSet3GPPConfigItem . . . 2084
9.62.4.35 pack_wds_SLQSSetIPFamilyPreference . . . 2084
9.62.4.36 pack_wds_SLQSSetWdsEventCallback . . . 2085
9.62.4.37 pack_wds_SLQSSGetDHCPv4ClientConfig . . . 2085
9.62.4.38 pack_wds_SLQSSGetLoopback . . . 2085
9.62.4.39 pack_wds_SLQSSSetDHCPv4ClientConfig . . . 2086
9.62.4.40 pack_wds_SLQSSSetLoopback . . . 2086
9.62.4.41 pack_wds_SLQSStartDataSession . . . 2087
9.62.4.42 pack_wds_SLQSStopDataSession . . . 2087
9.62.4.43 pack_wds_SLQSWdsGoActive . . . 2088
9.62.4.44 pack_wds_SLQSWdsGoDormant . . . 2088
9.62.4.45 pack_wds_SLQSWdsSetEventReport . . . 2088
9.62.4.46 pack_wds_SLQSWdsSwiPDPRuntimeSettings . . . 2089
9.62.4.47 unpack_wds_DHCPv4ClientLease_ind . . . 2089
9.62.4.48 unpack_wds_DHCPv4ClientLeaseChange . . . 2090
9.62.4.49 unpack_wds_GetAutoconnect . . . 2090
9.62.4.50 unpack_wds_GetByteTotals . . . 2090
9.62.4.51 unpack_wds_GetConnectionRate . . . 2091
9.62.4.52 unpack_wds_GetDataBearerTechnology . . . 2091
9.62.4.53 unpack_wds_GetDefaultProfile . . . 2091

9.62.4.54 unpack_wds_GetDefaultProfileNum . . . 2092
9.62.4.55 unpack_wds_GetDormancyState . . . 2092
9.62.4.56 unpack_wds_GetLastMobileIPError . . . 2092
9.62.4.57 unpack_wds_GetMobileIP . . . 2093
9.62.4.58 unpack_wds_GetMobileIPProfile . . . 2093
9.62.4.59 unpack_wds_GetPacketStatistics . . . 2093
9.62.4.60 unpack_wds_GetPacketStatus . . . 2094
9.62.4.61 unpack_wds_GetSessionDuration . . . 2094
9.62.4.62 unpack_wds_GetSessionState . . . 2094
9.62.4.63 unpack_wds_RMSetTransferStatistics . . . 2095
9.62.4.64 unpack_wds_RMTransferStatistics_ind . . . 2095
9.62.4.65 unpack_wds_SetAutoconnect . . . 2095
9.62.4.66 unpack_wds_SetDefaultProfile . . . 2096
9.62.4.67 unpack_wds_SetDefaultProfileNum . . . 2096
9.62.4.68 unpack_wds_SetMobileIP . . . 2096
9.62.4.69 unpack_wds_SetMobileIPParameters . . . 2097
9.62.4.70 unpack_wds_SetMobileIPProfile . . . 2097
9.62.4.71 unpack_wds_SLQSCreateProfile . . . 2097
9.62.4.72 unpack_wds_SLQSDeleteProfile . . . 2098
9.62.4.73 unpack_wds_SLQSGet3GPPConfigItem . . . 2098
9.62.4.74 unpack_wds_SLQSGetCurrDataSystemStat . . . 2098
9.62.4.75 unpack_wds_SLQSGetCurrentChannelRate . . . 2099
9.62.4.76 unpack_wds_SLQSGetDataBearerTechnology . . . 2099
9.62.4.77 unpack_wds_SLQSGetDUNCallInfo . . . 2099
9.62.4.78 unpack_wds_SLQSGetProfileSettings . . . 2100
9.62.4.79 unpack_wds_SLQSGetRuntimeSettings . . . 2100
9.62.4.80 unpack_wds_SLQSModifyProfile . . . 2100
9.62.4.81 unpack_wds_SLQSResetPacketStatics . . . 2101
9.62.4.82 unpack_wds_SLQSSet3GPPConfigItem . . . 2101
9.62.4.83 unpack_wds_SLQSSetIPFamilyPreference . . . 2101
9.62.4.84 unpack_wds_SLQSSetPacketSrvStatusCallback . . . 2102
9.62.4.85 unpack_wds_SLQSSetWdsEventCallback . . . 2102
9.62.4.86 unpack_wds_SLQSSetWdsEventCallback_ind . . . 2102
9.62.4.87 unpack_wds_SLQSSGetDHCPv4ClientConfig . . . 2103
9.62.4.88 unpack_wds_SLQSSGetLoopback . . . 2103
9.62.4.89 unpack_wds_SLQSSSetDHCPv4ClientConfig . . . 2103
9.62.4.90 unpack_wds_SLQSSSetLoopback . . . 2104
9.62.4.91 unpack_wds_SLQSStartDataSession . . . 2104
9.62.4.92 unpack_wds_SLQSStopDataSession . . . 2104
9.62.4.93 unpack_wds_SLQSWdsGoActive . . . 2105

9.62.4.94 unpack_wds_SLQSWdsGoDormant 2105
9.62.4.95 unpack_wds_SLQSWdsSetEventReport 2105
9.62.4.96 unpack_wds_SLQSWdsSwiPDPRuntimeSettings 2105

Index **2107**

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 . . . 175
cellParams . . . 175
changeUIMPIN . . . 176
channelRate . . . 177
ChannelRate . . . 178
CLIPResp . . . 178
CLIRResp . . . 179
ClkInfo . . . 180
CNAPResp . . . 181
COLPResp . . . 182
COLRResp . . . 183
CommInfo . . . 183
ConnectionStatus . . . 185
connectionStatus . . . 186
connectNumInfo . . . 187
CrashInfo . . . 188
crashInfoParams . . . 189
CrashInfoParams . . . 190
crashInformation . . . 190
CreateProfileIn . . . 191
CreateProfileOut . . . 192
CsgId . . . 193
CSGID . . . 193
CUGInfo . . . 194
curAMRConfig . . . 195
CurrDataSysStat . . . 196
currentCatEvent . . . 197
CurrentImgList . . . 197
currentPLMN . . . 198
CurrImageInfo . . . 199
CurrNetworkInfo . . . 200
currNetworkInfo . . . 202
custFeaturesInfo . . . 203
custFeaturesSetting . . . 206
custSettingInfo . . . 207
custSettingList . . . 208
dataBearers . . . 209
DataBearerTech . . . 209
DataBearerTechExt . . . 211

dataBearerTechnology . . . 212
dataRate . . . 213
dataSrvCapabilities . . . 214
DataStatusDetail . . . 215
DataULongLongTlv . . . 217
DataULongTlv . . . 217
DcsUsbPortNames . . . 217
delAssistDataStatus . . . 217
depersonalizationInformation . . . 218
detailSvcInfo . . . 219
DeviceConfigDetail . . . 221
DHCPOption . . . 222
DHCPOptionList . . . 222
diagInfo . . . 223
dirNum . . . 223
dms_ActivationStatusTlv . . . 224
dms_OperatingModeTlv . . . 225
dmsCurrentPRLInfo . . . 225
DMScustSettingInfo . . . 226
DMScustSettingList . . . 227
DMSgetCustomFeatureV2 . . . 227
DMSgetCustomInput . . . 228
dmsIndicationRegisterReq . . . 229
dmsSwiGetResetInfo . . . 229
Domain . . . 230
DomainNameList . . . 230
DRCParams . . . 231
DTMFInfo . . . 231
DTMFLengths . . . 232
DUNCallInfoInd . . . 233
dunchannelRate . . . 234
ecioListElement . . . 234
ECIOThresh . . . 235
ECTNum . . . 236
encryptedPIN1 . . . 236
eriDataparams . . . 237
ERIFileparams . . . 237
errorRateListElement . . . 238
eTWSPLMNInfoTlv . . . 239
extDispRecInfo . . . 239
FactorySequenceNumber . . . 240
fileAttributes . . . 240
fileInfo . . . 244
FirmwareUpdatStat . . . 244
FMSImageElement . . . 246
FMSImageIdElement . . . 247
FMSImageIDEntries . . . 248
FMSImageList . . . 248
FMSPrefImageList . . . 249
fwinfo_s . . . 249
GERANInfo . . . 250
geranInstInfo . . . 252
getAllCallInformation . . . 253
getAllCallRmtPtyName . . . 253
getAllCallRmtPtyNum . . . 254
GetAudioPathConfigReq . . . 254
GetAudioPathConfigResp . . . 255
GetAudioProfileReq . . . 257

GetAudioProfileResp . . . 257
GetAudioVolTLBConfigReq . . . 259
GetAudioVolTLBConfigResp . . . 259
getCallFWExtInfo . . . 260
getCallFWInfo . . . 260
getCustomFeatureV2 . . . 261
getCustomInput . . . 262
getDUNCallInfoReq . . . 262
getDUNCallInfoResp . . . 264
getDyingGaspCfg . . . 266
getDyingGaspStatistics . . . 267
GetErrRateResp . . . 268
GetHRPDStatsResp . . . 269
GetIMSSMSConfigParams . . . 269
GetIMSUserConfigParams . . . 270
GetIMSVoIPConfigResp . . . 271
GetInstIDResp . . . 273
GetM2MAudioProfileReq . . . 274
GetM2MAudioProfileResp . . . 274
GetM2MAudioVolumeReq . . . 275
GetM2MAudioVolumeResp . . . 276
GetM2MAVMuteReq . . . 276
GetM2MAVMuteResp . . . 277
GetM2MSpkrGainReq . . . 277
GetM2MSpkrGainResp . . . 278
getMsgWaitingInfo . . . 278
GetNetworkTimeResp . . . 279
GetRegMgrConfigParams . . . 279
GetSessionIDResp . . . 280
GetSIPConfigResp . . . 280
GnssData . . . 282
gnssSvInfoNotification . . . 283
GPRSQoS . . . 284
GPRSRequestedQoS . . . 285
GPSStateInfo . . . 286
gpsTime_s . . . 290
gsmCellInfo . . . 290
GSMRSSIThresh . . . 292
GSMSrvStatusInfo . . . 292
GSMSysInfo . . . 293
gyroAcceptReady_s . . . 296
gyroTempAcceptReady_s . . . 297
HDRECIOThresh . . . 297
HDRIOThresh . . . 298
HDRPersonalityInd . . . 298
HDRPersonalityResp . . . 299
HDRProtSubtypResp . . . 299
HDRRSSIThresh . . . 300
HDRSINRThresh . . . 301
HDRSINRThreshold . . . 301
HDRSSInfo . . . 302
hdrSSInfo . . . 303
HDRSysInfo . . . 304
homeSIDNID . . . 306
hotSwapStatus . . . 307
image_info_t . . . 307
ImageElement . . . 308
ImageIdElement . . . 309

ImageIDEntries . . . 310
ImageList . . . 311
IMSAIndRegisterInfo . . . 311
imsaPdpStatusInfo . . . 312
imsaRatStatusInfo . . . 313
IMSARegistrationStatus . . . 314
imsaRegStatusInfo . . . 314
IMSAServiceStatus . . . 315
IMSASupportedFieldsResp . . . 317
IMSASupportedMsgInfo . . . 318
imsaSvcStatusInfo . . . 318
imsCfgIndRegisterInfo . . . 319
imsRegMgrConfigInfo . . . 320
imsSIPConfigInfo . . . 321
imsSMSConfigInfo . . . 322
imsUserConfigInfo . . . 323
imsVoIPConfigInfo . . . 323
IndFieldsList . . . 326
infoInterFreq . . . 326
IOThresh . . . 327
IPv4Addr . . . 328
IPv4Info . . . 328
IPv6Addr . . . 329
IPV6AddressInfo . . . 330
ipv6AddressInfo . . . 330
IPV6GWAddressInfo . . . 330
IPv6Info . . . 331
IPv6TrafCls . . . 332
LibPackGPRSRequestedQoS . . . 332
LibpackProfile3GPP . . . 333
LibpackProfile3GPP2 . . . 339
LibPackprofile_3GPP . . . 344
LibPackprofile_3GPP2 . . . 350
LibPackQosClassID . . . 355
LibPackTFTIDParams . . . 356
LibPackUMTSQoS . . . 358
LibPackUMTSReqQoSSigInd . . . 360
lineCtrlInfo . . . 361
loc_BdsSV . . . 361
loc_BdsSVInfo . . . 362
loc_CellDb . . . 363
loc_ClkInfo . . . 363
loc_GnssData . . . 364
loc_gpsTime . . . 366
loc_IPv4Info . . . 366
loc_IPv6Info . . . 367
loc_LocApplicationInfo . . . 367
loc_precisionDilution . . . 368
loc_satelliteInfo . . . 369
loc_sensorDataUsage . . . 371
loc_SV . . . 372
loc_SVInfo . . . 373
loc_svUsedforFix . . . 373
loc_urlAddr . . . 374
LocApplicationInfo . . . 375
LocDelAssDataReq . . . 376
LOCEventRegisterReqResp . . . 377
LOCExtPowerStateReqResp . . . 379

LocGetServerReq . . . 379
LocInjectPositionReq . . . 380
LocInjectSensorDataReq . . . 384
LocSetCradleMountReq . . . 385
LOCStartReq . . . 386
LOCStopReq . . . 388
LteCQIParm . . . 388
lteEARFCN . . . 389
lteGsmCellInfo . . . 390
LTEInfo . . . 391
LTEInfoInterfreq . . . 393
LTEInfoIntrafreq . . . 393
LTEInfoNeighboringGSM . . . 396
LTEInfoNeighboringWCDMA . . . 396
LteNasReleaseInfo_s . . . 397
ltePCI . . . 398
lteRsrpinformation . . . 398
LTERSRPThresh . . . 399
LTERSRQThresh . . . 399
LTERSSIThresh . . . 400
LteSccRxInfoResp . . . 400
LTESigRptCfg . . . 401
LTESigRptConfig . . . 401
lteSnrinformation . . . 402
LTESNRThresh . . . 403
LTESNRThreshold . . . 403
LTESSInfo . . . 404
lteSSInfo . . . 405
LTESysInfo . . . 406
lteWcdmaCellInfo . . . 408
messageModeTlv . . . 409
messageWaitingInfoContent . . . 409
minBasedIMSI . . . 410
mitigationDevList . . . 411
MNRInfo . . . 412
ModifyProfileIn . . . 412
ModifyProfileOut . . . 413
msgWaitingInfo . . . 413
namName . . . 414
nas_acqOrderPref . . . 415
nas_ActPilotPNElement . . . 415
nas_AddCDMASysInfo . . . 416
nas_AddSysInfo . . . 416
nas_CallBarringSysInfo . . . 417
nas_callBarStatus . . . 418
nas_CDMAChannel . . . 418
nas_CDMAECIOThresh . . . 419
nas_CDMAInfo . . . 420
nas_CDMARSSIThresh . . . 421
nas_CDMASysInfo . . . 421
nas_CDMASysInfoExt . . . 425
nas_cellParams . . . 425
nas_CommInfo . . . 426
nas_CSGID . . . 428
nas_currentPLMN . . . 429
nas_dataSrvCapabilities . . . 430
nas_DataStatusDetail . . . 431
nas_detailSvcInfo . . . 433

nas_DeviceConfigDetail . . . 434
nas_dirNum . . . 435
nas_DRCParams . . . 435
nas_ecioListElement . . . 436
nas_ECIOThresh . . . 437
nas_errorRateListElement . . . 437
nas_GERANInfo . . . 438
nas_geranInstInfo . . . 440
nas_gsmCellInfo . . . 441
nas_GSMRSSIThresh . . . 442
nas_GSMSrvStatusInfo . . . 442
nas_GSMSysInfo . . . 443
nas_HDRECIOThresh . . . 446
nas_HDRIOThresh . . . 446
nas_HDRRSSIThresh . . . 447
nas_HDRSINRThresh . . . 448
nas_HDRSINRThreshold . . . 448
nas_HDRSysInfo . . . 449
nas_homeSIDNID . . . 451
nas_infoInterFreq . . . 452
nas_IOThresh . . . 453
nas_lteEARFCN . . . 453
nas_lteGsmCellInfo . . . 454
nas_LTEInfo . . . 455
nas_LTEInfoInterfreq . . . 458
nas_LTEInfoIntrafreq . . . 458
nas_LTEInfoNeighboringGSM . . . 460
nas_LTEInfoNeighboringWCDMA . . . 461
nas_ltePCI . . . 462
nas_lteRsrpinformation . . . 462
nas_LTERSRPThresh . . . 463
nas_LTERSRQThresh . . . 463
nas_LTERSSIThresh . . . 464
nas_LTESigRptCfg . . . 464
nas_LTESigRptConfig . . . 465
nas_lteSnrinformation . . . 466
nas_LTESNRThresh . . . 467
nas_LTESNRThreshold . . . 467
nas_LTESysInfo . . . 468
nas_lteWcdmaCellInfo . . . 470
nas_minBasedIMSI . . . 471
nas_MNRInfo . . . 472
nas_namName . . . 473
nas_netSelectionPref . . . 473
nas_NetworkStat1x . . . 474
nas_NetworkStatEVDO . . . 476
nas_nmrCellInfo . . . 478
nas_operatorNameString . . . 479
nas_OperatorPLMNData . . . 479
nas_operatorPLMNList . . . 480
nas_PhyCaAggPcellInfo . . . 481
nas_PhyCaAggScellDlBw . . . 482
nas_PhyCaAggScellIndex . . . 482
nas_PhyCaAggScellIndType . . . 483
nas_PhyCaAggScellInfo . . . 483
nas_PilotSetData . . . 486
nas_PilotSetParams . . . 486
nas_PLMNNetworkName . . . 487

nas_PLMNNetworkNameData . . . 488
nas_protocolSubtypeElement . . . 489
nas_qaQmi3Gpp2TimeZone . . . 491
nas_QmiNas3GppNetworkInfo . . . 491
nas_QmiNas3GppNetworkRAT . . . 493
nas_QmisNasPcsDigit . . . 493
nas_RejectReasonTlv . . . 494
nas_RFInfoTlv . . . 494
nas_roamIndList . . . 495
nas_RSRPThresh . . . 496
nas_rsrqInformation . . . 497
nas_RSRQThresh . . . 497
nas_RSSIThresh . . . 498
nas_rxInfo . . . 499
nas_RxSigInfo . . . 500
nas_rxSignalStrengthListElement . . . 501
nas_SccRxInfo . . . 501
nas_serviceProviderName . . . 502
nas_servSystem . . . 503
nas_sidNid . . . 505
nas_SignalStrengthTlv . . . 505
nas_SLQSSignalStrengthsIndReq . . . 505
nas_SLQSSignalStrengthsInformation . . . 507
nas_SLQSSignalStrengthsTlv . . . 508
nas_SrvStatusInfo . . . 509
nas_sysInfoCommon . . . 509
nas_TDSCDMAECIOThresh . . . 512
nas_TDSCDMARSCPThresh . . . 512
nas_TDSCDMARSSIThresh . . . 513
nas_TDSCDMASINRCONFThresh . . . 513
nas_TDSCDMASINRThresh . . . 514
nas_timeInfo . . . 514
nas_trueIMSI . . . 516
nas_txInfo . . . 517
nas_UMTSInfo . . . 517
nas_UMTSinstInfo . . . 519
nas_umtsLTENbrCell . . . 520
nas_UniversalTime . . . 521
nas_wcdmaCellInfo . . . 522
nas_WCDMAECIOThresh . . . 523
nas_WCDMAInfoLTENeighborCell . . . 523
nas_WCDMARSSIThresh . . . 524
nas_WCDMASysInfo . . . 525
nas_wcdmaUARFCN . . . 528
NASBandPreferenceTlv . . . 529
nasCellLocationInfoResp . . . 529
NASEmergencyModeTlv . . . 530
nasGet3GPP2SubscriptionInfoReq . . . 531
nasGet3GPP2SubscriptionInfoResp . . . 531
nasGetHDRColorCodeResp . . . 532
nasGetLTECphyCa . . . 532
NasGetLTECphyCaInfo . . . 533
nasGetLTECphyCaResp . . . 533
nasGetSigInfoResp . . . 534
nasGetSysInfoResp . . . 535
nasGetTxRxInfoReq . . . 538
nasGetTxRxInfoResp . . . 538
NASGWAcqOrderPrefTlv . . . 539

nasIndicationRegisterReq . . . 539
nasInitNetworkReg . . . 542
NASLTEBandPreferenceTlv . . . 543
NASLteNasReleaseInfoTlv . . . 544
NASModePreferenceTlv . . . 545
NASNetSelPreferenceTlv . . . 545
nasNetworkReject . . . 545
nasNetworkTime . . . 548
nasOperatorNameResp . . . 549
NASOTAMessageTlv . . . 550
NASPhyCaAggPcellInfo . . . 551
NASPhyCaAggScellDlBw . . . 552
NASPhyCaAggScellIndex . . . 552
NASPhyCaAggScellIndType . . . 553
NASPhyCaAggScellInfo . . . 553
nasPLMNNameReq . . . 554
nasPLMNNameResp . . . 555
NASPRLPreferenceTlv . . . 558
NASQmiCbkNasSwiOTAMessageInd . . . 558
NASQmiCbkNasSystemSelPrefInd . . . 559
NASRoamPreferenceTlv . . . 560
NASServDomainPrefTlv . . . 560
NASServingSystemInfo . . . 561
nasSigInfo . . . 562
nasSwiGetChannelLockResp . . . 563
NasSwiIndReg . . . 564
nasSwiSetChannelLockReq . . . 565
nasSysInfo . . . 566
NASTimeInfoTlv . . . 569
nasTimers . . . 569
netSelectionPref . . . 570
NetStats . . . 571
NetworkDebugResp . . . 572
NetworkStat1x . . . 573
NetworkStatEVDO . . . 575
newMTMessageTlv . . . 576
newPwdData . . . 577
nmrCellInfo . . . 577
NSSAudioCtrl . . . 579
NWProfile . . . 579
omaDmConfigTlv . . . 580
omaDmConfigTlvExt . . . 581
omaDmFotaTlv . . . 583
omaDmFotaTlvExt . . . 585
omaDmNotificationsTlv . . . 587
operatorNameString . . . 588
OperatorPLMNData . . . 589
operatorPLMNList . . . 589
pack_dms_ActivateAutomatic_t . . . 590
pack_dms_GetCustFeaturesV2_t . . . 590
pack_dms_ResetToFactoryDefaults_t . . . 591
pack_dms_SetActivationStatusCallback_t . . . 591
pack_dms_SetCrashAction_t . . . 592
pack_dms_SetCustFeature_t . . . 592
pack_dms_SetCustFeaturesV2_t . . . 594
pack_dms_SetEventReport_t . . . 595
pack_dms_SetPower_t . . . 596
pack_dms_SetUSBComp_t . . . 596

pack_dms_SLQSDmsSwiIndicationRegister_t . . . 597
pack_dms_SLQSSwiGetCrashInfo_t . . . 598
pack_dms_SLQSSwiSetDyingGaspCfg_t . . . 598
pack_dms_SLQSSwiSetHostDevInfo_t . . . 599
pack_dms_SLQSSwiSetOSInfo_t . . . 599
pack_dms_UIMChangePIN_t . . . 600
pack_dms_UIMGetControlKeyStatus_t . . . 601
pack_dms_UIMGetICCID_t . . . 601
pack_dms_UIMSetControlKeyProtection_t . . . 601
pack_dms_UIMSetPINProtection_t . . . 602
pack_dms_UIMUnblockControlKey_t . . . 603
pack_dms_UIMUnblockPIN_t . . . 603
pack_dms_UIMVerifyPIN_t . . . 604
pack_fms_GetImagesPreference_t . . . 605
pack_fms_GetStoredImages_t . . . 605
pack_fms_SetImagesPreference_t . . . 605
pack_loc_Delete_Assist_Data_t . . . 606
pack_loc_EventRegister_t . . . 607
pack_loc_SetExtPowerState_t . . . 609
pack_loc_SetOperationMode_t . . . 609
pack_loc_SLQSLOCGetBestAvailPos_t . . . 610
pack_loc_SLQSLOCGetServer_t . . . 611
pack_loc_SLQSLOCInjectPosition_t . . . 611
pack_loc_SLQSLOCInjectSensorData_t . . . 616
pack_loc_SLQSLOCInjectUTCTime_t . . . 618
pack_loc_SLQSLOCSetCradleMountConfig_t . . . 618
pack_loc_Start_t . . . 619
pack_loc_Stop_t . . . 621
pack_nas_InitiateDomainAttach_t . . . 621
pack_nas_SetACCOLC_t . . . 622
pack_nas_SetCDMANetworkParameters_t . . . 622
pack_nas_SetNetworkPreference_t . . . 624
pack_nas_SLQSConfigSigInfo_t . . . 625
pack_nas_SLQSGetPLMNName_t . . . 626
pack_nas_SLQSInitiateNetworkRegistration_t . . . 627
pack_nas_SLQSNasConfigSigInfo2_t . . . 628
pack_nas_SLQSNasGet3GPP2Subscription_t . . . 632
pack_nas_SLQSNasGetTxRxInfo_t . . . 633
pack_nas_SLQSNasIndicationRegisterExt_t . . . 633
pack_nas_SLQSNasSwiIndicationRegister_t . . . 636
pack_nas_SLQSNASSwiSetChannelLock_t . . . 637
pack_nas_SLQSSetBandPreference_t . . . 638
pack_nas_SLQSSetSignalStrengthsCallback_t . . . 640
pack_nas_SLQSSetSysSelectionPref_t . . . 640
pack_nas_SLQSSwiPSDetach_t . . . 646
pack_pds_PDSInjectTimeReference_t . . . 646
pack_pds_ResetPDSData_t . . . 647
pack_pds_SetEventReportCallback_t . . . 647
pack_pds_SetPDSDefaults_t . . . 648
pack_pds_SetPDSState_t . . . 649
pack_pds_SetPortAutomaticTracking_t . . . 649
pack_pds_SetServiceAutomaticTracking_t . . . 650
pack_pds_SetXTRAAutomaticDownload_t . . . 650
pack_pds_SetXTRANetwork_t . . . 651
pack_pds_SLQSGetAGPSConfig_t . . . 651
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t . . . 652
pack_pds_SLQSPDSInjectPositionData_t . . . 652
pack_pds_SLQSSetAGPSConfig_t . . . 654

pack_pds_SLQSSetPositionMethodState_t . . . 655
pack_pds_StartPDSTrackingSessionExt_t . . . 656
pack_qmi_t . . . 657
pack_qos_SLQSQosSwiReadApnExtraParams_t . . . 658
pack_qos_SLQSQosSwiReadDataStats_t . . . 658
pack_qos_SLQSSetQosEventCallback_t . . . 659
pack_rms_SetSMSWake_t . . . 659
pack_sar_SLQSSetRfSarState_t . . . 660
pack_sms_SaveSMS_t . . . 660
pack_sms_SendSMS_t . . . 661
pack_sms_SetNewSMSCallback_t . . . 662
pack_sms_SetSMSCAddress_t . . . 662
pack_sms_SLQSDeleteSMS_t . . . 663
pack_sms_SLQSGetSMS_t . . . 663
pack_sms_SLQSGetSmsBroadcastConfig_t . . . 664
pack_sms_SLQSGetSMSList_t . . . 664
pack_sms_SLQSModifySMSStatus_t . . . 665
pack_sms_SLQSSendAsyncSMS_t . . . 666
pack_sms_SLQSSetIndicationRegister_t . . . 666
pack_sms_SLQSSetSmsBroadcastActivation_t . . . 667
pack_sms_SLQSSetSmsBroadcastConfig_t . . . 667
pack_sms_SLQSSetSmsStorage_t . . . 668
pack_sms_SLQSSmsGetMaxStorageSize_t . . . 668
pack_sms_SLQSSmsSetRoutes_t . . . 669
pack_swiloc_SwiLocSetAutoStart_t . . . 669
pack_swioma_SLQSOMADMCancelSession_t . . . 671
pack_swioma_SLQSOMADMGetSessionInfo_t . . . 671
pack_swioma_SLQSOMADMSendSelection_t . . . 672
pack_swioma_SLQSOMADMSetSettings_t . . . 672
pack_swioma_SLQSOMADMStartSession_t . . . 673
pack_uim_ChangePin_t . . . 674
pack_uim_ReadTransparent_t . . . 675
pack_uim_SetPinProtection_t . . . 676
pack_uim_SLQSUIMAuthenticate_t . . . 677
pack_uim_SLQSUIMDepersonalization_t . . . 678
pack_uim_SLQSUIMEventRegister_t . . . 678
pack_uim_SLQSUIMGetConfiguration_t . . . 678
pack_uim_SLQSUIMGetFileAttributes_t . . . 679
pack_uim_SLQSUIMPowerDown_t . . . 680
pack_uim_SLQSUIMPowerUp_t . . . 680
pack_uim_SLQSUIMRefreshComplete_t . . . 681
pack_uim_SLQSUIMRefreshGetLastEvent_t . . . 681
pack_uim_SLQSUIMRefreshOK_t . . . 682
pack_uim_SLQSUIMRefreshRegister_t . . . 682
pack_uim_SLQSUIMSwitchSlot_t . . . 683
pack_uim_UnblockPin_t . . . 684
pack_uim_VerifyPin_t . . . 685
pack_voice_AnswerUSSD_t . . . 686
pack_voice_OriginateUSSD_t . . . 686
pack_voice_SLQSOriginateUSSD_t . . . 686
pack_voice_SLQSVoiceALSSelectLine_t . . . 687
pack_voice_SLQSVoiceALSSetLineSwitching_t . . . 687
pack_voice_SLQSVoiceAnswerCall_t . . . 688
pack_voice_SLQSVoiceBindSubscription_t . . . 688
pack_voice_SLQSVoiceBurstDTMF_t . . . 689
pack_voice_SLQSVoiceDialCall_t . . . 689
pack_voice_SLQSVoiceEndCall_t . . . 691
pack_voice_SLQSVoiceGetCallBarring_t . . . 691

pack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 692
pack_voice_SLQSVoiceGetCallInfo_t . . . 693
pack_voice_SLQSVoiceGetCallWaiting_t . . . 693
pack_voice_SLQSVoiceGetConfig_t . . . 694
pack_voice_SLQSVoiceIndicationRegister_t . . . 696
pack_voice_SLQSVoiceManageCalls_t . . . 696
pack_voice_SLQSVoiceOrigUSSDNoWait_t . . . 697
pack_voice_SLQSVoiceSendFlash_t . . . 698
pack_voice_SLQSVoiceSetCallBarringPassword_t . . . 698
pack_voice_SLQSVoiceSetConfig_t . . . 699
pack_voice_SLQSVoiceSetPreferredPrivacy_t . . . 701
pack_voice_SLQSVoiceSetSUPSService_t . . . 701
pack_voice_SLQSVoiceStartContDTMF_t . . . 703
pack_voice_SLQSVoiceStopContDTMF_t . . . 704
pack_wds_DHCPv4ClientLeaseChange_t . . . 704
pack_wds_GetDefaultProfile_t . . . 705
pack_wds_GetDefaultProfileNum_t . . . 705
pack_wds_GetDormancyState_t . . . 705
pack_wds_GetLastMobileIPError_t . . . 706
pack_wds_GetMobileIP_t . . . 706
pack_wds_GetMobileIPProfile_t . . . 706
pack_wds_GetPacketStatistics_t . . . 706
pack_wds_GetPacketStatus_t . . . 707
pack_wds_GetSessionDuration_t . . . 707
pack_wds_RMSetTransferStatistics_t . . . 707
pack_wds_SetAutoconnect_t . . . 708
pack_wds_SetDefaultProfile_t . . . 708
pack_wds_SetDefaultProfileNum_t . . . 710
pack_wds_SetMobileIP_t . . . 711
pack_wds_SetMobileIPParameters_t . . . 711
pack_wds_SetMobileIPProfile_t . . . 712
pack_wds_SLQSCreateProfile_t . . . 714
pack_wds_SLQSDeleteProfile_t . . . 715
pack_wds_SLQSGetCurrDataSystemStat_t . . . 715
pack_wds_SLQSGetDataBearerTechnology_t . . . 716
pack_wds_SLQSGetDUNCallInfo_t . . . 716
pack_wds_SLQSGetProfileSettings_t . . . 717
pack_wds_SLQSGetRuntimeSettings_t . . . 718
pack_wds_SLQSModifyProfile_t . . . 719
pack_wds_SLQSSet3GPPConfigItem_t . . . 720
pack_wds_SLQSSetIPFamilyPreference_t . . . 722
pack_wds_SLQSSetWdsEventCallback_t . . . 722
pack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 723
pack_wds_SLQSSSetDHCPv4ClientConfig_t . . . 723
pack_wds_SLQSSSetLoopback_t . . . 724
pack_wds_SLQSStartDataSession_t . . . 724
pack_wds_SLQSStopDataSession_t . . . 725
pack_wds_SLQSWdsSetEventReport_t . . . 726
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 728
PackCreateProfileOut . . . 728
packgetDyingGaspCfg . . . 729
packgetDyingGaspStatistics . . . 729
PCMparams . . . 730
PCSCFFQDNAddress . . . 730
PCSCFFQDNAddressList . . . 731
PCSCFIPv4ServerAddressList . . . 731
PDSPositionData . . . 732
PDSPosMethodStateReq . . . 734

peerNumberInfo . . . 735
personalizationStatus . . . 736
PhyCaAggPcellInfo . . . 737
PhyCaAggScellDlBw . . . 738
PhyCaAggScellIndex . . . 739
PhyCaAggScellIndType . . . 739
PhyCaAggScellInfo . . . 740
PilotSetData . . . 742
PilotSetParams . . . 743
pktErrRate . . . 743
PlmnID . . . 744
PLMNNetworkName . . . 744
PLMNNetworkNameData . . . 745
Port . . . 747
precisionDilution_s . . . 747
PrefImageList . . . 748
prefVoiceSO . . . 748
Profile3GPP . . . 750
Profile3GPP2 . . . 756
ProfileIdentifier . . . 761
protocolSubtypeElement . . . 762
PSDetachReq . . . 763
qaQmi3Gpp2TimeZone . . . 764
qaQmiInterfaceInfo . . . 765
qaQmiServingSystemParam . . . 765
QmiCbkCatEventStatusReportInd . . . 769
QmiCbkLocBestAvailPosInd . . . 769
QmiCbkLocCradleMountInd . . . 775
QmiCbkLocEngineStateInd . . . 776
QmiCbkLocEventTimeSyncInd . . . 776
QmiCbkLocGetServerInd . . . 777
QmiCbkLocInjectPositionInd . . . 778
QmiCbkLocInjectSensorDataInd . . . 779
QmiCbkLocInjectTimeInd . . . 780
QmiCbkLocInjectUTCTimeInd . . . 781
QmiCbkLocPositionReportInd . . . 782
QmiCbkLocSensorStreamingInd . . . 787
QmiCbkLocSetExtPowerConfigInd . . . 788
QmiCbkNasLTECphyCaInfo . . . 789
QmiCbkSwiOmaDmEventStatusReportInd . . . 790
QmiCbkSwiOmaDmEventStatusReportIndExt . . . 790
QmiCbkTmdMitiLvlRptInd . . . 790
QmiCbkWdsStatisticsIndState . . . 791
qmifwinfo_s . . . 792
QmiNas3GppNetworkInfo . . . 792
QmiNasGetRFBandInfoResp . . . 794
QmiNasPerformNetworkScanResp . . . 794
qmiSmsMessageList . . . 794
qmiWDSDataBearerTechnology . . . 795
QmiWdsIpAddressInfo . . . 796
qmiWdsRunTimeSettings . . . 797
QosClassID . . . 800
QosEventInfo . . . 801
QosFlowInfo . . . 803
QosFlowInfoState . . . 804
QosMap . . . 804
RankIndicatorInd . . . 805
readResult . . . 805

readTransparentInfo . . . 806
redirNumInfo . . . 806
registerRefresh . . . 808
remainingRetries . . . 809
remotePartyName . . . 809
remotePartyNum . . . 810
ReqFieldsList . . . 811
RespFieldsList . . . 812
RFBandInfoElements . . . 812
rmTrasnferStaticsReq . . . 813
roamIndList . . . 814
RoamingInfo . . . 815
roamTimer . . . 815
RSRPThresh . . . 816
rsrqInformation . . . 816
RSRQThresh . . . 817
RSSIThresh . . . 817
RXAGCList . . . 818
RXAVCList . . . 819
rxInfo . . . 820
RXPCMIIRFltr . . . 821
RxSigInfo . . . 822
rxSignalStrengthListElement . . . 823
sApnExtraParams . . . 824
satelliteInfo . . . 825
SccRxInfo . . . 828
sensorData . . . 828
sensorData_t . . . 830
sensorDataUsage_s . . . 832
serialNumbersInfo . . . 832
serviceProviderName . . . 834
ServingSystemInfo . . . 834
servSystem . . . 836
sessionInfo . . . 837
sessionInfoExt . . . 838
sessionInfoTlv . . . 838
sessionInfoTlvExt . . . 839
SetAudioPathConfigReq . . . 839
SetAudioProfileReq . . . 841
SetAudioVolTLBConfigReq . . . 842
SetAudioVolTLBConfigResp . . . 843
setCustomSettingV2 . . . 844
setDyingGaspCfg . . . 844
SetIMSSMSConfigReq . . . 845
SetIMSSMSConfigResp . . . 845
SetIMSUserConfigReq . . . 846
SetIMSUserConfigResp . . . 846
SetIMSVoIPConfigReq . . . 847
SetIMSVoIPConfigResp . . . 849
SetM2MAudioAVCFGReq . . . 850
SetM2MAudioLPBKReq . . . 851
SetM2MAudioProfileReq . . . 851
SetM2MAudioVolumeReq . . . 852
SetM2MAVMuteReq . . . 853
SetM2MSpkrGainReq . . . 854
setPINProtection . . . 854
SetRegMgrConfigReq . . . 855
SetRegMgrConfigResp . . . 856

setSignalStrengthInfo . . . 856
SetSIPConfigReq . . . 861
SetSIPConfigResp . . . 862
sGetDeviceSeriesResult . . . 863
sidNid . . . 863
sigInfo . . . 864
signalInfo . . . 865
SignalStrengthDataType . . . 866
slot_t . . . 866
slotInf . . . 867
slotInfo . . . 869
slots_t . . . 870
slqsautoconnect . . . 870
SLQSDeleteProfileParams . . . 871
slqsfwinfo_s . . . 872
SlqsNas3GppNetworkInfo . . . 873
SlqsNasPcsDigit . . . 874
slqssendasyncsmsparams_s . . . 875
slqssendsmsparams_s . . . 877
slqsSessionStateInfo . . . 878
slqsSignalStrengthInfo . . . 879
SLQSSignalStrengthsIndReq . . . 882
SLQSSignalStrengthsInformation . . . 884
slqsWdsEventInfo . . . 885
sms_BroadcastConfig . . . 887
sms_CDMABroadcastConfig . . . 887
sms_getIndicationReg . . . 888
sms_getMsgWaitingInfo . . . 889
sms_getTransLayerInfo . . . 889
sms_getTransNWRegInfo . . . 890
sms_maxStorageSizeReq . . . 891
sms_maxStorageSizeResp . . . 891
sms_messageWaitingInfoContent . . . 892
sms_msgProtocolResp . . . 893
sms_qaQmi3GPP2BroadcastCfgInfo . . . 893
sms_qaQmi3GPPBroadcastCfgInfo . . . 894
sms_routeEntry . . . 894
sms_sendAsyncsmsParams . . . 895
sms_setIndicationReg . . . 898
sms_setRoutesReq . . . 898
sms_transLayerInfo . . . 899
SMSAsyncRawSend_s . . . 900
SMSCAddress . . . 901
sMSCAddress . . . 902
sMSCAddressTlv . . . 902
sMSEtwsMessage . . . 903
SMSEtwsMessage . . . 903
sMSEtwsMessageTlv . . . 904
SMSEtwsPlmn . . . 904
sMSEtwsPlmn . . . 905
SMSEventInfo_s . . . 905
smsMaxStorageSizeReq . . . 907
smsMaxStorageSizeResp . . . 907
SMSMemoryInfo . . . 908
SMSMessageMode . . . 908
sMSMessageMode . . . 909
smsMsgprotocolResp . . . 909
SMSMTMessage . . . 910

sMSMTMessage . . . 910
sMSOnIMS . . . 911
SMSOnIMS . . . 911
sMSOnIMSTlv . . . 911
smsRouteEntry . . . 912
smsSetRoutesReq . . . 913
SMSTransferRouteMTMessage . . . 914
sMSTransferRouteMTMessage . . . 914
sQosFlowStat . . . 915
sQosStat . . . 916
SrvStatusInfo . . . 917
ssdatasession_params . . . 918
SupportedMsgList . . . 921
SUPSInfo . . . 921
SV . . . 922
SVInfo . . . 923
svUsedforFix_s . . . 924
SWI_STRUCT_CarrierImage . . . 924
SwiLocGetAutoStartResp . . . 925
SwiLocSetAutoStartReq . . . 927
swiModemStatusResp . . . 929
SwiOTAMsg_s . . . 929
swiPDPRuntimeSettingsReq . . . 930
swiPDPRuntimeSettingsResp . . . 931
swiQosFilter . . . 933
swiQosFlow . . . 936
swiQosGranted . . . 939
swiQosIds . . . 939
swiQosModifyReq . . . 940
swiQosReq . . . 940
swiRMTrasnferStaticsReq . . . 941
sysInfoCommon . . . 942
t_gpsTime . . . 944
t_sensor . . . 944
t_Sv . . . 945
TDSCDMAECIOThresh . . . 945
TDSCDMARSCPThresh . . . 945
TDSCDMARSSIThresh . . . 946
tdscdmaSigInfoExt . . . 946
TDSCDMASigInfoExt . . . 947
TDSCDMASINRCONFThresh . . . 948
TDSCDMASINRThresh . . . 948
tempData_t . . . 949
tempratureData . . . 950
TFTIDParams . . . 951
timeInfo . . . 953
TmdDeRegNotMitigationLvlReq . . . 955
TmdGetMitigationDevListResp . . . 955
TmdGetMitigationLvlReq . . . 956
TmdGetMitigationLvlResp . . . 956
TmdMitigationLvlIndReq . . . 957
TmdRegNotMitigationLvlReq . . . 957
tokenBucket . . . 958
Tos . . . 958
transferRouteMessageTlv . . . 959
TransferStatInd . . . 960
transferStatInd . . . 960
TransferStatsDataType . . . 961

TrStatInd . . . 961
trueIMSI . . . 961
TXAGCList . . . 962
txInfo . . . 963
TXPCMIIRFltr . . . 964
uim_appStatus . . . 965
uim_authenticateResult . . . 968
uim_authenticationData . . . 969
uim_cardResult . . . 970
uim_cardStatus . . . 971
uim_changeUIMPIN . . . 972
uim_depersonalizationInformation . . . 973
uim_encryptedPIN1 . . . 974
uim_fileAttributes . . . 975
uim_fileInfo . . . 978
uim_hotSwapStatus . . . 979
uim_personalizationStatus . . . 979
uim_readResult . . . 980
uim_readTransparentInfo . . . 981
uim_refreshevent . . . 981
uim_registerRefresh . . . 983
uim_remainingRetries . . . 984
uim_sessionInformation . . . 985
uim_setPINProtection . . . 985
uim_slotInfo . . . 986
uim_UIMSessionInformation . . . 988
uim_unblockUIMPIN . . . 989
uim_verifyUIMPIN . . . 990
UIMAuthenticateReq . . . 990
UIMAuthenticateResp . . . 991
UIMChangePinReq . . . 992
UIMDepersonalizationReq . . . 993
UIMDepersonalizationResp . . . 993
UIMEventRegisterReqResp . . . 993
UIMGetCardStatusResp . . . 994
UIMGetConfigurationReq . . . 994
UIMGetConfigurationResp . . . 995
UIMGetFileAttributesReq . . . 996
UIMGetFileAttributesResp . . . 996
UIMGetSlotsStatusResp . . . 997
UIMPinResp . . . 998
UIMPowerDownReq . . . 998
UIMPowerUpReq . . . 999
UIMReadTransparentReq . . . 999
UIMReadTransparentResp . . . 1000
UIMRefreshCompleteReq . . . 1001
UIMRefreshEvent . . . 1002
UIMRefreshGetLastEventReq . . . 1003
UIMRefreshGetLastEventResp . . . 1004
UIMRefreshOKReq . . . 1004
UIMRefreshRegisterReq . . . 1005
UIMSessionInformation . . . 1005
UIMSetPinProtectionReq . . . 1006
UIMSlotsStatus . . . 1007
UIMSlotStatus . . . 1007
UIMSlotStatusChangeInfo . . . 1009
UIMStatusChangeInfo . . . 1009
UIMSwitchSlotReq . . . 1009

UIMUnblockPinReq . . . 1010
UIMVerifyPinReq . . . 1011
UMTSInfo . . . 1012
UMTSinstInfo . . . 1014
umtsLTENbrCell . . . 1014
UMTSMinQoS . . . 1015
UMTSQoS . . . 1018
UMTSReqQoSSigInd . . . 1020
unblockUIMPIN . . . 1021
UniversalTime . . . 1022
unpack_dms_GetActivationState_t . . . 1023
unpack_dms_GetBandCapability_t . . . 1024
unpack_dms_GetCrashAction_t . . . 1026
unpack_dms_GetCustFeature_t . . . 1026
unpack_dms_GetCustFeaturesV2_t . . . 1028
unpack_dms_GetDeviceCap_t . . . 1029
unpack_dms_GetDeviceCapabilities_t . . . 1030
unpack_dms_GetDeviceHardwareRev_t . . . 1031
unpack_dms_GetDeviceMfr_t . . . 1032
unpack_dms_GetDeviceSerialNumbers_t . . . 1033
unpack_dms_GetFirmwareInfo_t . . . 1034
unpack_dms_GetFirmwareRevision_t . . . 1036
unpack_dms_GetFirmwareRevisions_t . . . 1036
unpack_dms_GetFSN_t . . . 1037
unpack_dms_GetHardwareRevision_t . . . 1038
unpack_dms_GetIMSI_t . . . 1038
unpack_dms_GetManufacturer_t . . . 1039
unpack_dms_GetModelID_t . . . 1039
unpack_dms_GetNetworkTime_t . . . 1040
unpack_dms_GetOfflineReason_t . . . 1041
unpack_dms_GetPower_t . . . 1041
unpack_dms_GetPRLVersion_t . . . 1042
unpack_dms_GetSerialNumbers_t . . . 1043
unpack_dms_GetUSBComp_t . . . 1044
unpack_dms_GetVoiceNumber_t . . . 1045
unpack_dms_ResetToFactoryDefaults_t . . . 1046
unpack_dms_SetActivationStatusCallback_t . . . 1047
unpack_dms_SetCrashAction_t . . . 1047
unpack_dms_SetCustFeature_t . . . 1047
unpack_dms_SetCustFeaturesV2_t . . . 1048
unpack_dms_SetEventReport_ind_t . . . 1048
unpack_dms_SetEventReport_t . . . 1049
unpack_dms_SetFirmwarePreference_t . . . 1049
unpack_dms_SetPower_t . . . 1050
unpack_dms_SetUSBComp_t . . . 1050
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t . . . 1050
unpack_dms_SLQSDmsSwiGetResetInfo_t . . . 1051
unpack_dms_SLQSDmsSwiIndicationRegister_t . . . 1052
unpack_dms_SLQSGetBandCapability_t . . . 1053
unpack_dms_SLQSGetERIFile_t . . . 1056
unpack_dms_SLQSSwiClearDyingGaspStatistics_t . . . 1056
unpack_dms_SLQSSwiGetCrashInfo_t . . . 1057
unpack_dms_SLQSSwiGetDyingGaspCfg_t . . . 1057
unpack_dms_SLQSSwiGetDyingGaspStatistics_t . . . 1058
unpack_dms_SLQSSwiGetFirmwareCurr_t . . . 1058
unpack_dms_SLQSSwiGetFwUpdateStatus_t . . . 1059
unpack_dms_SLQSSwiGetHostDevInfo_t . . . 1061
unpack_dms_SLQSSwiGetOSInfo_t . . . 1062

unpack_dms_SLQSSwiGetSerialNoExt_t . . . 1062
unpack_dms_SLQSSwiSetDyingGaspCfg_t . . . 1063
unpack_dms_SLQSSwiSetHostDevInfo_t . . . 1063
unpack_dms_SLQSSwiSetOSInfo_t . . . 1063
unpack_dms_SLQSUIMGetState_t . . . 1064
unpack_dms_UIMGetControlKeyStatus_t . . . 1064
unpack_dms_UIMGetICCID_t . . . 1065
unpack_dms_UIMGetPINStatus_t . . . 1066
unpack_dms_UIMSetControlKeyProtection_t . . . 1067
unpack_dms_UIMSetPINProtection_t . . . 1068
unpack_dms_UIMUnblockControlKey_t . . . 1069
unpack_fms_GetImagesPreference_t . . . 1069
unpack_fms_GetStoredImages_t . . . 1070
unpack_fms_SetImagesPreference_t . . . 1070
unpack_loc_BestAvailPos_Ind_t . . . 1071
unpack_loc_Delete_Assist_Data_t . . . 1076
unpack_loc_DeleteAssistData_Ind_t . . . 1077
unpack_loc_EngineState_Ind_t . . . 1078
unpack_loc_EventNMEA_Ind_t . . . 1078
unpack_loc_EventRegister_t . . . 1079
unpack_loc_GetServer_Ind_t . . . 1079
unpack_loc_GnssSvInfo_Ind_t . . . 1081
unpack_loc_PositionRpt_Ind_t . . . 1081
unpack_loc_SetExtPowerConfig_Ind_t . . . 1088
unpack_loc_SetExtPowerState_t . . . 1088
unpack_loc_SetOperationMode_Ind_t . . . 1089
unpack_loc_SetOperationMode_t . . . 1089
unpack_loc_SLQSLOCGetBestAvailPos_t . . . 1090
unpack_loc_Start_t . . . 1090
unpack_loc_Stop_t . . . 1090
unpack_nas_GetCDMANetworkParameters_t . . . 1091
unpack_nas_GetHomeNetwork3GPP2_t . . . 1093
unpack_nas_GetHomeNetwork_t . . . 1095
unpack_nas_GetNetworkPreference_t . . . 1096
unpack_nas_GetRFInfo_t . . . 1097
unpack_nas_GetServingNetwork_t . . . 1098
unpack_nas_GetServingNetworkCapabilities_t . . . 1100
unpack_nas_GetSignalStrengths_t . . . 1101
unpack_nas_PerformNetworkScan_t . . . 1102
unpack_nas_SetDataCapabilitiesCallback_ind_t . . . 1103
unpack_nas_SetEventReportInd_t . . . 1103
unpack_nas_SetNasLTECphyCaIndCallback_ind_t . . . 1104
unpack_nas_SetNetworkPreference_t . . . 1105
unpack_nas_SetRoamingIndicatorCallback_ind_t . . . 1105
unpack_nas_SetServingSystemCallback_ind_t . . . 1106
unpack_nas_SLQSGetErrorRate_t . . . 1106
unpack_nas_SlqsGetLTECphyCAInfo_t . . . 1107
unpack_nas_SLQSGetNetworkTime_t . . . 1108
unpack_nas_SLQSGetOperatorNameData_t . . . 1108
unpack_nas_SLQSGetPLMNName_t . . . 1109
unpack_nas_SLQSGetServingSystem_t . . . 1110
unpack_nas_SLQSGetSignalStrength_t . . . 1114
unpack_nas_SLQSGetSysInfo_t . . . 1116
unpack_nas_SLQSGetSysSelectionPref_t . . . 1119
unpack_nas_SLQSNasGet3GPP2Subscription_t . . . 1123
unpack_nas_SLQSNasGetCellLocationInfo_t . . . 1124
unpack_nas_SLQSNasGetHDRColorCode_t . . . 1126
unpack_nas_SLQSNasGetSigInfo_t . . . 1126

unpack_nas_SLQSNasGetTxRxInfo_t . . . 1127
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t . . . 1128
unpack_nas_SLQSNasSigInfoCallback_ind_t . . . 1128
unpack_nas_SLQSNASSwiGetChannelLock_t . . . 1130
unpack_nas_SLQSNasSwiModemStatus_t . . . 1130
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t . . . 1131
unpack_nas_SLQSNasTimerCallback_ind_t . . . 1131
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t . . . 1132
unpack_nas_SLQSSwiGetHDRPersonality_t . . . 1132
unpack_nas_SLQSSwiGetHDRProtSubtype_t . . . 1133
unpack_nas_SLQSSwiGetHRPDStats_t . . . 1134
unpack_nas_SLQSSwiGetLteCQI_t . . . 1134
unpack_nas_SLQSSwiGetLteSccRxInfo_t . . . 1135
unpack_nas_SLQSSwiNetworkDebug_t . . . 1135
unpack_nas_SLQSSysInfoCallback_ind_t . . . 1136
unpack_omaDmConfigTlv_t . . . 1139
unpack_omaDmFotaTlv_t . . . 1140
unpack_omaDmNotificationsTlv_t . . . 1142
unpack_pds_GetPDSDefaults_t . . . 1143
unpack_pds_GetPDSState_t . . . 1144
unpack_pds_GetPortAutomaticTracking_t . . . 1145
unpack_pds_GetServiceAutomaticTracking_t . . . 1145
unpack_pds_GetXTRAAutomaticDownload_t . . . 1145
unpack_pds_GetXTRANetwork_t . . . 1146
unpack_pds_GetXTRAValidity_t . . . 1146
unpack_pds_SetEventReport_Ind_t . . . 1147
unpack_pds_SetPdsState_Ind_t . . . 1149
unpack_pds_SLQSGetAGPSConfig_t . . . 1149
unpack_pds_SLQSGetGPSStateInfo_t . . . 1150
unpack_qmi_t . . . 1154
unpack_qos_dataRate_t . . . 1155
unpack_qos_IPv4Addr_t . . . 1155
unpack_qos_IPv6Addr_t . . . 1156
unpack_qos_IPv6TrafCls_t . . . 1156
unpack_qos_pktErrRate_t . . . 1157
unpack_qos_Port_t . . . 1157
unpack_qos_QosFlowInfo_t . . . 1157
unpack_qos_QosFlowInfoState_t . . . 1159
unpack_qos_SLQSQosGetNetworkStatus_t . . . 1160
unpack_qos_SLQSQosSwiReadApnExtraParams_t . . . 1160
unpack_qos_SLQSQosSwiReadDataStats_t . . . 1161
unpack_qos_SLQSSetQosEventCallback_ind_t . . . 1163
unpack_qos_SLQSSetQosNWStatusCallback_ind_t . . . 1163
unpack_qos_SLQSSetQosPriEventCallback_ind_t . . . 1164
unpack_qos_SLQSSetQosStatusCallback_ind_t . . . 1164
unpack_qos_swiQosFilter_t . . . 1165
unpack_qos_swiQosFlow_t . . . 1169
unpack_qos_tokenBucket_t . . . 1173
unpack_qos_Tos_t . . . 1173
unpack_QosFlowStat_t . . . 1174
unpack_rms_GetSMSWake_t . . . 1175
unpack_RMTransferStatistics_ind_t . . . 1175
unpack_sar_SLQSGetRfSarState_t . . . 1176
unpack_sms_GetSMSCAddress_t . . . 1177
unpack_sms_SaveSMS_t . . . 1178
unpack_sms_SendSMS_t . . . 1178
unpack_sms_SetNewSMSCallback_ind_t . . . 1178
unpack_sms_SetNewSMSCallback_t . . . 1180

unpack_sms_SLQSDeleteSMS_t . . . 1180
unpack_sms_SLQSGetIndicationRegister_t . . . 1180
unpack_sms_SLQSGetMessageWaiting_t . . . 1180
unpack_sms_SLQSGetSMS_t . . . 1181
unpack_sms_SLQSGetSmsBroadcastConfig_t . . . 1181
unpack_sms_SLQSGetSMSList_t . . . 1182
unpack_sms_SLQSGetTransLayerInfo_t . . . 1182
unpack_sms_SLQSGetTransNWRegInfo_t . . . 1183
unpack_sms_SLQSModifySMSStatus_t . . . 1183
unpack_sms_SLQSSmsGetMaxStorageSize_t . . . 1183
unpack_sms_SLQSSmsGetMessageProtocol_t . . . 1184
unpack_sms_SLQSSwiGetSMSStorage_t . . . 1184
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t . . . 1185
unpack_swiloc_SwiLocGetAutoStart_t . . . 1185
unpack_swioma_SLQSOMADMAlertCallback_ind_t . . . 1187
unpack_swioma_SLQSOMADMGetSessionInfo_t . . . 1188
unpack_swioma_SLQSOMADMGetSettings_t . . . 1190
unpack_swioma_SLQSOMADMStartSession_t . . . 1192
unpack_uim_ChangePin_t . . . 1192
unpack_uim_GetCardStatus_t . . . 1193
unpack_uim_ReadTransparent_t . . . 1194
unpack_uim_SetPinProtection_t . . . 1195
unpack_uim_SetUimSlotStatusChangeCallback_ind_t . . . 1195
unpack_uim_SLQSUIMAuthenticate_t . . . 1196
unpack_uim_SLQSUIMDepersonalization_t . . . 1197
unpack_uim_SLQSUIMEventRegister_t . . . 1197
unpack_uim_SLQSUIMGetConfiguration_t . . . 1197
unpack_uim_SLQSUIMGetFileAttributes_t . . . 1198
unpack_uim_SLQSUIMGetSlotsStatus_t . . . 1199
unpack_uim_SLQSUIMRefreshGetLastEvent_t . . . 1199
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t . . . 1200
unpack_uim_UnblockPin_t . . . 1200
unpack_uim_VerifyPin_t . . . 1201
unpack_voice_SLQSOriginateUSSD_t . . . 1201
unpack_voice_SLQSVoiceAnswerCall_t . . . 1202
unpack_voice_SLQSVoiceBurstDTMF_t . . . 1202
unpack_voice_SLQSVoiceDialCall_t . . . 1203
unpack_voice_SLQSVoiceEndCall_t . . . 1204
unpack_voice_SLQSVoiceGetAllCallInfo_t . . . 1204
unpack_voice_SLQSVoiceGetCallBarring_t . . . 1207
unpack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 1208
unpack_voice_SLQSVoiceGetCallInfo_t . . . 1209
unpack_voice_SLQSVoiceGetCallWaiting_t . . . 1212
unpack_voice_SLQSVoiceGetCLIP_t . . . 1214
unpack_voice_SLQSVoiceGetCLIR_t . . . 1215
unpack_voice_SLQSVoiceGetCNAP_t . . . 1216
unpack_voice_SLQSVoiceGetCOLP_t . . . 1217
unpack_voice_SLQSVoiceGetCOLR_t . . . 1219
unpack_voice_SLQSVoiceGetConfig_t . . . 1220
unpack_voice_SLQSVoiceManageCalls_t . . . 1222
unpack_voice_SLQSVoiceSendFlash_t . . . 1222
unpack_voice_SLQSVoiceSetCallBarringPassword_t . . . 1222
unpack_voice_SLQSVoiceSetConfig_t . . . 1224
unpack_voice_SLQSVoiceSetSUPSService_t . . . 1225
unpack_voice_SLQSVoiceStartContDTMF_t . . . 1226
unpack_voice_SLQSVoiceStopContDTMF_t . . . 1227
unpack_wds_DHCPv4ClientLease_ind_t . . . 1227
unpack_wds_GetAutoconnect_t . . . 1228

unpack_wds_GetByteTotals_t . . . 1228
unpack_wds_GetConnectionRate_t . . . 1229
unpack_wds_GetDataBearerTechnology_t . . . 1229
unpack_wds_GetDefaultProfile_t . . . 1230
unpack_wds_GetDefaultProfileNum_t . . . 1232
unpack_wds_GetDormancyState_t . . . 1233
unpack_wds_GetLastMobileIPError_t . . . 1233
unpack_wds_GetMobileIP_t . . . 1234
unpack_wds_GetMobileIPProfile_t . . . 1234
unpack_wds_GetPacketStatistics_t . . . 1236
unpack_wds_GetPacketStatus_t . . . 1237
unpack_wds_GetSessionDuration_t . . . 1239
unpack_wds_GetSessionState_t . . . 1240
unpack_wds_RMSetTransferStatistics_t . . . 1240
unpack_wds_SetMobileIPProfile_t . . . 1240
unpack_wds_SLQSCreateProfile_t . . . 1240
unpack_wds_SLQSDeleteProfile_t . . . 1241
unpack_wds_SLQSGet3GPPConfigItem_t . . . 1241
unpack_wds_SLQSGetCurrDataSystemStat_t . . . 1243
unpack_wds_SLQSGetCurrentChannelRate_t . . . 1243
unpack_wds_SLQSGetDataBearerTechnology_t . . . 1244
unpack_wds_SLQSGetDUNCallInfo_t . . . 1245
unpack_wds_SLQSGetProfileSettings_t . . . 1248
unpack_wds_SLQSGetRuntimeSettings_t . . . 1248
unpack_wds_SLQSModifyProfile_t . . . 1250
unpack_wds_SLQSSetIPFamilyPreference_t . . . 1251
unpack_wds_SLQSSetPacketSrvStatusCallback_t . . . 1251
unpack_wds_SLQSSetWdsEventCallback_ind_t . . . 1252
unpack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 1254
unpack_wds_SLQSSGetLoopback_t . . . 1255
unpack_wds_SLQSStartDataSession_t . . . 1255
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 1256
UnPackGetProfileSettingOut . . . 1258
unpackWdsProfileParam . . . 1258
urlAddr . . . 1259
USBCompConfig . . . 1259
USBCompParams . . . 1260
USSDNoWaitIndicationInfo . . . 1262
USSDRespFNetwork . . . 1263
USSInfo . . . 1263
USSResp . . . 1264
UUSInfo . . . 1265
verifyUIMPIN . . . 1266
voice_airTimer . . . 1266
voice_allCallsAlphaIDInfo . . . 1267
voice_allCallsDiagInfo . . . 1267
voice_allCallsUUSInfo . . . 1268
voice_alphaIDInfo . . . 1268
voice_arrAlertingPattern . . . 1269
voice_arrAlertingType . . . 1270
voice_arrAlphaID . . . 1271
voice_arrCalledPartyNum . . . 1271
voice_arrCallEndReason . . . 1272
voice_arrCallInfo . . . 1272
voice_arrConnectPartyNum . . . 1273
voice_arrDiagInfo . . . 1274
voice_arrRedirPartyNum . . . 1274
voice_arrRemotePartyName . . . 1275

voice_arrRemotePartyNum . . . 1275
voice_arrSvcOption . . . 1276
voice_arrUUSInfo . . . 1276
voice_burstDTMFInfo . . . 1277
voice_calledPartySubAdd . . . 1278
voice_callFwdTypeAndPlan . . . 1278
voice_callFWExtInfo . . . 1279
voice_callFWInfo . . . 1281
voice_callInfo . . . 1282
voice_ccSUPSType . . . 1284
voice_CLIPResp . . . 1284
voice_CLIRResp . . . 1285
voice_CNAPResp . . . 1286
voice_COLPResp . . . 1287
voice_COLRResp . . . 1287
voice_connectNumInfo . . . 1288
voice_CUGInfo . . . 1290
voice_curAMRConfig . . . 1290
voice_diagInfo . . . 1291
voice_DTMFLengths . . . 1292
voice_getAllCallInformation . . . 1292
voice_getAllCallRmtPtyName . . . 1293
voice_getAllCallRmtPtyNum . . . 1293
voice_getCallFWExtInfo . . . 1294
voice_getCallFWInfo . . . 1295
voice_peerNumberInfo . . . 1295
voice_prefVoiceSO . . . 1297
voice_remotePartyName . . . 1299
voice_remotePartyNum . . . 1300
voice_roamTimer . . . 1300
voice_USSInfo . . . 1301
voice_UUSInfo . . . 1302
voiceALSSelectLineInfo . . . 1303
voiceALSSetLineSwitchInfo . . . 1303
voiceAnswerCall . . . 1304
voiceBindSubscriptionInfo . . . 1304
voiceBurstDTMFInfo . . . 1304
voiceCallInfoReq . . . 1305
voiceCallInfoResp . . . 1305
voiceCallRequestParams . . . 1308
voiceCallResponseParams . . . 1310
voiceContDTMFinfo . . . 1311
voiceDTMFEventInfo . . . 1312
voiceFlashInfo . . . 1312
voiceGetAllCallInfo . . . 1313
voiceGetCallBarringReq . . . 1316
voiceGetCallBarringResp . . . 1316
voiceGetCallFWReq . . . 1318
voiceGetCallFWResp . . . 1318
voiceGetCallWaitInfo . . . 1320
voiceGetCLIPResp . . . 1321
voiceGetCLIRResp . . . 1322
voiceGetCNAPResp . . . 1324
voiceGetCOLPResp . . . 1325
voiceGetCOLRResp . . . 1326
voiceGetConfigReq . . . 1327
voiceGetConfigResp . . . 1329
voiceIndicationRegisterInfo . . . 1331

voiceInfoRec . . . 1332
voiceManageCallsReq . . . 1334
voiceManageCallsResp . . . 1335
voiceOrigUSSDNoWaitInfo . . . 1336
voiceOTASPStatusInfo . . . 1336
voicePrivacyInfo . . . 1337
voiceSetAllCallStatusCbkInfo . . . 1338
voiceSetCallBarringPwdInfo . . . 1340
voiceSetCallBarringPwdResp . . . 1341
voiceSetConfigReq . . . 1342
voiceSetConfigResp . . . 1344
voiceSetPrefPrivacy . . . 1345
voiceSetSUPSServiceReq . . . 1346
voiceSetSUPSServiceResp . . . 1348
voiceStopContDTMFinfo . . . 1349
voiceSUPSInfo . . . 1350
voiceSUPSNotification . . . 1352
wcdmaCellInfo . . . 1354
WCDMAECIOThresh . . . 1355
WCDMAInfoLTENeighborCell . . . 1355
wcdmaLongMsgDecodingParams . . . 1356
wcdmaMsgDecodingParams . . . 1358
wcdmaMsgEncodingParams . . . 1359
WCDMARSSIThresh . . . 1360
WCDMASysInfo . . . 1361
wcdmaUARFCN . . . 1364
wds_currNetworkInfo . . . 1365
wds_DataULongLongTlv . . . 1366
wds_DataULongTlv . . . 1367
wds_DHCPLeaseOptTlv . . . 1367
wds_DHCPLeaseStateTlv . . . 1368
wds_DHCPOpt . . . 1368
wds_DHCPProfileIdTlv . . . 1369
wds_DHCPv4HWConfig . . . 1369
wds_DHCPv4Option . . . 1370
wds_DHCPv4OptionList . . . 1371
wds_DHCPv4ProfileId . . . 1371
wds_Domain . . . 1372
wds_DomainNameList . . . 1372
wds_GPRSQoS . . . 1373
wds_IPv4AdTlv . . . 1373
wds_IPV6AddressInfo . . . 1374
wds_IPV6GWAddressInfo . . . 1374
wds_PCSCFFQDNAddress . . . 1375
wds_PCSCFFQDNAddressList . . . 1375
wds_PCSCFIPv4ServerAddressList . . . 1376
wds_ProfileIdentifier . . . 1376
wds_profileInfo . . . 1377
wds_TrStatInd . . . 1377
wds_UMTSMinQoS . . . 1378
WdsByteTotals . . . 1380
WdsByteTotalsElmnts . . . 1381
WdsClientLeaseChange . . . 1382
WdsConnectionRate . . . 1382
WdsConnectionRateElmnts . . . 1383
WdsDHCPv4ClientLeaseInd . . . 1384
WdsDHCPv4Config . . . 1385
wdsDhcpv4HwConfig . . . 1385

WdsDHCPv4HWConfig . . . 1386
WdsDHCPv4Option . . . 1386
wdsDhcpv4Option . . . 1387
wdsDhcpv4OptionList . . . 1388
WdsDHCPv4OptionList . . . 1388
wdsDhcpv4ProfileId . . . 1388
WdsDHCPv4ProfileId . . . 1389
WDSGetLoopbackData . . . 1389
WdsIpAddressInfoReq . . . 1390
WdsPktStatisticsElmnts . . . 1391
WdsPktStatisticsReq . . . 1393
WdsPktStatisticsResp . . . 1393
WdsProfileParam . . . 1394
WdsRunTimeSettings . . . 1394
wdsSetEventReportReq . . . 1395
WDSSetLoopbackData . . . 1397
WDSSWICurrentChannelRates . . . 1397

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 . . . 1399
common.h . . . 1399
dms.h . . . 1404
fms.h . . . 1456
loc.h . . . 1459
nas.h . . . 1476
pds.h . . . 1520
qaCbkCatEventReportInd.h . . . 1541
qaCbkSwiOmaDmEventReportInd.h . . . 1543
qaGobiApiAudio.h
Audio Service API function prototypes . . . 1544
qaGobiApiCat.h
Card Application Toolkit API function headers . . . 1547
qaGobiApiCbk.h
Callback Service API function prototypes . . . 1549
qaGobiApiDcs.h
Device Connectivity Service API function prototypes . . . 1636
qaGobiApiDms.h
Device Management Service API function prototypes . . . 1644
qaGobiApiFms.h
Firmware Management Service API function prototypes . . . 1682
qaGobiApiIms.h
IMS Service API function prototypes . . . 1701
qaGobiApiImsa.h
IMSA Service API function prototypes . . . 1707
qaGobiApiLoc.h
Location API function prototypes . . . 1710
qaGobiApiNas.h
Network Access Service API function prototypes . . . 1718
qaGobiApiOmadm.h
Open Mobile Alliance Device Management Service API function prototypes . . . 1765
qaGobiApiPds.h
Position Determination Service API function prototypes . . . 1768
qaGobiApiQos.h
Quality of Service API function prototypes . . . 1783

qaGobiApiRms.h
Remote Management Service API function prototypes . . . 1789
qaGobiApiSar.h
Specific Absorption Rate API function prototypes . . . 1791
qaGobiApiSms.h
Short Message Service API function prototypes . . . 1793
qaGobiApiSwi.h
SWI API function prototypes . . . 1815
qaGobiApiSwiAudio.h
M2M Audio Service API function prototypes . . . 1816
qaGobiApiSwiAvms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI AVMS QMI Service revision 1.14 . . . 1822
qaGobiApiSwiOmadms.h
SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6 . . . 1834
qaGobiApiTableBandClasses.h
Network Access Service API Band Classes table . . . 1845
qaGobiApiTableCallControlReturnReasons.h
Call Control Return Reasons table . . . 1848
qaGobiApiTableCallEndReasons.h
Wireless Data Service Call End Reasons . . . 1849
qaGobiApiTableCarrierCodes.h
Carrier Codes table . . . 1865
qaGobiApiTableCodingScheme.h
Data Coding Scheme . . . 1867
qaGobiApiTableGpsCapabilityCodes.h
Position Determination Service API GPS Capability Codes . . . 1870
qaGobiApiTablePowerModes.h
Device Management Service API Power Modes table . . . 1871
qaGobiApiTableRadioInterfaces.h
Network Access Service API Radio Interfaces table . . . 1871
qaGobiApiTableRegionCodes.h
Region Codes table . . . 1872
qaGobiApiTableServiceOptions.h
Voice Service Options . . . 1872
qaGobiApiTableSupServiceInfoClasses.h
Voice Supplementary Service Information Classes . . . 1875
qaGobiApiTableSwiAudio.h
Swi Audio related tables . . . 1875
qaGobiApiTableSwiOMADMSessionStatus.h
Session Status table . . . 1876
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h
Update Complete Status table . . . 1877
qaGobiApiTableVoiceCallEndReasons.h
Voice Service Call and supplementary services end reasons . . . 1878
qaGobiApiTmd.h
Thermal Mitigation Device API function prototypes . . . 1885
qaGobiApiUim.h
Uim Service API function prototypes . . . 1887
qaGobiApiVoice.h
Voice Service API function prototypes . . . 1899
qaGobiApiWds.h
Wireless Data Service API function prototypes . . . 1920
qaNasGetRFBandInfo.h . . . 1963
qaNasPerformNetworkScan.h . . . 1964
qmerrno.h . . . 1965
qos.h . . . 1972

rms.h 1980
sar.h 1982
sms.h 1984
SwiDataTypes.h
SWI data types 2006
swiloc.h 2007
swioma.h 2009
SWIWWANCMAPI.h 2018
uim.h 2018
voice.h 2036
wds.h 2064

Chapter 6

Module Documentation

6.1 Device Connectivity Service (DCS)

Files

- file qaGobiApiDcs.h

  Device Connectivity Service API function prototypes.

6.1.1 Detailed Description

6.2 Wireless Data Service (WDS)

Files

- file qaGobiApiTableCallEndReasons.h

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

  Carrier Codes table.
- file qaGobiApiTableRegionCodes.h

  Region Codes table.
- file qaGobiApiWds.h

  Wireless Data Service API function prototypes.

6.2.1 Detailed Description

6.3 Device Management Service (DMS)

Files

- file qaGobiApiTablePowerModes.h

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

  Device Management Service API function prototypes.

6.3.1 Detailed Description

6.4 Network Access Service (NAS)

Files

- file qaGobiApiTableRadioInterfaces.h

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

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

  Network Access Service API function prototypes.

6.4.1 Detailed Description

6.5 CallBacK registration (CBK)

Files

- file qaGobiApiCbk.h

  Callback Service API function prototypes.

6.5.1 Detailed Description

6.6 Short Message Service (SMS)

Files

- file qaGobiApiSms.h

  Short Message Service API function prototypes.

6.6.1 Detailed Description

6.7 Position Determination Service (PDS)

Files

- file qaGobiApiTableGpsCapabilityCodes.h

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

  Position Determination Service API function prototypes.

6.7.1 Detailed Description

6.8 Card Application Toolkit (CAT)

Files

- file qaGobiApiCat.h

  Card Application Toolkit API function headers.

6.8.1 Detailed Description

6.9 Remote Management Service (RMS)

Files

- file qaGobiApiRms.h

  Remote Management Service API function prototypes.

6.9.1 Detailed Description

6.10 Firmware Management Service (FMS)

Files

- file qaGobiApiFms.h

  Firmware Management Service API function prototypes.

6.10.1 Detailed Description

6.11 Open Mobile Alliance Service (OMA)

Files

- file qaGobiApiOmadm.h

  Open Mobile Alliance Device Management Service API function prototypes.

6.11.1 Detailed Description

6.12 Specific Absorption Rate (SAR)

Files

- file qaGobiApiSar.h

  Specific Absorption Rate API function prototypes.

6.12.1 Detailed Description

6.13 SWI Open Mobile Alliance Service (SWIOMA)

Files

- file qaGobiApiTableSwiOMADMUpdateCompleteStatus.h

  Update Complete Status table.
- file qaGobiApiTableSwiOMADMSessionStatus.h

  Session Status table.
- file qaGobiApiSwiOmadms.h

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

6.13.1 Detailed Description

6.14 Voice Service (VOICE)

Files

- file qaGobiApiTableServiceOptions.h

  Voice Service Options.
- file qaGobiApiTableVoiceCallEndReasons.h

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

  Data Coding Scheme.
- file qaGobiApiTableCallControlReturnReasons.h

  Call Control Return Reasons table.
- file qaGobiApiTableSupServiceInfoClasses.h

  Voice Supplementary Service Information Classes.
- file qaGobiApiVoice.h

  Voice Service API function prototypes.

6.14.1 Detailed Description

6.15 Non-service specific APIs (SWI)

Files

- file qaGobiApiSwi.h

  SWI API function prototypes.

6.15.1 Detailed Description

6.16 User Identity Module Service (UIM)

Files

- file qaGobiApiUim.h

  Uim Service API function prototypes.

6.16.1 Detailed Description

6.17 Audio Service (AUDIO)

Files

- file qaGobiApiAudio.h

  Audio Service API function prototypes.

6.17.1 Detailed Description

6.18 Quality of Service (QOS)

Files

- file qaGobiApiQos.h

  Quality of Service API function prototypes.

6.18.1 Detailed Description

6.19 IMS Service (IMS)

Files

- file qaGobiApiIms.h

  IMS Service API function prototypes.
- file qaGobiApiImsa.h

  IMSA Service API function prototypes.

6.19.1 Detailed Description

6.20 SWI Audio Service(SWIAUDIO)

Files

- file qaGobiApiTableSwiAudio.h

  Swi Audio related tables.
- file qaGobiApiSwiAudio.h

  M2M Audio Service API function prototypes.

6.20.1 Detailed Description

6.21 Location Service(LOC)

Files

- file qaGobiApiLoc.h

  Location API function prototypes.

6.21.1 Detailed Description

6.22 Thermal Mitigation Device(TMD)

Files

- file qaGobiApiTmd.h

  Thermal Mitigation Device API function prototypes.

6.22.1 Detailed Description

6.23 AVMS (SWIAVMS)

Files

- file qaGobiApiSwiAvms.h

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

6.23.1 Detailed Description

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

pRegTransLayerInfoEvt	- • 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
pRegTransNWRegInfoEvt	- • 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
pRegCallStatInfoEvt	- • 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::pConnectionRetry-Timers

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

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).

8.113.2 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

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).

8.129.2 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 crash information parameters

Parameters

crashStatus[OUT]	• Device Crash Status • 0 - no crash • 1 - crash has occurred
crashInfo[OUT]	• See crashInformation

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

pDevCrashStatus[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 crash information

Parameters

numCrashes[OUT]	• number of instances of the remaining fields
crashId[OUT]	• ramdom crash id assigned at crash
crashData	• crash data[OUT]
crashStrlen[IN/OUT]	• length of the crashString field as an input, length of the crashString field returned by API as an OUTPUT
crashString	• crash string
gcdumpStrlen[OUT]	• length of the gcdumpString field as an input, length of the gcdumpString field returned by API as an OUTPUT
gcdumpString[OUT]	• gcdump string for the crash

8.134.2 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

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

8.147.2 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

TlvPresent	• Tlv Present
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

TlvPresent	• Tlv Present
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) • NULL terminated ASCII string.
value_length	• length of cust_value field
cust_value	• Customization Setting Value (Maximum 8 bytes)
cust_attr	• Customization Setting attribute through QMI – 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 DMScustSettingInfo 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

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

8.175.2 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) • NULL terminated ASCII string.
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

pGetResetInd	[IN,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

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

8.185.2 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 Detailed Description

This structure contains Extended Roaming Indicator(ERI) file parameters

Parameters

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

8.190.2 Field Documentation

8.190.2.1 uint8_t eriDataparams::eriData[1024]

8.190.2.2 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 ~ 0FF - Insignia defined error code * 100 ~ 1FF - Sierra defined error code * See qaGobiApiTableFwDldErrorCodes.h for more detailed information – 0x800000nn - FDT/SSDP reported errors while nn will be the exact error number * 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

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

pDataBearerTech	• 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
pLastCallTXOKBytesCnt	• 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.
pLastCallRXOKBytesCnt	• 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.
pMdmCallDurationActive	• 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

pDestSMSNum[-OUT]	• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars. • Optional parameter.
pDestSMS-Content[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

pTimeStamp[O-UT]	• Timestamp of the last time power loss was detected and Dying Gasp feature was triggered. • UTC time in seconds since Jan 06, 1980 (GPS Epoch).
pSMS-AttemptedFlag[-OUT]	• Indicates whether device attempted to send SMS in the last power loss event. • 0 - SMS not attempted • 1 - SMS attempted • This only indicates device sent the SMS but does not guarantee network delivery.

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

pCDMAFrameErrRate[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.
pHDRPackErrRate[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
pSMSOverIPNw-Ind	• SMS over IP Network Indication Flag – TRUE - Turn on mobile-originated SMS – FALSE - Turn off mobile-originated SMS
pPhoneCtxtURI-Len[IN/OUT]	• Size in bytes assigned to the Phone context Universal Resource Identifier to follow
pPhoneCtxtURI	• Phone context universal resource identifier • 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
pIMSDomain-Len[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

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.

8.264.2 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[SLQS_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 Detailed Description

This structure is used to store image information

Parameters

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

8.268.2 Field Documentation

8.268.2.1 uint8_t image_info_t::buildID[255]

8.268.2.2 uint8_t image_info_t::buildIDLen

8.268.2.3 uint8_t image_info_t::imageType

8.268.2.4 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_SERVICE_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

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 - Enabled – False - Disabled
pScrAmrEnable	• Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB) • Values: – True - Enabled – False - Disabled
pScrAmrWb-Enable	• 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 IPv4Info Struct Reference

Data Fields

- ULONG address
- WORD port
- BYTE TlvPresent

8.292.1 Detailed Description

This structure contains LOC Server IPV4 info field.

Parameters

address	• IPv4 address.
port	• IPv4 port.
TlvPresent	• Tlv Present or not

8.292.2 Field Documentation

8.292.2.1 ULONG IPv4Info::address

8.292.2.2 WORD IPv4Info::port

8.292.2.3 BYTE IPv4Info::TlvPresent

8.293 IPv6Addr Struct Reference

Data Fields

- BYTE addr [16]
- BYTE prefixLen

8.293.1 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.

8.293.2 Field Documentation

8.293.2.1 BYTE IPv6Addr::addr[16]

8.293.2.2 BYTE IPv6Addr::prefixLen

8.294 IPV6AddressInfo Struct Reference

Data Fields

- BYTE IPV6PrefixLen
- USHORT IPAddressV6 [8]

8.294.1 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.

8.294.2 Field Documentation

8.294.2.1 USHORT IPV6AddressInfo::IPAddressV6[8]

8.294.2.2 BYTE IPV6AddressInfo::IPV6PrefixLen

8.295 ipv6AddressInfo Struct Reference

Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

8.295.1 Detailed Description

This structure contains the IPv6 address information

Parameters

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

8.295.2 Field Documentation

8.295.2.1 uint16_t ipv6AddressInfo::IPAddressV6[8]

8.295.2.2 uint8_t ipv6AddressInfo::IPV6PrefixLen

8.296 IPV6GWAddressInfo Struct Reference

Data Fields

- BYTE gwV6PrefixLen
- USHORT gwAddressV6 [8]

8.296.1 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.

8.296.2 Field Documentation

8.296.2.1 USHORT IPV6GWAddressInfo::gwAddressV6[8]

8.296.2.2 BYTE IPV6GWAddressInfo::gwV6PrefixLen

8.297 IPv6Info Struct Reference

Data Fields

- WORD address [8]
- ULONG port
- BYTE TlvPresent

8.297.1 Detailed Description

This structure contains LOC Get Server IPV6 info field.

Parameters

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

8.297.2 Field Documentation

8.297.2.1 WORD IPv6Info::address[8]

8.297.2.2 ULONG IPv6Info::port

8.297.2.3 BYTE IPv6Info::TlvPresent

8.298 IPv6TrafCls Struct Reference

Data Fields

- BYTE val
- BYTE mask

8.298.1 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

8.298.2 Field Documentation

8.298.2.1 BYTE IPv6TrafCls::mask

8.298.2.2 BYTE IPv6TrafCls::val

8.299 LibPackGPRSRequestedQoS Struct Reference

Data Fields

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

8.299.1 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

8.299.2 Field Documentation

8.299.2.1 uint32_t LibPackGPRSRequestedQoS::delayClass

8.299.2.2 uint32_t LibPackGPRSRequestedQoS::meanThroughputClass

8.299.2.3 uint32_t LibPackGPRSRequestedQoS::peakThroughputClass

8.299.2.4 uint32_t LibPackGPRSRequestedQoS::precedenceClass

8.299.2.5 uint32_t LibPackGPRSRequestedQoS::reliabilityClass

8.300 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

8.300.1 Detailed Description

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

Parameters

pProfileName	• One or more uint8_ts describing the profile
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
pPdpDataComp-Type	• 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.
pPriDNSIPv4-AddPref	• Primary DNS IPv4 Address Preference
pSecDNSIPv4-AddPref	• Secondary DNS IPv4 Address Preference
pUMTSReqQoS	• UMTS Requested QoS
pUMTSMinQoS	• UMTS Minimum QoS
pGPRS-RequestedQoS	• GPRS Requested QoS
pGPRS-MinimumQoS	• 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

pPrimaryDNSIPv6addpref	• 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
pSecondaryDNSIPv6addpref	• Secondary DNS IPv6 address preference
paddrAllocationPref	• 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
pAPNDisabledFlag	• 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 SLQSGetProfileSettings().
pPDNInactivTimeout	• 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().
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 SLQSGetProfileSettings().

8.300.2 Field Documentation

8.300.2.1 uint8_t* LibpackProfile3GPP::pAddrAllocPref

8.300.2.2 uint8_t* LibpackProfile3GPP::pAPNClass

8.300.2.3 uint8_t* LibpackProfile3GPP::pAPNDisabledFlag

8.300.2.4 uint8_t* LibpackProfile3GPP::pAPNName

8.300.2.5 uint16_t* LibpackProfile3GPP::pAPNnameSize

8.300.2.6 uint8_t* LibpackProfile3GPP::pAuthenticationPref

8.300.2.7 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSMinimumQoS

8.300.2.8 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSRequestedQos

8.300.2.9 uint8_t* LibpackProfile3GPP::pImCnFlag

8.300.2.10 uint32_t* LibpackProfile3GPP::pIPv4AddrPref

8.300.2.11 uint16_t* LibpackProfile3GPP::pIPv6AddPref

8.300.2.12 uint8_t* LibpackProfile3GPP::pPassword

8.300.2.13 uint16_t* LibpackProfile3GPP::pPasswordSize

8.300.2.14 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingDhcp

8.300.2.15 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingPCO

8.300.2.16 uint32_t* LibpackProfile3GPP::pPDNInactivTimeout

8.300.2.17 uint8_t* LibpackProfile3GPP::pPdpAccessConFlag

8.300.2.18 uint8_t* LibpackProfile3GPP::pPdpContext

8.300.2.19 uint8_t* LibpackProfile3GPP::pPdpDataCompType

8.300.2.20 uint8_t* LibpackProfile3GPP::pPdpHdrCompType

8.300.2.21 uint8_t* LibpackProfile3GPP::pPDPtype

8.300.2.22 uint32_t* LibpackProfile3GPP::pPriDNSIPv4AddPref

8.300.2.23 uint16_t* LibpackProfile3GPP::pPriDNSIPv6addpref

8.300.2.24 uint8_t* LibpackProfile3GPP::pPrimaryID

8.300.2.25 uint8_t* LibpackProfile3GPP::pProfilename

8.300.2.26 uint16_t* LibpackProfile3GPP::pProfilenameSize

8.300.2.27 LibPackQosClassID* LibpackProfile3GPP::pQosClassID

8.300.2.28 uint32_t* LibpackProfile3GPP::pSecDNSIPv4AddPref

8.300.2.29 uint16_t* LibpackProfile3GPP::pSecDNSIPv6addpref

8.300.2.30 uint8_t* LibpackProfile3GPP::pSecondaryFlag

8.300.2.31 LibPackTFTIDParams* LibpackProfile3GPP::pTFTID1Params

8.300.2.32 LibPackTFTIDParams* LibpackProfile3GPP::pTFTID2Params

8.300.2.33 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSMinQoS

8.300.2.34 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSMinQosSigInd

8.300.2.35 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSReqQoS

8.300.2.36 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSReqQoSSigInd

8.300.2.37 **uint8_t*** LibpackProfile3GPP::pUsername

8.300.2.38 **uint16_t*** LibpackProfile3GPP::pUsernameSize

8.301 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

8.301.1 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 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 SLQSGetProfile-Settings().
pAPNEnabled3-GPP2	• 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 SLQSGetProfile-Settings().

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().

8.301.2 Field Documentation

8.301.2.1 uint8_t* LibpackProfile3GPP2::pAllowLinger

8.301.2.2 uint8_t* LibpackProfile3GPP2::pAPNClass3GPP2

8.301.2.3 uint8_t* LibpackProfile3GPP2::pAPNEnabled3GPP2

8.301.2.4 uint8_t* LibpackProfile3GPP2::pApnString

8.301.2.5 uint16_t* LibpackProfile3GPP2::pApnStringSize

8.301.2.6 uint8_t* LibpackProfile3GPP2::pAppPriority

8.301.2.7 uint32_t* LibpackProfile3GPP2::pAppType

8.301.2.8 uint8_t* LibpackProfile3GPP2::pAuthPassword

8.301.2.9 uint16_t* LibpackProfile3GPP2::pAuthPasswordSize

8.301.2.10 uint8_t* LibpackProfile3GPP2::pAuthProtocol

8.301.2.11 uint8_t* LibpackProfile3GPP2::pAuthRetryCount

8.301.2.12 uint16_t* LibpackProfile3GPP2::pAuthTimeout

8.301.2.13 uint8_t* LibpackProfile3GPP2::pDataMode

8.301.2.14 uint8_t* LibpackProfile3GPP2::pDataRate

8.301.2.15 uint16_t* LibpackProfile3GPP2::pIpcpAckTimeout

8.301.2.16 uint8_t* LibpackProfile3GPP2::pIpcpCreqRetryCount

8.301.2.17 uint8_t* LibpackProfile3GPP2::pIsPcscfAddressNedded

8.301.2.18 uint16_t* LibpackProfile3GPP2::pLcpAckTimeout

8.301.2.19 uint8_t* LibpackProfile3GPP2::pLcpCreqRetryCount

8.301.2.20 uint8_t* LibpackProfile3GPP2::pNegoDnsSrvrPref

8.301.2.21 uint32_t* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2

8.301.2.22 uint8_t* LibpackProfile3GPP2::pPdnType

8.301.2.23 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimer1x

8.301.2.24 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimerDO

8.301.2.25 uint32_t* LibpackProfile3GPP2::pPrimaryV4DnsAddress

8.301.2.26 uint16_t* LibpackProfile3GPP2::pPriV6DnsAddress

8.301.2.27 uint8_t* LibpackProfile3GPP2::pRATType

8.301.2.28 uint32_t* LibpackProfile3GPP2::pSecondaryV4DnsAddress

8.301.2.29 uint16_t* LibpackProfile3GPP2::pSecV6DnsAddress

8.301.2.30 uint8_t* LibpackProfile3GPP2::pUserId

8.301.2.31 uint16_t* LibpackProfile3GPP2::pUserIdSize

8.302 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

8.302.1 Detailed Description

This structure contains Input/Output parameters of pack_wds_SLQSCreateProfile

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

Parameters

pProfileName	• One or more bytes describing the profile
pProfilename-Size;	• This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 bytes.
pPDPType	• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile – 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
pPdpDataComp-Type	• 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.
pPriDNSIPv4-AddPref	• Primary DNS IPv4 Address Preference
pSecDNSIPv4-AddPref	• Secondary DNS IPv4 Address Preference
pUMTSReqQoS	• UMTS Requested QoS
pUMTSMinQoS	• UMTS Minimum QoS
pGPRS-RequestedQoS	• GPRS Minimum QoS
pUsername	• User name
pUsernameSize;	• This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 bytes.
pPassword	• Password
pPasswordSize;	• This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 bytes.
pAuthentication-Pref	• Authentication Preference – 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

pPrimaryDNSIPv6addpref	• 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
pSecondaryDNSIPv6addpref	• Secondary DNS IPv6 address preference
paddrAllocationPref	• 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
pAPNDisabledFlag	• 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 SLQSGetProfileSettings().
pPDNInactivTimeout	• 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().
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 SLQSGetProfileSettings().

8.302.2 Field Documentation

8.302.2.1 uint8_t* LibPackprofile_3GPP::pAddrAllocPref

8.302.2.2 uint8_t* LibPackprofile_3GPP::pAPNClass

8.302.2.3 uint8_t* LibPackprofile_3GPP::pAPNDisabledFlag

8.302.2.4 uint8_t* LibPackprofile_3GPP::pAPNName

8.302.2.5 uint16_t* LibPackprofile_3GPP::pAPNnameSize

8.302.2.6 uint8_t* LibPackprofile_3GPP::pAuthenticationPref

8.302.2.7 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSMinimumQoS

8.302.2.8 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSRequestedQos

8.302.2.9 uint8_t* LibPackprofile_3GPP::pImCnFlag

8.302.2.10 uint32_t* LibPackprofile_3GPP::pIPv4AddrPref

8.302.2.11 uint16_t* LibPackprofile_3GPP::pIPv6AddPref

8.302.2.12 uint8_t* LibPackprofile_3GPP::pPassword

8.302.2.13 uint16_t* LibPackprofile_3GPP::pPasswordSize

8.302.2.14 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingDhcp

8.302.2.15 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingPCO

8.302.2.16 uint32_t* LibPackprofile_3GPP::pPDNInactivTimeout

8.302.2.17 uint8_t* LibPackprofile_3GPP::pPdpAccessConFlag

8.302.2.18 uint8_t* LibPackprofile_3GPP::pPdpContext

8.302.2.19 uint8_t* LibPackprofile_3GPP::pPdpDataCompType

8.302.2.20 uint8_t* LibPackprofile_3GPP::pPdpHdrCompType

8.302.2.21 uint8_t* LibPackprofile_3GPP::pPDPtype

8.302.2.22 uint32_t* LibPackprofile_3GPP::pPriDNSIPv4AddPref

8.302.2.23 uint16_t* LibPackprofile_3GPP::pPriDNSIPv6addpref

8.302.2.24 uint8_t* LibPackprofile_3GPP::pPrimaryID

8.302.2.25 uint8_t* LibPackprofile_3GPP::pProfilename

8.302.2.26 uint16_t* LibPackprofile_3GPP::pProfilenameSize

8.302.2.27 LibPackQosClassID* LibPackprofile_3GPP::pQosClassID

8.302.2.28 uint32_t* LibPackprofile_3GPP::pSecDNSIPv4AddPref

8.302.2.29 uint16_t* LibPackprofile_3GPP::pSecDNSIPv6addpref

8.302.2.30 uint8_t* LibPackprofile_3GPP::pSecondaryFlag

8.302.2.31 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID1Params

8.302.2.32 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID2Params

8.302.2.33 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSMinQoS

8.302.2.34 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSMinQosSigInd

8.302.2.35 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSReqQoS

8.302.2.36 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSReqQoSSigInd

8.302.2.37 uint8_t* LibPackprofile_3GPP::pUsername

8.302.2.38 uint16_t* LibPackprofile_3GPP::pUsernameSize

8.303 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

8.303.1 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 SLQSGetProfile-Settings().
pAPNEnabled3-GPP2	• 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 SLQSGetProfile-Settings().

pPDNInactiv-Timeout3GPP2	• 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().
pAPNClass3GP-P2	• 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().

8.303.2 Field Documentation

8.303.2.1 uint8_t* LibPackprofile_3GPP2::pAllowLinger

8.303.2.2 uint8_t* LibPackprofile_3GPP2::pAPNClass3GPP2

8.303.2.3 uint8_t* LibPackprofile_3GPP2::pAPNEnabled3GPP2

8.303.2.4 uint8_t* LibPackprofile_3GPP2::pApnString

8.303.2.5 uint16_t* LibPackprofile_3GPP2::pApnStringSize

8.303.2.6 uint8_t* LibPackprofile_3GPP2::pAppPriority

8.303.2.7 uint32_t* LibPackprofile_3GPP2::pAppType

8.303.2.8 uint8_t* LibPackprofile_3GPP2::pAuthPassword

8.303.2.9 uint16_t* LibPackprofile_3GPP2::pAuthPassword_tSize

8.303.2.10 uint8_t* LibPackprofile_3GPP2::pAuthProtocol

8.303.2.11 uint8_t* LibPackprofile_3GPP2::pAuthRetryCount

8.303.2.12 uint16_t* LibPackprofile_3GPP2::pAuthTimeout

8.303.2.13 uint8_t* LibPackprofile_3GPP2::pDataMode

8.303.2.14 uint8_t* LibPackprofile_3GPP2::pDataRate

8.303.2.15 uint16_t* LibPackprofile_3GPP2::pIpcpAckTimeout

8.303.2.16 uint8_t* LibPackprofile_3GPP2::pIpcpCreqRetryCount

8.303.2.17 uint8_t* LibPackprofile_3GPP2::pIsPcscfAddressNedded

8.303.2.18 uint16_t* LibPackprofile_3GPP2::pLcpAckTimeout

8.303.2.19 uint8_t* LibPackprofile_3GPP2::pLcpCreqRetryCount

8.303.2.20 uint8_t* LibPackprofile_3GPP2::pNegoDnsSrvrPref

8.303.2.21 uint32_t* LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2

8.303.2.22 uint8_t* LibPackprofile_3GPP2::pPdnType

8.303.2.23 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimer1x

8.303.2.24 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimerDO

8.303.2.25 uint32_t* LibPackprofile_3GPP2::pPrimaryV4DnsAddress

8.303.2.26 uint16_t* LibPackprofile_3GPP2::pPriV6DnsAddress

8.303.2.27 uint8_t* LibPackprofile_3GPP2::pRATType

8.303.2.28 uint32_t* LibPackprofile_3GPP2::pSecondaryV4DnsAddress

8.303.2.29 uint16_t* LibPackprofile_3GPP2::pSecV6DnsAddress

8.303.2.30 uint8_t* LibPackprofile_3GPP2::pUserId

8.303.2.31 uint16_t* LibPackprofile_3GPP2::pUserIdSize

8.304 LibPackQosClassID Struct Reference

Data Fields

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

8.304.1 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	• Maximum DL bit rate

gUlBitRate	• Guaranteed UL bit rate
maxUlBitRate	• Maximum UL bit rate

8.304.2 Field Documentation

8.304.2.1 uint8_t LibPackQosClassID::gDlBitRate

8.304.2.2 uint32_t LibPackQosClassID::gUlBitRate

8.304.2.3 uint32_t LibPackQosClassID::maxDlBitRate

8.304.2.4 uint32_t LibPackQosClassID::maxUlBitRate

8.304.2.5 uint8_t LibPackQosClassID::QCI

8.305 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

8.305.1 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

ipVersion	• IP version number – 4 - IPv4 – 6 - IPv6
pSourceIP	• 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

8.305.2 Field Documentation

8.305.2.1 uint16_t LibPackTFTIDParams::destPortRangeEnd

8.305.2.2 uint32_t LibPackTFTIDParams::destPortRangeStart

8.305.2.3 uint8_t LibPackTFTIDParams::eValid

8.305.2.4 uint8_t LibPackTFTIDParams::filterId

8.305.2.5 uint32_t LibPackTFTIDParams::flowLabel

8.305.2.6 uint32_t LibPackTFTIDParams::IPSECSPI

8.305.2.7 uint8_t LibPackTFTIDParams::ipVersion

8.305.2.8 uint8_t LibPackTFTIDParams::nextHeader

8.305.2.9 uint16_t* LibPackTFTIDParams::pSourceIP

8.305.2.10 uint8_t LibPackTFTIDParams::sourceIPMask

8.305.2.11 uint16_t LibPackTFTIDParams::srcPortRangeEnd

8.305.2.12 uint16_t LibPackTFTIDParams::srcPortRangeStart

8.305.2.13 uint16_t LibPackTFTIDParams::tosMask

8.306 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

8.306.1 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.

8.306.2 Field Documentation

8.306.2.1 uint8_t LibPackUMTSQoS::deliveryErrSDU

8.306.2.2 uint32_t LibPackUMTSQoS::grntDownlinkBitrate

8.306.2.3 uint32_t LibPackUMTSQoS::grntUplinkBitrate

8.306.2.4 uint32_t LibPackUMTSQoS::maxDownlinkBitrate

8.306.2.5 uint32_t LibPackUMTSQoS::maxSDUSize

8.306.2.6 uint32_t LibPackUMTSQoS::maxUplinkBitrate

8.306.2.7 uint8_t LibPackUMTSQoS::qosDeliveryOrder

8.306.2.8 uint8_t LibPackUMTSQoS::resBerRatio

8.306.2.9 uint8_t LibPackUMTSQoS::sduErrorRatio

8.306.2.10 uint8_t LibPackUMTSQoS::trafficClass

8.306.2.11 uint32_t LibPackUMTSQoS::trafficPriority

8.306.2.12 uint32_t LibPackUMTSQoS::transferDelay

8.307 LibPackUMTSReqQoSSigInd Struct Reference

Data Fields

- LibPackUMTSQoS UMTSReqQoS
- uint8_t SigInd

8.307.1 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

8.307.2 Field Documentation

8.307.2.1 uint8_t LibPackUMTSReqQoSSigInd::SigInd

8.307.2.2 LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS

8.308 lineCtrlInfo Struct Reference

Data Fields

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

8.308.1 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

8.308.2 Field Documentation

8.308.2.1 BYTE lineCtrlInfo::polarityIncluded

8.308.2.2 BYTE lineCtrlInfo::pwrDenialTime

8.308.2.3 BYTE lineCtrlInfo::revPolarity

8.308.2.4 BYTE lineCtrlInfo::toggleMode

8.309 loc_BdsSV Struct Reference

Data Fields

- uint16_t id
- uint8_t mask

8.309.1 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

8.309.2 Field Documentation

8.309.2.1 uint16_t loc_BdsSV::id

8.309.2.2 uint8_t loc_BdsSV::mask

8.310 loc_BdsSVInfo Struct Reference

Data Fields

- uint8_t len
- loc_BdsSV * pSV

8.310.1 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

8.310.2 Field Documentation

8.310.2.1 uint8_t loc_BdsSVInfo::len

8.310.2.2 loc_BdsSV* loc_BdsSVInfo::pSV

8.311 loc_CellDb Struct Reference

Data Fields

- uint32_t mask

8.311.1 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

8.311.2 Field Documentation

8.311.2.1 uint32_t loc_CellDb::mask

8.312 loc_ClkInfo Struct Reference

Data Fields

- uint32_t mask

8.312.1 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

8.312.2 Field Documentation

8.312.2.1 uint32_t loc_ClkInfo::mask

8.313 loc_GnssData Struct Reference

Data Fields

- uint64_t mask

8.313.1 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

8.313.2 Field Documentation

8.313.2.1 uint64_t loc_GnssData::mask

8.314 loc_gpsTime Struct Reference

Data Fields

- uint16_t gpsWeek
- uint32_t gpsTimeOfWeekMs

8.314.1 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

8.314.2 Field Documentation

8.314.2.1 uint32_t loc_gpsTime::gpsTimeOfWeekMs

8.314.2.2 uint16_t loc_gpsTime::gpsWeek

8.315 loc_IPv4Info Struct Reference

Data Fields

- uint32_t address
- uint16_t port
- uint8_t TlvPresent

8.315.1 Detailed Description

This structure contains LOC Server IPV4 info field.

Parameters

address	• IPv4 address.
port	• IPv4 port.
TlvPresent	• Tlv Present or not

8.315.2 Field Documentation

8.315.2.1 uint32_t loc_IPv4Info::address

8.315.2.2 uint16_t loc_IPv4Info::port

8.315.2.3 uint8_t loc_IPv4Info::TlvPresent

8.316 loc_IPv6Info Struct Reference

Data Fields

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

8.316.1 Detailed Description

This structure contains LOC Get Server IPV6 info field.

Parameters

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

8.316.2 Field Documentation

8.316.2.1 uint16_t loc_IPv6Info::address[8]

8.316.2.2 uint32_t loc_IPv6Info::port

8.316.2.3 uint8_t loc_IPv6Info::TlvPresent

8.317 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

8.317.1 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

8.317.2 Field Documentation

8.317.2.1 uint8_t loc_LocApplicationInfo::appNameLength

8.317.2.2 uint8_t loc_LocApplicationInfo::appProviderLength

8.317.2.3 uint8_t loc_LocApplicationInfo::appVersionLength

8.317.2.4 uint8_t loc_LocApplicationInfo::appVersionValid

8.317.2.5 uint8_t* loc_LocApplicationInfo::pAppName

8.317.2.6 uint8_t* loc_LocApplicationInfo::pAppProvider

8.317.2.7 uint8_t* loc_LocApplicationInfo::pAppVersion

8.318 loc_precisionDilution Struct Reference

Data Fields

- uint32_t PDOP

- uint32_t HDOP
- uint32_t VDOP

8.318.1 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)

8.318.2 Field Documentation

8.318.2.1 uint32_t loc_precisionDilution::HDOP

8.318.2.2 uint32_t loc_precisionDilution::PDOP

8.318.2.3 uint32_t loc_precisionDilution::VDOP

8.319 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

8.319.1 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

8.319.2 Field Documentation

8.319.2.1 float loc_satelliteInfo::azimuth

8.319.2.2 float loc_satelliteInfo::elevation

8.319.2.3 uint16_t loc_satelliteInfo::gnssSvId

8.319.2.4 uint8_t loc_satelliteInfo::healthStatus

8.319.2.5 float loc_satelliteInfo::snr

8.319.2.6 uint8_t loc_satelliteInfo::svInfoMask

8.319.2.7 uint8_t loc_satelliteInfo::svListLen

8.319.2.8 uint32_t loc_satelliteInfo::svStatus

8.319.2.9 uint32_t loc_satelliteInfo::system

8.319.2.10 uint32_t loc_satelliteInfo::validMask

8.320 loc_sensorDataUsage Struct Reference

Data Fields

- uint32_t usageMask
- uint32_t aidingIndicatorMask

8.320.1 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
aidingIndicator-Mask	• Specifies which results were aided by sensors. – Value * 0x00000001 - AIDED_HEADING * 0x00000002 - AIDED_SPEED * 0x00000004 - AIDED_POSITION * 0x00000008 - AIDED_VELOCITY

8.320.2 Field Documentation

8.320.2.1 uint32_t loc_sensorDataUsage::aidingIndicatorMask

8.320.2.2 uint32_t loc_sensorDataUsage::usageMask

8.321 loc_SV Struct Reference

Data Fields

- uint16_t id
- uint32_t system
- uint8_t mask

8.321.1 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

8.321.2 Field Documentation

8.321.2.1 uint16_t loc_SV::id

8.321.2.2 uint8_t loc_SV::mask

8.321.2.3 uint32_t loc_SV::system

8.322 loc_SVInfo Struct Reference

Data Fields

- uint8_t len
- loc_SV * pSV

8.322.1 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

8.322.2 Field Documentation

8.322.2.1 uint8_t loc_SVInfo::len

8.322.2.2 loc_SV* loc_SVInfo::pSV

8.323 loc_svUsedforFix Struct Reference

Data Fields

- uint8_t gnssSvUsedList_len
- uint16_t gnssSvUsedList [255]

8.323.1 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

8.323.2 Field Documentation

8.323.2.1 uint16_t loc_svUsedforFix::gnssSvUsedList[255]

8.323.2.2 uint8_t loc_svUsedforFix::gnssSvUsedList_len

8.324 loc_urlAddr Struct Reference

Data Fields

- char address [256]
- uint TlvPresent

8.324.1 Detailed Description

This structure contains LOC Get Server URL address field.

Parameters

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

8.324.2 Field Documentation

8.324.2.1 char loc_urlAddr::address[256]

8.324.2.2 uint loc_urlAddr::TlvPresent

8.325 LocApplicationInfo Struct Reference

Data Fields

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

8.325.1 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

8.325.2 Field Documentation

8.325.2.1 BYTE LocApplicationInfo::appNameLength

8.325.2.2 BYTE LocApplicationInfo::appProviderLength

8.325.2.3 CHAR LocApplicationInfo::appVersionLength

8.325.2.4 BYTE LocApplicationInfo::appVersionValid

8.325.2.5 CHAR* LocApplicationInfo::pAppName

8.325.2.6 CHAR* LocApplicationInfo::pAppProvider

8.325.2.7 CHAR* LocApplicationInfo::pAppVersion

8.326 LocDelAssDataReq Struct Reference

Data Fields

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

8.326.1 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

8.326.2 Field Documentation

8.326.2.1 BdsSVInfo* LocDelAssDataReq::pBdsSVInfo

8.326.2.2 CellDb* LocDelAssDataReq::pCellDb

8.326.2.3 ClkInfo* LocDelAssDataReq::pClkInfo

8.326.2.4 GnssData* LocDelAssDataReq::pGnssData

8.326.2.5 SVInfo* LocDelAssDataReq::pSVInfo

8.327 LOCEventRegisterReqResp Struct Reference

Data Fields

- ULONGLONG eventRegister

8.327.1 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.

8.327.2 Field Documentation

8.327.2.1 ULONGLONG LOCEventRegisterReqResp::eventRegister

8.328 LOCExtPowerStateReqResp Struct Reference

Data Fields

- ULONG extPowerState

8.328.1 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

8.328.2 Field Documentation

8.328.2.1 ULONG LOCExtPowerStateReqResp::extPowerState

8.329 LocGetServerReq Struct Reference

Data Fields

- ULONG serverType
- BYTE * pServerAddrTypeMask

8.329.1 Detailed Description

This structure contains parameters to gets the location server from the location engine.

Parameters

serverType	• Type of server. • Valid values – eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE – eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC – eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP – eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4)- Server type is custom PDE
pServerAddr-TypeMask	• Optional parameter • Type of address the client wants. If unspecified, the indication will contain all the types of addresses it has for the specified server type. • Valid bitmasks – 0x01 – IPv4 – 0x02 – IPv6 – 0x04 – URL

8.329.2 Field Documentation

8.329.2.1 BYTE* LocGetServerReq::pServerAddrTypeMask

8.329.2.2 ULONG LocGetServerReq::serverType

8.330 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

8.330.1 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.

8.330.2 Field Documentation

8.330.2.1 altitudeSrcInfo∗ LocInjectPositionReq::pAltitudeSrcInfo

8.330.2.2 ULONG∗ LocInjectPositionReq::pAltitudeWrtEllipsoid

8.330.2.3 ULONG∗ LocInjectPositionReq::pAltitudeWrtMeanSeaLevel

8.330.2.4 BYTE* LocInjectPositionReq::pHorConfidence

8.330.2.5 ULONG* LocInjectPositionReq::pHorReliability

8.330.2.6 ULONG* LocInjectPositionReq::pHorUncCircular

8.330.2.7 ULONGLONG* LocInjectPositionReq::pLatitude

8.330.2.8 ULONGLONG* LocInjectPositionReq::pLongitude

8.330.2.9 ULONG* LocInjectPositionReq::pPositionSrc

8.330.2.10 BYTE* LocInjectPositionReq::pRawHorConfidence

8.330.2.11 ULONG* LocInjectPositionReq::pRawHorUncCircular

8.330.2.12 ULONG* LocInjectPositionReq::pTimestampAge

8.330.2.13 ULONGLONG* LocInjectPositionReq::pTimestampUtc

8.330.2.14 BYTE* LocInjectPositionReq::pVertConfidence

8.330.2.15 ULONG* LocInjectPositionReq::pVertReliability

8.330.2.16 ULONG* LocInjectPositionReq::pVertUnc

8.331 LocInjectSensorDataReq Struct Reference

Data Fields

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

8.331.1 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

8.331.2 Field Documentation

8.331.2.1 sensorData* LocInjectSensorDataReq::pAcceleroData

8.331.2.2 tempratureData* LocInjectSensorDataReq::pAcceleroTempData

8.331.2.3 ULONG* LocInjectSensorDataReq::pAcceleroTimeSrc

8.331.2.4 sensorData* LocInjectSensorDataReq::pGyroData

8.331.2.5 tempratureData* LocInjectSensorDataReq::pGyroTempData

8.331.2.6 ULONG* LocInjectSensorDataReq::pGyroTimeSrc

8.331.2.7 ULONG* LocInjectSensorDataReq::pOpaqueIdentifier

8.332 LocSetCradleMountReq Struct Reference

Data Fields

- ULONG state
- BYTE * pConfidence

8.332.1 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

8.332.2 Field Documentation

8.332.2.1 BYTE* LocSetCradleMountReq::pConfidence

8.332.2.2 ULONG LocSetCradleMountReq::state

8.333 LOCStartReq Struct Reference

Data Fields

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

8.333.1 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

8.333.2 Field Documentation

8.333.2.1 struct LocApplicationInfo* LOCStartReq::pApplicationInfo

8.333.2.2 ULONG* LOCStartReq::pConfigAltitudeAssumed

8.333.2.3 ULONG* LOCStartReq::pHorizontalAccuracyLvl

8.333.2.4 ULONG* LOCStartReq::pIntermediateReportState

8.333.2.5 ULONG* LOCStartReq::pMinIntervalTime

8.333.2.6 ULONG* LOCStartReq::pRecurrenceType

8.333.2.7 BYTE LOCStartReq::SessionId

8.334 LOCStopReq Struct Reference

Data Fields

- BYTE sessionId

8.334.1 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

8.334.2 Field Documentation

8.334.2.1 BYTE LOCStopReq::sessionId

8.335 LteCQIParm Struct Reference

Data Fields

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

8.335.1 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

8.335.2 Field Documentation

8.335.2.1 BYTE LteCQIParm::CQIValueCW0

8.335.2.2 BYTE LteCQIParm::CQIValueCW1

8.335.2.3 BYTE LteCQIParm::ValidityCW0

8.335.2.4 BYTE LteCQIParm::ValidityCW1

8.336 lteEARFCN Struct Reference

Data Fields

- BYTE status
- ULONG earfcn0
- ULONG earfcn1

8.336.1 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.

8.336.2 Field Documentation

8.336.2.1 ULONG lteEARFCN::earfcn0

8.336.2.2 ULONG lteEARFCN::earfcn1

8.336.2.3 BYTE lteEARFCN::status

8.337 lteGsmCellInfo Struct Reference

Data Fields

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

8.337.1 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.

8.337.2 Field Documentation

8.337.2.1 BYTE lteGsmCellInfo::cellReselPriority

8.337.2.2 BYTE lteGsmCellInfo::cells_len

8.337.2.3 gsmCellInfo lteGsmCellInfo::GsmCellInfo[255]

8.337.2.4 BYTE lteGsmCellInfo::nccPermitted

8.337.2.5 BYTE lteGsmCellInfo::threshGsmHigh

8.337.2.6 BYTE lteGsmCellInfo::threshGsmLow

8.338 LTEInfo Struct Reference

Data Fields

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

8.338.1 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

8.338.2 Field Documentation

8.338.2.1 BYTE LTEInfo::band

8.338.2.2 BYTE LTEInfo::bandwidth

8.338.2.3 BYTE LTEInfo::emmConnState

8.338.2.4 BYTE LTEInfo::emmState

8.338.2.5 BYTE LTEInfo::emmSubState

8.338.2.6 WORD LTEInfo::RXChan

8.338.2.7 WORD LTEInfo::TXChan

8.339 LTEInfoInterfreq Struct Reference

Data Fields

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

8.339.1 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.

8.339.2 Field Documentation

8.339.2.1 BYTE LTEInfoInterfreq::freqsLen

8.339.2.2 infoInterFreq LTEInfoInterfreq::InfoInterfreq[255]

8.339.2.3 BYTE LTEInfoInterfreq::ueInIdle

8.340 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]

8.340.1 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.

8.340.2 Field Documentation

8.340.2.1 cellParams LTEInfoIntrafreq::CellParams[255]

8.340.2.2 BYTE LTEInfoIntrafreq::cellReselPriority

8.340.2.3 BYTE LTEInfoIntrafreq::cellsLen

8.340.2.4 WORD LTEInfoIntrafreq::earfcn

8.340.2.5 ULONG LTEInfoIntrafreq::globalCellId

8.340.2.6 BYTE LTEInfoIntrafreq::plmn[3]

8.340.2.7 WORD LTEInfoIntrafreq::servingCellId

8.340.2.8 BYTE LTEInfoIntrafreq::sIntraSearch

8.340.2.9 BYTE LTEInfoIntrafreq::sNonIntraSearch

8.340.2.10 WORD LTEInfoIntrafreq::tac

8.340.2.11 BYTE LTEInfoIntrafreq::threshServingLow

8.340.2.12 BYTE LTEInfoIntrafreq::ueInIdle

8.341 LTEInfoNeighboringGSM Struct Reference

Data Fields

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

8.341.1 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.

8.341.2 Field Documentation

8.341.2.1 BYTE LTEInfoNeighboringGSM::freqsLen

8.341.2.2 lteGsmCellInfo LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.341.2.3 BYTE LTEInfoNeighboringGSM::ueInIdle

8.342 LTEInfoNeighboringWCDMA Struct Reference

Data Fields

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

8.342.1 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.

8.342.2 Field Documentation

8.342.2.1 BYTE LTEInfoNeighboringWCDMA::freqsLen

8.342.2.2 lteWcdmaCellInfo LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.342.2.3 BYTE LTEInfoNeighboringWCDMA::ueInIdle

8.343 LteNasReleaseInfo_s Struct Reference

Data Fields

- BYTE nas_release
- BYTE nas_major
- BYTE nas_minor

8.343.1 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

8.343.2 Field Documentation

8.343.2.1 BYTE LteNasReleaseInfo_s::nas_major

8.343.2.2 BYTE LteNasReleaseInfo_s::nas_minor

8.343.2.3 BYTE LteNasReleaseInfo_s::nas_release

8.344 ltePCI Struct Reference

Data Fields

- BYTE status
- ULONG earfcn
- ULONG pci

8.344.1 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

8.344.2 Field Documentation

8.344.2.1 ULONG ltePCI::earfcn

8.344.2.2 ULONG ltePCI::pci

8.344.2.3 BYTE ltePCI::status

8.345 lteRsrpinformation Struct Reference

Data Fields

- SHORT rsrplevel

8.345.1 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).

8.345.2 Field Documentation

8.345.2.1 SHORT lteRsrpinformation::rsrplevel

8.346 LTERSRPThresh Struct Reference

Data Fields

- BYTE LTERSRPThreshListLen
- WORD * pLTERSRPThreshList

8.346.1 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).

8.346.2 Field Documentation

8.346.2.1 BYTE LTERSRPThresh::LTERSRPThreshListLen

8.346.2.2 WORD* LTERSRPThresh::pLTERSRPThreshList

8.347 LTERSRQThresh Struct Reference

Data Fields

- BYTE LTERSRQThreshListLen
- WORD * pLTERSRQThreshList

8.347.1 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)

8.347.2 Field Documentation

8.347.2.1 BYTE LTERSRQThresh::LTERSRQThreshListLen

8.347.2.2 WORD* LTERSRQThresh::pLTERSRQThreshList

8.348 LTERSSIThresh Struct Reference

Data Fields

- BYTE LTERSSIThreshListLen
- WORD * pLTERSSIThreshList

8.348.1 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)

8.348.2 Field Documentation

8.348.2.1 BYTE LTERSSIThresh::LTERSSIThreshListLen

8.348.2.2 WORD* LTERSSIThresh::pLTERSSIThreshList

8.349 LteSccRxInfoResp Struct Reference

Data Fields

- SccRxInfo * pSccRxInfo

8.349.1 Detailed Description

This structure contains information about the SLQSSwiGetLteSccRxInfo response parameters.

Parameters

pSccRxInfo	[Optional] • See SccRxInfo for more information

8.349.2 Field Documentation

8.349.2.1 SccRxInfo* LteSccRxInfoResp::pSccRxInfo

8.350 LTESigRptCfg Struct Reference

Data Fields

- BYTE rptRate
- BYTE avgPeriod

8.350.1 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

8.350.2 Field Documentation

8.350.2.1 BYTE LTESigRptCfg::avgPeriod

8.350.2.2 BYTE LTESigRptCfg::rptRate

8.351 LTESigRptConfig Struct Reference

Data Fields

- BYTE rptRate
- BYTE avgPeriod

8.351.1 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

8.351.2 Field Documentation

8.351.2.1 BYTE LTESigRptConfig::avgPeriod

8.351.2.2 BYTE LTESigRptConfig::rptRate

8.352 lteSnrinformation Struct Reference

Data Fields

- SHORT snrlevel

8.352.1 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.

8.352.2 Field Documentation

8.352.2.1 SHORT lteSnrinformation::snrlevel

8.353 LTESNRThresh Struct Reference

Data Fields

- BYTE LTESNRThresListLen
- SHORT * pLTESNRThresList

8.353.1 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

8.353.2 Field Documentation

8.353.2.1 BYTE LTESNRThresh::LTESNRThresListLen

8.353.2.2 SHORT* LTESNRThresh::pLTESNRThresList

8.354 LTESNRThreshold Struct Reference

Data Fields

- BYTE LTESNRThreshListLen
- WORD * pLTESNRThreshList

8.354.1 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).

8.354.2 Field Documentation

8.354.2.1 BYTE LTESNRThreshold::LTESNRThreshListLen

8.354.2.2 WORD* LTESNRThreshold::pLTESNRThreshList

8.355 LTESSInfo Struct Reference

Data Fields

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

8.355.1 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,

8.355.2 Field Documentation

8.355.2.1 SHORT LTESSInfo::rsrp

8.355.2.2 INT8 LTESSInfo::rsrq

8.355.2.3 INT8 LTESSInfo::rssi

8.355.2.4 SHORT LTESSInfo::snr

8.356 lteSSInfo Struct Reference

Data Fields

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

8.356.1 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,

8.356.2 Field Documentation

8.356.2.1 int16_t lteSSInfo::rsrp

8.356.2.2 int8_t lteSSInfo::rsrq

8.356.2.3 int8_t lteSSInfo::rssi

8.356.2.4 int16_t lteSSInfo::snr

8.357 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

8.357.1 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

8.357.2 Field Documentation

8.357.2.1 ULONG LTESysInfo::cellId

8.357.2.2 BYTE LTESysInfo::cellIdValid

8.357.2.3 WORD LTESysInfo::lac

8.357.2.4 BYTE LTESysInfo::lacValid

8.357.2.5 BYTE LTESysInfo::MCC[3]

8.357.2.6 BYTE LTESysInfo::MNC[3]

8.357.2.7 BYTE LTESysInfo::networkIdValid

8.357.2.8 BYTE LTESysInfo::regRejectInfoValid

8.357.2.9 BYTE LTESysInfo::rejCause

8.357.2.10 BYTE LTESysInfo::rejectSrvDomain

8.357.2.11 sysInfoCommon LTESysInfo::sysInfoLTE

8.357.2.12 WORD LTESysInfo::tac

8.357.2.13 BYTE LTESysInfo::tacValid

8.358 lteWcdmaCellInfo Struct Reference

Data Fields

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

8.358.1 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.
WCDMACell-Info[MAX_DES-CRIPTION_LE-NGTH]	• See wcdmaCellInfo for more information.

8.358.2 Field Documentation

8.358.2.1 BYTE lteWcdmaCellInfo::cellReselPriority

8.358.2.2 BYTE lteWcdmaCellInfo::cellsLen

8.358.2.3 WORD lteWcdmaCellInfo::threshXhigh

8.358.2.4 WORD lteWcdmaCellInfo::threshXlow

8.358.2.5 WORD lteWcdmaCellInfo::uarfcn

8.358.2.6 wcdmaCellInfo lteWcdmaCellInfo::WCDMACellInfo[255]

8.359 messageModeTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSMessageModeInfo MessageModeInfo

8.359.1 Detailed Description

Parameters

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

8.359.2 Field Documentation

8.359.2.1 sMSMessageModeInfo messageModeTlv::MessageModeInfo

8.359.2.2 uint8_t messageModeTlv::TlvPresent

8.360 messageWaitingInfoContent Struct Reference

Data Fields

- BYTE msgType

- BYTE activeInd
- BYTE msgCount

8.360.1 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

8.360.2 Field Documentation

8.360.2.1 BYTE messageWaitingInfoContent::activeInd

8.360.2.2 BYTE messageWaitingInfoContent::msgCount

8.360.2.3 BYTE messageWaitingInfoContent::msgType

8.361 minBasedIMSI Struct Reference

Data Fields

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

8.361.1 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

8.361.2 Field Documentation

8.361.2.1 WORD minBasedIMSI::imsiM1112

8.361.2.2 BYTE minBasedIMSI::imsiMS1[7]

8.361.2.3 BYTE minBasedIMSI::imsiMS2[3]

8.361.2.4 BYTE minBasedIMSI::mccM[3]

8.362 mitigationDevList Struct Reference

Data Fields

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

8.362.1 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.

8.362.2 Field Documentation

8.362.2.1 BYTE mitigationDevList::maxMitigationLevel

8.362.2.2 CHAR mitigationDevList::mitigationDevId[255]

8.362.2.3 BYTE mitigationDevList::mitigationDevIdLen

8.363 MNRInfo Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- ULONG rat

8.363.1 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

8.363.2 Field Documentation

8.363.2.1 WORD MNRInfo::mcc

8.363.2.2 WORD MNRInfo::mnc

8.363.2.3 ULONG MNRInfo::rat

8.364 ModifyProfileIn Struct Reference

Data Fields

- BYTE * pProfileID
- BYTE * pProfileType
- QmiProfileInfo curProfile

8.364.1 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

8.364.2 Field Documentation

8.364.2.1 QmiProfileInfo ModifyProfileIn::curProfile

8.364.2.2 BYTE* ModifyProfileIn::pProfileID

8.364.2.3 BYTE* ModifyProfileIn::pProfileType

8.365 ModifyProfileOut Struct Reference

Data Fields

- USHORT * pExtErrorCode

8.365.1 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.

8.365.2 Field Documentation

8.365.2.1 USHORT* ModifyProfileOut::pExtErrorCode

8.366 msgWaitingInfo Struct Reference

Data Fields

- BYTE numInstances
- messageWaitingInfoContent msgWaitInfo [0xFF]

8.366.1 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.

8.366.2 Field Documentation

8.366.2.1 messageWaitingInfoContent msgWaitingInfo::msgWaitInfo[0xFF]

8.366.2.2 BYTE msgWaitingInfo::numInstances

8.367 namName Struct Reference

Data Fields

- BYTE namNameLen
- BYTE namName [12]

8.367.1 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.

8.367.2 Field Documentation

8.367.2.1 BYTE namName::namName[12]

8.367.2.2 BYTE namName::namNameLen

8.368 nas_acqOrderPref Struct Reference

Data Fields

- uint8_t acqOrdeLen
- uint8_t * pAcqOrder

8.368.1 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

8.368.2 Field Documentation

8.368.2.1 uint8_t nas_acqOrderPref::acqOrdeLen

8.368.2.2 uint8_t* nas_acqOrderPref::pAcqOrder

8.369 nas_ActPilotPNElement Struct Reference

Data Fields

- uint16_t ActSetPilotPN
- uint8_t ActSetPilotPNStrength

8.369.1 Detailed Description

Parameters

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

8.369.2 Field Documentation

8.369.2.1 uint16_t nas_ActPilotPNElement::ActSetPilotPN

8.369.2.2 uint8_t nas_ActPilotPNElement::ActSetPilotPNStrength

8.370 nas_AddCDMASysInfo Struct Reference

Data Fields

- uint16_t geoSysIdx
- uint16_t regPrd

8.370.1 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.

8.370.2 Field Documentation

8.370.2.1 uint16_t nas_AddCDMASysInfo::geoSysIdx

8.370.2.2 uint16_t nas_AddCDMASysInfo::regPrd

8.371 nas_AddSysInfo Struct Reference

Data Fields

- uint16_t geoSysIdx
- uint32_t cellBroadcastCap

8.371.1 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

8.371.2 Field Documentation

8.371.2.1 uint32_t nas_AddSysInfo::cellBroadcastCap

8.371.2.2 uint16_t nas_AddSysInfo::geoSysIdx

8.372 nas_CallBarringSysInfo Struct Reference

Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

8.372.1 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

8.372.2 Field Documentation

8.372.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

8.372.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

8.373 nas_callBarStatus Struct Reference

Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

8.373.1 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

8.373.2 Field Documentation

8.373.2.1 uint32_t nas_callBarStatus::csBarStatus

8.373.2.2 uint32_t nas_callBarStatus::psBarStatus

8.374 nas_CDMAChannel Struct Reference

Data Fields

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

8.374.1 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

8.374.2 Field Documentation

8.374.2.1 uint16_t nas_CDMAChannel::priChA

8.374.2.2 uint16_t nas_CDMAChannel::priChB

8.374.2.3 uint16_t nas_CDMAChannel::secChA

8.374.2.4 uint16_t nas_CDMAChannel::secChB

8.375 nas_CDMAECIOThresh Struct Reference

Data Fields

- uint8_t CDMAECIOThreshListLen
- int16_t * pCDMAECIOThreshList

8.375.1 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)

8.375.2 Field Documentation

8.375.2.1 uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen

8.375.2.2 int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList

8.376 nas_CDMAInfo Struct Reference

Data Fields

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

8.376.1 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

8.376.2 Field Documentation

8.376.2.1 uint16_t nas_CDMAInfo::baseId

8.376.2.2 uint32_t nas_CDMAInfo::baseLat

8.376.2.3 uint32_t nas_CDMAInfo::baseLong

8.376.2.4 uint16_t nas_CDMAInfo::nid

8.376.2.5 uint16_t nas_CDMAInfo::refpn

8.376.2.6 uint16_t nas_CDMAInfo::sid

8.377 nas_CDMARSSIThresh Struct Reference

Data Fields

- uint8_t CDMARSSIThreshListLen
- int16_t * pCDMARSSIThreshList

8.377.1 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).

8.377.2 Field Documentation

8.377.2.1 uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen

8.377.2.2 int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList

8.378 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]

8.378.1 Detailed Description

Structure for storing the CDMA System Information.

Parameters

sysInfoCDMA	• See nas_sysInfoCommon for more information.
isSysPrlMatch-Valid	• Indicates whether the system PRL match is valid. – 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}.

8.378.2 Field Documentation

8.378.2.1 uint16_t nas_CDMASysInfo::baseId

8.378.2.2 uint32_t nas_CDMASysInfo::baseLat

8.378.2.3 uint32_t nas_CDMASysInfo::baseLong

8.378.2.4 uint8_t nas_CDMASysInfo::bsInfoValid

8.378.2.5 uint8_t nas_CDMASysInfo::bsPRev

8.378.2.6 uint8_t nas_CDMASysInfo::bsPRevValid

8.378.2.7 uint8_t nas_CDMASysInfo::ccsSupported

8.378.2.8 uint8_t nas_CDMASysInfo::ccsSupportedValid

8.378.2.9 uint8_t nas_CDMASysInfo::cdmaSysIdValid

8.378.2.10 uint8_t nas_CDMASysInfo::isSysPrlMatch

8.378.2.11 uint8_t nas_CDMASysInfo::isSysPrlMatchValid

8.378.2.12 uint8_t nas_CDMASysInfo::MCC[3]

8.378.2.13 uint8_t nas_CDMASysInfo::MNC[3]

8.378.2.14 uint16_t nas_CDMASysInfo::networkID

8.378.2.15 uint8_t nas_CDMASysInfo::networkIdValid

8.378.2.16 uint16_t nas_CDMASysInfo::packetZone

8.378.2.17 uint8_t nas_CDMASysInfo::packetZoneValid

8.378.2.18 uint8_t nas_CDMASysInfo::pRevInUse

8.378.2.19 uint8_t nas_CDMASysInfo::pRevInUseValid

8.378.2.20 nas_sysInfoCommon nas_CDMASysInfo::sysInfoCDMA

8.378.2.21 uint16_t nas_CDMASysInfo::systemID

8.379 nas_CDMASysInfoExt Struct Reference

Data Fields

- uint16_t MCC
- uint8_t imsi_11_12

8.379.1 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

8.379.2 Field Documentation

8.379.2.1 uint8_t nas_CDMASysInfoExt::imsi_11_12

8.379.2.2 uint16_t nas_CDMASysInfoExt::MCC

8.380 nas_cellParams Struct Reference

Data Fields

- uint16_t pci

- int16_t rsrq
- int16_t rsrp
- int16_t rssi
- int16_t srxlev

8.380.1 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.

8.380.2 Field Documentation

8.380.2.1 uint16_t nas_cellParams::pci

8.380.2.2 int16_t nas_cellParams::rsrp

8.380.2.3 int16_t nas_cellParams::rsrq

8.380.2.4 int16_t nas_cellParams::rssi

8.380.2.5 int16_t nas_cellParams::srxlev

8.381 nas_CommInfo Struct Reference

Data Fields

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

- uint8_t psState

8.381.1 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

8.381.2 Field Documentation

8.381.2.1 uint8_t nas_CommInfo::imsRegState

8.381.2.2 uint8_t nas_CommInfo::modemMode

8.381.2.3 uint8_t nas_CommInfo::psState

8.381.2.4 uint8_t nas_CommInfo::systemMode

8.381.2.5 int8_t nas_CommInfo::temperature

8.382 nas_CSGID Struct Reference

Data Fields

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

8.382.1 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

8.382.2 Field Documentation

8.382.2.1 uint32_t nas_CSGID::id

8.382.2.2 uint16_t nas_CSGID::mcc

8.382.2.3 uint16_t nas_CSGID::mnc

8.382.2.4 uint8_t nas_CSGID::mncPcsDigits

8.382.2.5 uint8_t nas_CSGID::rat

8.383 nas_currentPLMN Struct Reference

Data Fields

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

8.383.1 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

8.383.2 Field Documentation

8.383.2.1 uint16_t nas_currentPLMN::MCC

8.383.2.2 uint16_t nas_currentPLMN::MNC

8.383.2.3 uint8_t nas_currentPLMN::netDescr[255]

8.383.2.4 uint8_t nas_currentPLMN::netDescrLength

8.384 nas_dataSrvCapabilities Struct Reference

Data Fields

- uint8_t dataCapabilitiesLen
- uint8_t dataCapabilities [32]

8.384.1 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+

8.384.2 Field Documentation

8.384.2.1 uint8_t nas_dataSrvCapabilities::dataCapabilities[32]

8.384.2.2 uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen

8.385 nas_DataStatusDetail Struct Reference

Data Fields

- uint32_t IPAddress
- uint8_t LastErrCode

8.385.1 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

8.385.2 Field Documentation

8.385.2.1 uint32_t nas_DataStatusDetail::IPAddress

8.385.2.2 uint8_t nas_DataStatusDetail::LastErrCode

8.386 nas_detailSvcInfo Struct Reference

Data Fields

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

8.386.1 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

8.386.2 Field Documentation

8.386.2.1 uint8_t nas_detailSvcInfo::hdrHybrid

8.386.2.2 uint8_t nas_detailSvcInfo::hdrSrvStatus

8.386.2.3 uint8_t nas_detailSvcInfo::isSysForbidden

8.386.2.4 uint8_t nas_detailSvcInfo::srvCapability

8.386.2.5 uint8_t nas_detailSvcInfo::srvStatus

8.387 nas_DeviceConfigDetail Struct Reference

Data Fields

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

8.387.1 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

8.387.2 Field Documentation

8.387.2.1 uint8_t nas_DeviceConfigDetail::Chipset

8.387.2.2 uint8_t nas_DeviceConfigDetail::HWVersion

8.387.2.3 uint8_t nas_DeviceConfigDetail::QLIC

8.387.2.4 uint8_t nas_DeviceConfigDetail::Technology

8.388 nas_dirNum Struct Reference

Data Fields

- uint8_t dirNumLen
- uint8_t dirNum [255]

8.388.1 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.

8.388.2 Field Documentation

8.388.2.1 uint8_t nas_dirNum::dirNum[255]

8.388.2.2 uint8_t nas_dirNum::dirNumLen

8.389 nas_DRCParams Struct Reference

Data Fields

- uint8_t DRCValue
- uint8_t DRCCover

8.389.1 Detailed Description

Parameters

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

8.389.2 Field Documentation

8.389.2.1 uint8_t nas_DRCParams::DRCCover

8.389.2.2 uint8_t nas_DRCParams::DRCValue

8.390 nas_ecioListElement Struct Reference

Data Fields

- int16_t ecio
- uint8_t radioIf

8.390.1 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

8.390.2 Field Documentation

8.390.2.1 int16_t nas_ecioListElement::ecio

8.390.2.2 uint8_t nas_ecioListElement::radioIf

8.391 nas_ECIOThresh Struct Reference

Data Fields

- uint8_t ECIOThresListLen
- int16_t * pECIOThresList

8.391.1 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

8.391.2 Field Documentation

8.391.2.1 uint8_t nas_ECIOThresh::ECIOThresListLen

8.391.2.2 int16_t* nas_ECIOThresh::pECIOThresList

8.392 nas_errorRateListElement Struct Reference

Data Fields

- uint16_t errorRate
- uint8_t radioIf

8.392.1 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

8.392.2 Field Documentation

8.392.2.1 uint16_t nas_errorRateListElement::errorRate

8.392.2.2 uint8_t nas_errorRateListElement::radioIf

8.393 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]

8.393.1 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.

8.393.2 Field Documentation

8.393.2.1 uint16_t nas_GERANInfo::arfcn

8.393.2.2 uint8_t nas_GERANInfo::bsic

8.393.2.3 uint32_t nas_GERANInfo::cellID

8.393.2.4 nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]

8.393.2.5 uint16_t nas_GERANInfo::lac

8.393.2.6 uint8_t nas_GERANInfo::nmrInst

8.393.2.7 uint8_t nas_GERANInfo::plmn[3]

8.393.2.8 uint16_t nas_GERANInfo::rxLev

8.393.2.9 uint32_t nas_GERANInfo::timingAdvance

8.394 nas_geranInstInfo Struct Reference

Data Fields

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

8.394.1 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.

8.394.2 Field Documentation

8.394.2.1 uint16_t nas_geranInstInfo::geranArfcn

8.394.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

8.394.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

8.394.2.4 int16_t nas_geranInstInfo::geranRssi

8.395 nas_gsmCellInfo Struct Reference

Data Fields

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

8.395.1 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.

8.395.2 Field Documentation

8.395.2.1 uint16_t nas_gsmCellInfo::arfcn

8.395.2.2 uint8_t nas_gsmCellInfo::band1900

8.395.2.3 uint8_t nas_gsmCellInfo::bsicId

8.395.2.4 uint8_t nas_gsmCellInfo::cellIdValid

8.395.2.5 int16_t nas_gsmCellInfo::rssi

8.395.2.6 int16_t nas_gsmCellInfo::srxlev

8.396 nas_GSMRSSIThresh Struct Reference

Data Fields

- uint8_t GSMRSSIThreshListLen
- int16_t * pGSMRSSIThreshList

8.396.1 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)

8.396.2 Field Documentation

8.396.2.1 uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen

8.396.2.2 int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList

8.397 nas_GSMSrvStatusInfo Struct Reference

Data Fields

- uint8_t srvStatus
- uint8_t trueSrvStatus
- uint8_t isPrefDataPath

8.397.1 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

8.397.2 Field Documentation

8.397.2.1 uint8_t nas_GSMSrvStatusInfo::isPrefDataPath

8.397.2.2 uint8_t nas_GSMSrvStatusInfo::srvStatus

8.397.2.3 uint8_t nas_GSMSrvStatusInfo::trueSrvStatus

8.398 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

8.398.1 Detailed Description

Structure for storing the GSM System Information.

Parameters

sysInfoGSM	• See nas_sysInfoCommon for more information.
lacValid	• Indicates whether the location area code is valid.. – 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

8.398.2 Field Documentation

8.398.2.1 uint32_t nas_GSMSysInfo::cellId

8.398.2.2 uint8_t nas_GSMSysInfo::cellIdValid

8.398.2.3 uint8_t nas_GSMSysInfo::dtmSupp

8.398.2.4 uint8_t nas_GSMSysInfo::dtmSuppValid

8.398.2.5 uint8_t nas_GSMSysInfo::egprsSupp

8.398.2.6 uint8_t nas_GSMSysInfo::egprsSuppValid

8.398.2.7 uint16_t nas_GSMSysInfo::lac

8.398.2.8 uint8_t nas_GSMSysInfo::lacValid

8.398.2.9 uint8_t nas_GSMSysInfo::MCC[3]

8.398.2.10 uint8_t nas_GSMSysInfo::MNC[3]

8.398.2.11 uint8_t nas_GSMSysInfo::networkIdValid

8.398.2.12 uint8_t nas_GSMSysInfo::regRejectInfoValid

8.398.2.13 uint8_t nas_GSMSysInfo::rejCause

8.398.2.14 uint8_t nas_GSMSysInfo::rejectSrvDomain

8.398.2.15 nas_sysInfoCommon nas_GSMSysInfo::sysInfoGSM

8.399 nas_HDRECIOThresh Struct Reference

Data Fields

- uint8_t HDRECIOThreshListLen
- int16_t * pHDRECIOThreshList

8.399.1 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).

8.399.2 Field Documentation

8.399.2.1 uint8_t nas_HDRECIOThresh::HDRECIOThreshListLen

8.399.2.2 int16_t* nas_HDRECIOThresh::pHDRECIOThreshList

8.400 nas_HDRIOThresh Struct Reference

Data Fields

- uint8_t HDRIOThreshListLen
- int16_t * pHDRIOThreshList

8.400.1 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).

8.400.2 Field Documentation

8.400.2.1 uint8_t nas_HDRIOThresh::HDRIOThreshListLen

8.400.2.2 int16_t* nas_HDRIOThresh::pHDRIOThreshList

8.401 nas_HDRRSSIThresh Struct Reference

Data Fields

- uint8_t HDRRSSIThreshListLen
- int16_t * pHDRRSSIThreshList

8.401.1 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).

8.401.2 Field Documentation

8.401.2.1 uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen

8.401.2.2 int16_t* nas_HDRRSSIThresh::pHDRRSSIThreshList

8.402 nas_HDRSINRThresh Struct Reference

Data Fields

- uint8_t HDRSINRThresListLen
- uint8_t * pHDRSINRThresList

8.402.1 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

8.402.2 Field Documentation

8.402.2.1 uint8_t nas_HDRSINRThresh::HDRSINRThresListLen

8.402.2.2 uint8_t* nas_HDRSINRThresh::pHDRSINRThresList

8.403 nas_HDRSINRThreshold Struct Reference

Data Fields

- uint8_t HDRSINRThreshListLen
- uint16_t * pHDRSINRThreshList

8.403.1 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

8.403.2 Field Documentation

8.403.2.1 uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen

8.403.2.2 uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList

8.404 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]

8.404.1 Detailed Description

Structure for storing the HDR System Information.

Parameters

sysInfoHDR	• See nas_sysInfoCommon for more information.

isSysPrlMatch-Valid	• Indicates whether the system PRL match is valid. – 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.

8.404.2 Field Documentation

8.404.2.1 uint8_t nas_HDRSysInfo::hdrActiveProt

8.404.2.2 uint8_t nas_HDRSysInfo::hdrActiveProtValid

8.404.2.3 uint8_t nas_HDRSysInfo::hdrPersonality

8.404.2.4 uint8_t nas_HDRSysInfo::hdrPersonalityValid

8.404.2.5 uint8_t nas_HDRSysInfo::is856SysId[16]

8.404.2.6 uint8_t nas_HDRSysInfo::is856SysIdValid

8.404.2.7 uint8_t nas_HDRSysInfo::isSysPrlMatch

8.404.2.8 uint8_t nas_HDRSysInfo::isSysPrlMatchValid

8.404.2.9 nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR

8.405 nas_homeSIDNID Struct Reference

Data Fields

- uint8_t numInstances
- nas_sidNid SidNid [255]

8.405.1 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

8.405.2 Field Documentation

8.405.2.1 uint8_t nas_homeSIDNID::numInstances

8.405.2.2 nas_sidNid nas_homeSIDNID::SidNid[255]

8.406 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]

8.406.1 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.

8.406.2 Field Documentation

8.406.2.1 uint8_t nas_infoInterFreq::cell_resel_priority

8.406.2.2 nas_cellParams nas_infoInterFreq::cellInterFreqParams[255]

8.406.2.3 uint8_t nas_infoInterFreq::cells_len

8.406.2.4 uint16_t nas_infoInterFreq::earfcn

8.406.2.5 uint8_t nas_infoInterFreq::threshXHigh

8.406.2.6 uint8_t nas_infoInterFreq::threshXLow

8.407 nas_IOThresh Struct Reference

Data Fields

- uint8_t IOThresListLen
- int32_t * pIOThresList

8.407.1 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

8.407.2 Field Documentation

8.407.2.1 uint8_t nas_IOThresh::IOThresListLen

8.407.2.2 int32_t* nas_IOThresh::pIOThresList

8.408 nas_lteEARFCN Struct Reference

Data Fields

- uint8_t status
- uint32_t earfcn0
- uint32_t earfcn1

8.408.1 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.

8.408.2 Field Documentation

8.408.2.1 uint32_t nas_lteEARFCN::earfcn0

8.408.2.2 uint32_t nas_lteEARFCN::earfcn1

8.408.2.3 uint8_t nas_lteEARFCN::status

8.409 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]

8.409.1 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.

8.409.2 Field Documentation

8.409.2.1 uint8_t nas_lteGsmCellInfo::cellReselPriority

8.409.2.2 uint8_t nas_lteGsmCellInfo::cells_len

8.409.2.3 nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]

8.409.2.4 uint8_t nas_lteGsmCellInfo::nccPermitted

8.409.2.5 uint8_t nas_lteGsmCellInfo::threshGsmHigh

8.409.2.6 uint8_t nas_lteGsmCellInfo::threshGsmLow

8.410 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

8.410.1 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

8.410.2 Field Documentation

8.410.2.1 uint8_t nas_LTEInfo::band

8.410.2.2 uint8_t nas_LTEInfo::bandwidth

8.410.2.3 uint8_t nas_LTEInfo::emmConnState

8.410.2.4 uint8_t nas_LTEInfo::emmState

8.410.2.5 uint8_t nas_LTEInfo::emmSubState

8.410.2.6 uint16_t nas_LTEInfo::RXChan

8.410.2.7 uint16_t nas_LTEInfo::TXChan

8.411 nas_LTEInfoInterfreq Struct Reference

Data Fields

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

8.411.1 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.

8.411.2 Field Documentation

8.411.2.1 uint8_t nas_LTEInfoInterfreq::freqsLen

8.411.2.2 nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq[255]

8.411.2.3 uint8_t nas_LTEInfoInterfreq::ueInIdle

8.412 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]

8.412.1 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.

8.412.2 Field Documentation

8.412.2.1 nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]

8.412.2.2 uint8_t nas_LTEInfoIntrafreq::cellReselPriority

8.412.2.3 uint8_t nas_LTEInfoIntrafreq::cellsLen

8.412.2.4 uint16_t nas_LTEInfoIntrafreq::earfcn

8.412.2.5 uint32_t nas_LTEInfoIntrafreq::globalCellId

8.412.2.6 uint8_t nas_LTEInfoIntrafreq::plmn[3]

8.412.2.7 uint16_t nas_LTEInfoIntrafreq::servingCellId

8.412.2.8 uint8_t nas_LTEInfoIntrafreq::sIntraSearch

8.412.2.9 uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch

8.412.2.10 uint16_t nas_LTEInfoIntrafreq::tac

8.412.2.11 uint8_t nas_LTEInfoIntrafreq::threshServingLow

8.412.2.12 uint8_t nas_LTEInfoIntrafreq::ueInIdle

8.413 nas_LTEInfoNeighboringGSM Struct Reference

Data Fields

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

8.413.1 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.

8.413.2 Field Documentation

8.413.2.1 uint8_t nas_LTEInfoNeighboringGSM::freqsLen

8.413.2.2 nas_lteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.413.2.3 uint8_t nas_LTEInfoNeighboringGSM::ueInIdle

8.414 nas_LTEInfoNeighboringWCDMA Struct Reference

Data Fields

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

8.414.1 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.

8.414.2 Field Documentation

8.414.2.1 uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen

8.414.2.2 nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.414.2.3 uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle

8.415 nas_ltePCI Struct Reference

Data Fields

- uint8_t status
- uint32_t earfcn
- uint32_t pci

8.415.1 Detailed Description

Parameters

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

8.415.2 Field Documentation

8.415.2.1 uint32_t nas_ltePCI::earfcn

8.415.2.2 uint32_t nas_ltePCI::pci

8.415.2.3 uint8_t nas_ltePCI::status

8.416 nas_lteRsrpinformation Struct Reference

Data Fields

- int16_t rsrplevel

8.416.1 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).

8.416.2 Field Documentation

8.416.2.1 int16_t nas_lteRsrpinformation::rsrplevel

8.417 nas_LTERSRPThresh Struct Reference

Data Fields

- uint8_t LTERSRPThreshListLen
- int16_t * pLTERSRPThreshList

8.417.1 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).

8.417.2 Field Documentation

8.417.2.1 uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen

8.417.2.2 int16_t* nas_LTERSRPThresh::pLTERSRPThreshList

8.418 nas_LTERSRQThresh Struct Reference

Data Fields

- uint8_t LTERSRQThreshListLen
- int16_t * pLTERSRQThreshList

8.418.1 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)

8.418.2 Field Documentation

8.418.2.1 uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen

8.418.2.2 int16_t* nas_LTERSRQThresh::pLTERSRQThreshList

8.419 nas_LTERSSIThresh Struct Reference

Data Fields

- uint8_t LTERSSIThreshListLen
- int16_t * pLTERSSIThreshList

8.419.1 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)

8.419.2 Field Documentation

8.419.2.1 uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen

8.419.2.2 int16_t* nas_LTERSSIThresh::pLTERSSIThreshList

8.420 nas_LTESigRptCfg Struct Reference

Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

8.420.1 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

8.420.2 Field Documentation

8.420.2.1 uint8_t nas_LTESigRptCfg::avgPeriod

8.420.2.2 uint8_t nas_LTESigRptCfg::rptRate

8.421 nas_LTESigRptConfig Struct Reference

Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

8.421.1 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

8.421.2 Field Documentation

8.421.2.1 uint8_t nas_LTESigRptConfig::avgPeriod

8.421.2.2 uint8_t nas_LTESigRptConfig::rptRate

8.422 nas_lteSnrinformation Struct Reference

Data Fields

- int16_t snrlevel

8.422.1 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.

8.422.2 Field Documentation

8.422.2.1 int16_t nas_lteSnrinformation::snrlevel

8.423 nas_LTESNRThresh Struct Reference

Data Fields

- uint8_t LTESNRThresListLen
- int16_t * pLTESNRThresList

8.423.1 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

8.423.2 Field Documentation

8.423.2.1 uint8_t nas_LTESNRThresh::LTESNRThresListLen

8.423.2.2 int16_t* nas_LTESNRThresh::pLTESNRThresList

8.424 nas_LTESNRThreshold Struct Reference

Data Fields

- uint8_t LTESNRThreshListLen
- int16_t * pLTESNRThreshList

8.424.1 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).

8.424.2 Field Documentation

8.424.2.1 uint8_t nas_LTESNRThreshold::LTESNRThreshListLen

8.424.2.2 int16_t* nas_LTESNRThreshold::pLTESNRThreshList

8.425 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

8.425.1 Detailed Description

Structure for storing the LTE System Information.

Parameters

sysInfoLTE	• See nas_sysInfoCommon for more information.
lacValid	• Indicates whether the location area code is valid.. – 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

8.425.2 Field Documentation

8.425.2.1 uint32_t nas_LTESysInfo::cellId

8.425.2.2 uint8_t nas_LTESysInfo::cellIdValid

8.425.2.3 uint16_t nas_LTESysInfo::lac

8.425.2.4 uint8_t nas_LTESysInfo::lacValid

8.425.2.5 uint8_t nas_LTESysInfo::MCC[3]

8.425.2.6 uint8_t nas_LTESysInfo::MNC[3]

8.425.2.7 uint8_t nas_LTESysInfo::networkIdValid

8.425.2.8 uint8_t nas_LTESysInfo::regRejectInfoValid

8.425.2.9 uint8_t nas_LTESysInfo::rejCause

8.425.2.10 uint8_t nas_LTESysInfo::rejectSrvDomain

8.425.2.11 nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE

8.425.2.12 uint16_t nas_LTESysInfo::tac

8.425.2.13 uint8_t nas_LTESysInfo::tacValid

8.426 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]

8.426.1 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 nas_wcdmaCellInfo for more information.

8.426.2 Field Documentation

8.426.2.1 uint8_t nas_lteWcdmaCellInfo::cellReselPriority

8.426.2.2 uint8_t nas_lteWcdmaCellInfo::cellsLen

8.426.2.3 uint16_t nas_lteWcdmaCellInfo::threshXhigh

8.426.2.4 uint16_t nas_lteWcdmaCellInfo::threshXlow

8.426.2.5 uint16_t nas_lteWcdmaCellInfo::uarfcn

8.426.2.6 nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]

8.427 nas_minBasedIMSI Struct Reference

Data Fields

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

- uint8_t imsiMS2 [3]

8.427.1 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

8.427.2 Field Documentation

8.427.2.1 uint16_t nas_minBasedIMSI::imsiM1112

8.427.2.2 uint8_t nas_minBasedIMSI::imsiMS1[7]

8.427.2.3 uint8_t nas_minBasedIMSI::imsiMS2[3]

8.427.2.4 uint8_t nas_minBasedIMSI::mccM[3]

8.428 nas_MNRInfo Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- uint32_t rat

8.428.1 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

8.428.2 Field Documentation

8.428.2.1 uint16_t nas_MNRInfo::mcc

8.428.2.2 uint16_t nas_MNRInfo::mnc

8.428.2.3 uint32_t nas_MNRInfo::rat

8.429 nas_namName Struct Reference

Data Fields

- uint8_t namNameLen
- uint8_t namName [12]

8.429.1 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.

8.429.2 Field Documentation

8.429.2.1 uint8_t nas_namName::namName[12]

8.429.2.2 uint8_t nas_namName::namNameLen

8.430 nas_netSelectionPref Struct Reference

Data Fields

- uint8_t netReg
- uint16_t mcc
- uint16_t mnc

8.430.1 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

8.430.2 Field Documentation

8.430.2.1 uint16_t nas_netSelectionPref::mcc

8.430.2.2 uint16_t nas_netSelectionPref::mnc

8.430.2.3 uint8_t nas_netSelectionPref::netReg

8.431 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

8.431.1 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

8.431.2 Field Documentation

8.431.2.1 uint8_t nas_NetworkStat1x::ActSetCnt

8.431.2.2 uint8_t nas_NetworkStat1x::NeighborSetCnt

8.431.2.3 nas_ActPilotPNElement* nas_NetworkStat1x::pActPilotPNElements

8.431.2.4 uint16_t* nas_NetworkStat1x::pNeighborSetPilotPN

8.431.2.5 uint16_t nas_NetworkStat1x::RX_EC_IO

8.431.2.6 uint32_t nas_NetworkStat1x::RX_PWR

8.431.2.7 uint16_t nas_NetworkStat1x::SO

8.431.2.8 uint8_t nas_NetworkStat1x::State

8.431.2.9 uint32_t nas_NetworkStat1x::TX_PWR

8.432 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

8.432.1 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)

8.432.2 Field Documentation

8.432.2.1 uint8_t nas_NetworkStatEVDO::MACIndex

8.432.2.2 uint16_t nas_NetworkStatEVDO::PER

8.432.2.3 uint16_t nas_NetworkStatEVDO::PilotEnergy

8.432.2.4 uint16_t* nas_NetworkStatEVDO::pSectorID

8.432.2.5 uint16_t nas_NetworkStatEVDO::RX_PWR

8.432.2.6 uint8_t nas_NetworkStatEVDO::SectorIDLen

8.432.2.7 uint8_t nas_NetworkStatEVDO::SNR

8.432.2.8 uint8_t nas_NetworkStatEVDO::State

8.433 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

8.433.1 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_LENGT-H]	• 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

8.433.2 Field Documentation

8.433.2.1 uint16_t nas_nmrCellInfo::nmrArfcn

8.433.2.2 uint8_t nas_nmrCellInfo::nmrBsic

8.433.2.3 uint32_t nas_nmrCellInfo::nmrCellID

8.433.2.4 uint16_t nas_nmrCellInfo::nmrLac

8.433.2.5 uint8_t nas_nmrCellInfo::nmrPlmn[3]

8.433.2.6 uint16_t nas_nmrCellInfo::nmrRxLev

8.434 nas_operatorNameString Struct Reference

Data Fields

- uint8_t PLMNName [255]

8.434.1 Detailed Description

Parameters

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

8.434.2 Field Documentation

8.434.2.1 uint8_t nas_operatorNameString::PLMNName[255]

8.435 nas_OperatorPLMNData Struct Reference

Data Fields

- uint8_t mcc [3]
- uint8_t mnc [3]

- uint16_t lac1
- uint16_t lac2
- uint8_t PLMNRecID

8.435.1 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.

8.435.2 Field Documentation

8.435.2.1 uint16_t nas_OperatorPLMNData::lac1

8.435.2.2 uint16_t nas_OperatorPLMNData::lac2

8.435.2.3 uint8_t nas_OperatorPLMNData::mcc[3]

8.435.2.4 uint8_t nas_OperatorPLMNData::mnc[3]

8.435.2.5 uint8_t nas_OperatorPLMNData::PLMNRecID

8.436 nas_operatorPLMNList Struct Reference

Data Fields

- uint16_t numInstance
- nas_OperatorPLMNData PLMNData [255]

8.436.1 Detailed Description

Parameters

numInstance	• Number of sets of the elements.

PLMNData	• Refer OperatorPLMNLData for details (Optional).

8.436.2 Field Documentation

8.436.2.1 uint16_t nas_operatorPLMNList::numInstance

8.436.2.2 nas_OperatorPLMNData nas_operatorPLMNList::PLMNData[255]

8.437 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

8.437.1 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.

8.437.2 Field Documentation

8.437.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggPcellInfo::dl_bw_value

8.437.2.2 uint16_t nas_PhyCaAggPcellInfo::freq

8.437.2.3 uint16_t nas_PhyCaAggPcellInfo::iLTEbandValue

8.437.2.4 uint16_t nas_PhyCaAggPcellInfo::pci

8.437.2.5 uint8_t nas_PhyCaAggPcellInfo::TlvPresent

8.438 nas_PhyCaAggScellDlBw Struct Reference

Data Fields

- NAS_LTE_CPHY_CA_BW_NRB_LITE dl_bw_value
- uint8_t TlvPresent

8.438.1 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.

8.438.2 Field Documentation

8.438.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellDlBw::dl_bw_value

8.438.2.2 uint8_t nas_PhyCaAggScellDlBw::TlvPresent

8.439 nas_PhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

8.439.1 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.

8.439.2 Field Documentation

8.439.2.1 uint8_t nas_PhyCaAggScellIndex::scell_idx

8.439.2.2 uint8_t nas_PhyCaAggScellIndex::TlvPresent

8.440 nas_PhyCaAggScellIndType Struct Reference

Data Fields

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

8.440.1 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.

8.440.2 Field Documentation

8.440.2.1 uint16_t nas_PhyCaAggScellIndType::freq

8.440.2.2 uint16_t nas_PhyCaAggScellIndType::pci

8.440.2.3 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellIndType::scell_state

8.440.2.4 uint8_t nas_PhyCaAggScellIndType::TlvPresent

8.441 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

8.441.1 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.

8.441.2 Field Documentation

8.441.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellInfo::dl_bw_value

8.441.2.2 uint16_t nas_PhyCaAggScellInfo::freq

8.441.2.3 uint16_t nas_PhyCaAggScellInfo::iLTEbandValue

8.441.2.4 uint16_t nas_PhyCaAggScellInfo::pci

8.441.2.5 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellInfo::scell_state

8.441.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

8.442 nas_PilotSetData Struct Reference

Data Fields

- uint8_t NumPilots
- nas_PilotSetParams * pPilotSetInfo

8.442.1 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.

8.442.2 Field Documentation

8.442.2.1 uint8_t nas_PilotSetData::NumPilots

8.442.2.2 nas_PilotSetParams* nas_PilotSetData::pPilotSetInfo

8.443 nas_PilotSetParams Struct Reference

Data Fields

- uint32_t PilotType
- uint16_t PilotPN
- uint16_t PilotStrength

8.443.1 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)

8.443.2 Field Documentation

8.443.2.1 uint16_t nas_PilotSetParams::PilotPN

8.443.2.2 uint16_t nas_PilotSetParams::PilotStrength

8.443.2.3 uint32_t nas_PilotSetParams::PilotType

8.444 nas_PLMNNetworkName Struct Reference

Data Fields

- uint8_t numInstance
- nas_PLMNNetworkNameData PLMNNetName [255]

8.444.1 Detailed Description

Parameters

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

8.444.2 Field Documentation

8.444.2.1 uint8_t nas_PLMNNetworkName::numInstance

8.444.2.2 nas_PLMNNetworkNameData nas_PLMNNetworkName::PLMNNetName[255]

8.445 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]

8.445.1 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.

8.445.2 Field Documentation

8.445.2.1 uint8_t nas_PLMNNetworkNameData::codingScheme

8.445.2.2 uint8_t nas_PLMNNetworkNameData::countryInitials

8.445.2.3 uint8_t nas_PLMNNetworkNameData::longName[255]

8.445.2.4 uint8_t nas_PLMNNetworkNameData::longNameLen

8.445.2.5 uint8_t nas_PLMNNetworkNameData::longNameSpareBits

8.445.2.6 uint8_t nas_PLMNNetworkNameData::shortName[255]

8.445.2.7 uint8_t nas_PLMNNetworkNameData::shortNameLen

8.445.2.8 uint8_t nas_PLMNNetworkNameData::shortNameSpareBits

8.446 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

8.446.1 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

8.446.2 Field Documentation

8.446.2.1 uint16_t nas_protocolSubtypeElement::AccessMac

8.446.2.2 uint16_t nas_protocolSubtypeElement::AuthProt

8.446.2.3 uint16_t nas_protocolSubtypeElement::ControlMac

8.446.2.4 uint16_t nas_protocolSubtypeElement::EncryptProt

8.446.2.5 uint16_t nas_protocolSubtypeElement::ForwardMac

8.446.2.6 uint16_t nas_protocolSubtypeElement::IdleState

8.446.2.7 uint16_t nas_protocolSubtypeElement::KeyExchange

8.446.2.8 uint16_t nas_protocolSubtypeElement::MultDisc

8.446.2.9 uint16_t nas_protocolSubtypeElement::PhysicalLayer

8.446.2.10 uint16_t nas_protocolSubtypeElement::ReverseMac

8.446.2.11 uint16_t nas_protocolSubtypeElement::SecProt

8.446.2.12 uint16_t nas_protocolSubtypeElement::VirtStream

8.447 nas_qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- uint8_t leapSeconds
- uint8_t localTimeOffset
- uint8_t daylightSavings

8.447.1 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)

8.447.2 Field Documentation

8.447.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

8.447.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

8.447.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

8.448 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]

8.448.1 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.

8.448.2 Field Documentation

8.448.2.1 char nas_QmiNas3GppNetworkInfo::Desription[255]

8.448.2.2 uint32_t nas_QmiNas3GppNetworkInfo::Forbidden

8.448.2.3 uint32_t nas_QmiNas3GppNetworkInfo::InUse

8.448.2.4 uint16_t nas_QmiNas3GppNetworkInfo::MCC

8.448.2.5 uint16_t nas_QmiNas3GppNetworkInfo::MNC

8.448.2.6 uint32_t nas_QmiNas3GppNetworkInfo::Preferred

8.448.2.7 uint32_t nas_QmiNas3GppNetworkInfo::Roaming

8.449 nas_QmiNas3GppNetworkRAT Struct Reference

Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t RAT

8.449.1 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

8.449.2 Field Documentation

8.449.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

8.449.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

8.449.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

8.450 nas_QmisNasPcsDigit Struct Reference

Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t includes_pcs_digit

8.450.1 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

8.450.2 Field Documentation

8.450.2.1 uint8_t nas_QmisNasPcsDigit::includes_pcs_digit

8.450.2.2 uint16_t nas_QmisNasPcsDigit::MCC

8.450.2.3 uint16_t nas_QmisNasPcsDigit::MNC

8.451 nas_RejectReasonTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t serviceDomain
- uint32_t rejectCause

8.451.1 Detailed Description

Parameters

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

8.451.2 Field Documentation

8.451.2.1 uint32_t nas_RejectReasonTlv::rejectCause

8.451.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

8.451.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

8.452 nas_RFInfoTlv Struct Reference

Data Fields

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

8.452.1 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

8.452.2 Field Documentation

8.452.2.1 uint32_t nas_RFInfoTlv::activeBandClass[255]

8.452.2.2 uint32_t nas_RFInfoTlv::activeChannel[255]

8.452.2.3 uint32_t nas_RFInfoTlv::radioInterface[255]

8.452.2.4 uint8_t nas_RFInfoTlv::radioInterfaceSize

8.452.2.5 uint8_t nas_RFInfoTlv::TlvPresent

8.453 nas_roamIndList Struct Reference

Data Fields

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

8.453.1 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

8.453.2 Field Documentation

8.453.2.1 uint8_t nas_roamIndList::numInstances

8.453.2.2 uint8_t nas_roamIndList::radioInterface[32]

8.453.2.3 uint8_t nas_roamIndList::roamIndicator[32]

8.454 nas_RSRPThresh Struct Reference

Data Fields

- uint8_t RSRPThresListLen
- int16_t * pRSRPThresList

8.454.1 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

8.454.2 Field Documentation

8.454.2.1 int16_t* nas_RSRPThresh::pRSRPThresList

8.454.2.2 uint8_t nas_RSRPThresh::RSRPThresListLen

8.455 nas_rsrqInformation Struct Reference

Data Fields

- int8_t rsrq
- uint8_t radioIf

8.455.1 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

8.455.2 Field Documentation

8.455.2.1 uint8_t nas_rsrqInformation::radioIf

8.455.2.2 int8_t nas_rsrqInformation::rsrq

8.456 nas_RSRQThresh Struct Reference

Data Fields

- uint8_t RSRQThresListLen
- int8_t * pRSRQThresList

8.456.1 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

8.456.2 Field Documentation

8.456.2.1 int8_t* nas_RSRQThresh::pRSRQThresList

8.456.2.2 uint8_t nas_RSRQThresh::RSRQThresListLen

8.457 nas_RSSIThresh Struct Reference

Data Fields

- uint8_t RSSIThresListLen
- int8_t * pRSSIThresList

8.457.1 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.

8.457.2 Field Documentation

8.457.2.1 int8_t* nas_RSSIThresh::pRSSIThresList

8.457.2.2 uint8_t nas_RSSIThresh::RSSIThresListLen

8.458 nas_rxInfo Struct Reference

Data Fields

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

8.458.1 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.

8.458.2 Field Documentation

8.458.2.1 int32_t nas_rxInfo::ecio

8.458.2.2 uint8_t nas_rxInfo::isRadioTuned

8.458.2.3 uint32_t nas_rxInfo::phase

8.458.2.4 int32_t nas_rxInfo::rscp

8.458.2.5 int32_t nas_rxInfo::rsrp

8.458.2.6 int32_t nas_rxInfo::rxPower

8.459 nas_RxSigInfo Struct Reference

Data Fields

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

8.459.1 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

8.459.2 Field Documentation

8.459.2.1 uint8_t nas_RxSigInfo::isRadioTuned

8.459.2.2 int32_t nas_RxSigInfo::rsrp

8.459.2.3 uint8_t nas_RxSigInfo::rxChainIndex

8.459.2.4 int32_t nas_RxSigInfo::rxPower

8.460 nas_rxSignalStrengthListElement Struct Reference

Data Fields

- int16_t rxSignalStrength
- uint8_t radioIf

8.460.1 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.

8.460.2 Field Documentation

8.460.2.1 uint8_t nas_rxSignalStrengthListElement::radioIf

8.460.2.2 int16_t nas_rxSignalStrengthListElement::rxSignalStrength

8.461 nas_SccRxInfo Struct Reference

Data Fields

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

8.461.1 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.

8.461.2 Field Documentation

8.461.2.1 uint8_t nas_SccRxInfo::numInstances

8.461.2.2 int32_t nas_SccRxInfo::rsrq

8.461.2.3 nas_RxSigInfo nas_SccRxInfo::sigInfo[255]

8.461.2.4 int16_t nas_SccRxInfo::snr

8.461.2.5 uint8_t nas_SccRxInfo::TlvPresent

8.462 nas_serviceProviderName Struct Reference

Data Fields

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

8.462.1 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.

8.462.2 Field Documentation

8.462.2.1 uint8_t nas_serviceProviderName::displayCondition

8.462.2.2 uint8_t nas_serviceProviderName::spn[255]

8.462.2.3 uint8_t nas_serviceProviderName::spnLength

8.463 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]

8.463.1 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

8.463.2 Field Documentation

8.463.2.1 uint8_t nas_servSystem::csAttachState

8.463.2.2 uint8_t nas_servSystem::numRadioInterfaces

8.463.2.3 uint8_t nas_servSystem::psAttachState

8.463.2.4 uint8_t nas_servSystem::radioInterface[32]

8.463.2.5 uint8_t nas_servSystem::regState

8.463.2.6 uint8_t nas_servSystem::selNetwork

8.464 nas_sidNid Struct Reference

Data Fields

- uint16_t nid
- uint16_t sid

8.464.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

nid	• Network ID
sid	• System ID

8.464.2 Field Documentation

8.464.2.1 uint16_t nas_sidNid::nid

8.464.2.2 uint16_t nas_sidNid::sid

8.465 nas_SignalStrengthTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- int8_t signalStrength
- uint32_t radioInterface

8.465.1 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

8.465.2 Field Documentation

8.465.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

8.465.2.2 int8_t nas_SignalStrengthTlv::signalStrength

8.465.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

8.466 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

8.466.1 Detailed Description

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

Parameters

rxSignalStrengthDelta	• 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.
ecioThresholdListLen	• Number of elements in the ECIO threshold list.
ecioThresholdList	• 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.
sinrThresholdListLen	• 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.

8.466.2 Field Documentation

8.466.2.1 uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta

8.466.2.2 int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.466.2.3 uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.466.2.4 uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta

8.466.2.5 uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta

8.466.2.6 uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta

8.466.2.7 uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta

8.466.2.8 uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.466.2.9 uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta

8.466.2.10 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.466.2.11 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.467 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

8.467.1 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

8.467.2 Field Documentation

8.467.2.1 nas_ecioListElement nas_SLQSSignalStrengthsInformation::ecioInfo

8.467.2.2 nas_errorRateListElement nas_SLQSSignalStrengthsInformation::errorRateInfo

8.467.2.3 uint32_t nas_SLQSSignalStrengthsInformation::io

8.467.2.4 nas_lteRsrpinformation nas_SLQSSignalStrengthsInformation::lteRsrpinfo

8.467.2.5 nas_lteSnrinformation nas_SLQSSignalStrengthsInformation::lteSnrinfo

8.467.2.6 nas_rsrqInformation nas_SLQSSignalStrengthsInformation::rsrqInfo

8.467.2.7 nas_rxSignalStrengthListElement nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.467.2.8 uint8_t nas_SLQSSignalStrengthsInformation::sinr

8.468 nas_SLQSSignalStrengthsTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- nas_SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo

8.468.1 Detailed Description

Parameters

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

8.468.2 Field Documentation

8.468.2.1 nas_SLQSSignalStrengthsInformation nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo

8.468.2.2 uint8_t nas_SLQSSignalStrengthsTlv::TlvPresent

8.469 nas_SrvStatusInfo Struct Reference

Data Fields

- uint8_t srvStatus
- uint8_t isPrefDataPath

8.469.1 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

8.469.2 Field Documentation

8.469.2.1 uint8_t nas_SrvStatusInfo::isPrefDataPath

8.469.2.2 uint8_t nas_SrvStatusInfo::srvStatus

8.470 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

8.470.1 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

8.470.2 Field Documentation

8.470.2.1 uint8_t nas_sysInfoCommon::isSysForbidden

8.470.2.2 uint8_t nas_sysInfoCommon::isSysForbiddenValid

8.470.2.3 uint8_t nas_sysInfoCommon::roamStatus

8.470.2.4 uint8_t nas_sysInfoCommon::roamStatusValid

8.470.2.5 uint8_t nas_sysInfoCommon::srvCapability

8.470.2.6 uint8_t nas_sysInfoCommon::srvCapabilityValid

8.470.2.7 uint8_t nas_sysInfoCommon::srvDomain

8.470.2.8 uint8_t nas_sysInfoCommon::srvDomainValid

8.471 nas_TDSCDMAECIOThresh Struct Reference

Data Fields

- uint8_t TDSCDMAECIOThreshListLen
- float * pTDSCDMAECIOThreshList

8.471.1 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.

8.471.2 Field Documentation

8.471.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.471.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.472 nas_TDSCDMARSCPThresh Struct Reference

Data Fields

- uint8_t TDSCDMARSCPThreshListLen
- int16_t * pTDSCDMARSCPThreshList

8.472.1 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).

8.472.2 Field Documentation

8.472.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.472.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.473 nas_TDSCDMARSSIThresh Struct Reference

Data Fields

- uint8_t TDSCDMARSSIThreshListLen
- float * pTDSCDMARSSIThreshList

8.473.1 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.

8.473.2 Field Documentation

8.473.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.473.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.474 nas_TDSCDMASINRCONFThresh Struct Reference

Data Fields

- uint8_t TDSCDMASINRCONFThreshListLen
- float * pTDSCDMASINRCONFThreshList

8.474.1 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

8.474.2 Field Documentation

8.474.2.1 float* nas_TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

8.474.2.2 uint8_t nas_TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

8.475 nas_TDSCDMASINRThresh Struct Reference

Data Fields

- uint8_t TDSCDMASINRThreshListLen
- float * pTDSCDMASINRThreshList

8.475.1 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

8.475.2 Field Documentation

8.475.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.475.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.476 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

8.476.1 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.

8.476.2 Field Documentation

8.476.2.1 uint8_t nas_timeInfo::day

8.476.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

8.476.2.3 uint8_t nas_timeInfo::dayOfWeek

8.476.2.4 uint8_t nas_timeInfo::hour

8.476.2.5 uint8_t nas_timeInfo::minute

8.476.2.6 uint8_t nas_timeInfo::month

8.476.2.7 uint8_t nas_timeInfo::radioInterface

8.476.2.8 uint8_t nas_timeInfo::second

8.476.2.9 int8_t nas_timeInfo::timeZone

8.476.2.10 uint8_t nas_timeInfo::TlvPresent

8.476.2.11 uint16_t nas_timeInfo::year

8.477 nas_trueIMSI Struct Reference

Data Fields

- uint8_t mccT [3]
- uint16_t imsiT1112
- uint8_t imsiTS1 [7]
- uint8_t imsiTS2 [3]
- uint8_t imsiTaddrNum

8.477.1 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

8.477.2 Field Documentation

8.477.2.1 uint16_t nas_truelMSI::imsiT1112

8.477.2.2 uint8_t nas_truelMSI::imsiTaddrNum

8.477.2.3 uint8_t nas_truelMSI::imsiTS1[7]

8.477.2.4 uint8_t nas_truelMSI::imsiTS2[3]

8.477.2.5 uint8_t nas_truelMSI::mccT[3]

8.478 nas_txInfo Struct Reference

Data Fields

- uint8_t isInTraffic
- int32_t txPower

8.478.1 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.

8.478.2 Field Documentation

8.478.2.1 uint8_t nas_txInfo::isInTraffic

8.478.2.2 int32_t nas_txInfo::txPower

8.479 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]

8.479.1 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.

8.479.2 Field Documentation

8.479.2.1 uint16_t nas_UMTSInfo::cellID

8.479.2.2 int16_t nas_UMTSInfo::ecio

8.479.2.3 uint8_t nas_UMTSInfo::geranInst

8.479.2.4 nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]

8.479.2.5 uint16_t nas_UMTSInfo::lac

8.479.2.6 uint8_t nas_UMTSInfo::plmn[3]

8.479.2.7 uint16_t nas_UMTSInfo::psc

8.479.2.8 int16_t nas_UMTSInfo::rscp

8.479.2.9 uint16_t nas_UMTSInfo::uarfcn

8.479.2.10 uint8_t nas_UMTSInfo::umtsInst

8.479.2.11 nas_UMTSinstInfo nas_UMTSInfo::UMTSInstInfo[255]

8.480 nas_UMTSinstInfo Struct Reference

Data Fields

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

8.480.1 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).

8.480.2 Field Documentation

8.480.2.1 int16_t nas_UMTSinstInfo::umtsEcio

8.480.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

8.480.2.3 int16_t nas_UMTSinstInfo::umtsRscp

8.480.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

8.481 nas_umtsLTENbrCell Struct Reference

Data Fields

- uint16_t earfcn
- uint16_t pci
- uint32_t rsrp
- uint32_t rsrq
- int16_t srxlev
- uint8_t cellIsTDD

8.481.1 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.

8.481.2 Field Documentation

8.481.2.1 uint8_t nas_umtsLTENbrCell::cellIsTDD

8.481.2.2 uint16_t nas_umtsLTENbrCell::earfcn

8.481.2.3 uint16_t nas_umtsLTENbrCell::pci

8.481.2.4 uint32_t nas_umtsLTENbrCell::rsrp

8.481.2.5 uint32_t nas_umtsLTENbrCell::rsrq

8.481.2.6 int16_t nas_umtsLTENbrCell::srxlev

8.482 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

8.482.1 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.

8.482.2 Field Documentation

8.482.2.1 uint8_t nas_UniversalTime::day

8.482.2.2 uint8_t nas_UniversalTime::dayOfWeek

8.482.2.3 uint8_t nas_UniversalTime::hour

8.482.2.4 uint8_t nas_UniversalTime::minute

8.482.2.5 uint8_t nas_UniversalTime::month

8.482.2.6 uint8_t nas_UniversalTime::second

8.482.2.7 uint16_t nas_UniversalTime::year

8.483 nas_wcdmaCellInfo Struct Reference

Data Fields

- uint16_t psc
- int16_t cpich_rscp
- int16_t cpich_ecno
- int16_t srxlev

8.483.1 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.

8.483.2 Field Documentation

8.483.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

8.483.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

8.483.2.3 uint16_t nas_wcdmaCellInfo::psc

8.483.2.4 int16_t nas_wcdmaCellInfo::srxlev

8.484 nas_WCDMAECIOThresh Struct Reference

Data Fields

- uint8_t WCDMAECIOThreshListLen
- int16_t * pWCDMAECIOThreshList

8.484.1 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)

8.484.2 Field Documentation

8.484.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

8.484.2.2 uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen

8.485 nas_WCDMAInfoLTENeighborCell Struct Reference

Data Fields

- uint32_t wcdmaRRCState
- uint8_t umtsLTENbrCellLen
- nas_umtsLTENbrCell UMTSLTENbrCell [255]

8.485.1 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.

8.485.2 Field Documentation

8.485.2.1 nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.485.2.2 uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.485.2.3 uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCState

8.486 nas_WCDMARSSIThresh Struct Reference

Data Fields

- uint8_t WCDMARSSIThreshListLen
- int16_t * pWCDMARSSIThreshList

8.486.1 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)

8.486.2 Field Documentation

8.486.2.1 int16_t* nas_WCDMARSSIThresh::pWCDMARSSIThreshList

8.486.2.2 uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen

8.487 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

8.487.1 Detailed Description

Structure for storing the WCDMA System Information.

Parameters

sysInfoWCDMA	• See nas_sysInfoCommon for more information.
lacValid	• Indicates whether the location area code is valid.. – 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.
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

8.487.2 Field Documentation

8.487.2.1 uint32_t nas_WCDMASysInfo::cellId

8.487.2.2 uint8_t nas_WCDMASysInfo::cellIdValid

8.487.2.3 uint8_t nas_WCDMASysInfo::hsCallStatus

8.487.2.4 uint8_t nas_WCDMASysInfo::hsCallStatusValid

8.487.2.5 uint8_t nas_WCDMASysInfo::hsInd

8.487.2.6 uint8_t nas_WCDMASysInfo::hsIndValid

8.487.2.7 uint16_t nas_WCDMASysInfo::lac

8.487.2.8 uint8_t nas_WCDMASysInfo::lacValid

8.487.2.9 uint8_t nas_WCDMASysInfo::MCC[3]

8.487.2.10 uint8_t nas_WCDMASysInfo::MNC[3]

8.487.2.11 uint8_t nas_WCDMASysInfo::networkIdValid

8.487.2.12 uint16_t nas_WCDMASysInfo::psc

8.487.2.13 uint8_t nas_WCDMASysInfo::pscValid

8.487.2.14 uint8_t nas_WCDMASysInfo::regRejectInfoValid

8.487.2.15 uint8_t nas_WCDMASysInfo::rejCause

8.487.2.16 uint8_t nas_WCDMASysInfo::rejectSrvDomain

8.487.2.17 nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA

8.488 nas_wcdmaUARFCN Struct Reference

Data Fields

- uint8_t status
- uint32_t uarfcn

8.488.1 Detailed Description

Parameters

status	• 0 - Disable • 1 - Enable
uarfcn	• UARFCN to which UE is locked

8.488.2 Field Documentation

8.488.2.1 uint8_t nas_wcdmaUARFCN::status

8.488.2.2 uint32_t nas_wcdmaUARFCN::uarfcn

8.489 NASBandPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t band_pref

8.489.1 Detailed Description

Structure used to store all Band Preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
band_pref	- Roaming Indication Value

8.489.2 Field Documentation

8.489.2.1 uint64_t NASBandPreferenceTlv::band_pref

8.489.2.2 uint8_t NASBandPreferenceTlv::TlvPresent

8.490 nasCellLocationInfoResp Struct Reference

Data Fields

- GERANInfo * pGERANInfo
- UMTSInfo * pUMTSInfo
- CDMAInfo * pCDMAInfo
- LTEInfoIntrafreq * pLTEInfoIntrafreq
- LTEInfoInterfreq * pLTEInfoInterfreq
- LTEInfoNeighboringGSM * pLTEInfoNeighboringGSM
- LTEInfoNeighboringWCDMA * pLTEInfoNeighboringWCDMA
- ULONG * pUMTSCellID
- WCDMAInfoLTENeighborCell * pWCDMAInfoLTENeighborCell

8.490.1 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.

8.490.2 Field Documentation

8.490.2.1 CDMAInfo∗ nasCellLocationInfoResp::pCDMAInfo

8.490.2.2 GERANInfo∗ nasCellLocationInfoResp::pGERANInfo

8.490.2.3 LTEInfoInterfreq∗ nasCellLocationInfoResp::pLTEInfoInterfreq

8.490.2.4 LTEInfoIntrafreq∗ nasCellLocationInfoResp::pLTEInfoIntrafreq

8.490.2.5 LTEInfoNeighboringGSM∗ nasCellLocationInfoResp::pLTEInfoNeighboringGSM

8.490.2.6 LTEInfoNeighboringWCDMA∗ nasCellLocationInfoResp::pLTEInfoNeighboringWCDMA

8.490.2.7 ULONG∗ nasCellLocationInfoResp::pUMTSCellID

8.490.2.8 UMTSInfo∗ nasCellLocationInfoResp::pUMTSInfo

8.490.2.9 WCDMAInfoLTENeighborCell∗ nasCellLocationInfoResp::pWCDMAInfoLTENeighborCell

8.491 NASEmergencyModeTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t EmerMode

8.491.1 Detailed Description

Structure used to store Emergency Mode TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
EmerMode	- Emergency Mode Value

8.491.2 Field Documentation

8.491.2.1 uint8_t NASEmergencyModeTlv::EmerMode

8.491.2.2 uint8_t NASEmergencyModeTlv::TlvPresent

8.492 nasGet3GPP2SubscriptionInfoReq Struct Reference

Data Fields

- BYTE namID

8.492.1 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.

8.492.2 Field Documentation

8.492.2.1 BYTE nasGet3GPP2SubscriptionInfoReq::namID

8.493 nasGet3GPP2SubscriptionInfoResp Struct Reference

Data Fields

- namName * pNAMNameInfo
- dirNum * pDirNum
- homeSIDNID * pHomeSIDNID
- minBasedIMSI * pMinBasedIMSI
- trueIMSI * pTrueIMSI
- CDMAChannel * pCDMAChannel

8.493.1 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

8.493.2 Field Documentation

8.493.2.1 CDMAChannel∗ nasGet3GPP2SubscriptionInfoResp::pCDMAChannel

8.493.2.2 dirNum∗ nasGet3GPP2SubscriptionInfoResp::pDirNum

8.493.2.3 homeSIDNID∗ nasGet3GPP2SubscriptionInfoResp::pHomeSIDNID

8.493.2.4 minBasedIMSI∗ nasGet3GPP2SubscriptionInfoResp::pMinBasedIMSI

8.493.2.5 namName∗ nasGet3GPP2SubscriptionInfoResp::pNAMNameInfo

8.493.2.6 trueIMSI∗ nasGet3GPP2SubscriptionInfoResp::pTrueIMSI

8.494 nasGetHDRColorCodeResp Struct Reference

Data Fields

- BYTE ∗ pColorCode

8.494.1 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

8.494.2 Field Documentation

8.494.2.1 BYTE∗ nasGetHDRColorCodeResp::pColorCode

8.495 nasGetLTECphyCa Struct Reference

Data Fields

- PhyCaAggScellIndType sPhyCaAggScellIndType
- PhyCaAggScellDlBw sPhyCaAggScellDlBw
- PhyCaAggScellInfo sPhyCaAggScellInfo
- PhyCaAggPcellInfo sPhyCaAggPcellInfo
- PhyCaAggScellIndex sPhyCaAggScellIndex

8.495.1 Field Documentation

8.495.1.1 PhyCaAggPcellInfo nasGetLTECphyCa::sPhyCaAggPcellInfo

8.495.1.2 PhyCaAggScellDlBw nasGetLTECphyCa::sPhyCaAggScellDlBw

8.495.1.3 PhyCaAggScellIndex nasGetLTECphyCa::sPhyCaAggScellIndex

8.495.1.4 PhyCaAggScellIndType nasGetLTECphyCa::sPhyCaAggScellIndType

8.495.1.5 PhyCaAggScellInfo nasGetLTECphyCa::sPhyCaAggScellInfo

8.496 NasGetLTECphyCaInfo Struct Reference

Data Fields

- NASPhyCaAggScellIndType PhyCaAggScellIndType
- NASPhyCaAggScellDlBw PhyCaAggScellDlBw
- NASPhyCaAggScellInfo PhyCaAggScellInfo
- NASPhyCaAggPcellInfo PhyCaAggPcellInfo
- NASPhyCaAggScellIndex PhyCaAggScellIndex

8.496.1 Field Documentation

8.496.1.1 NASPhyCaAggPcellInfo NasGetLTECphyCaInfo::PhyCaAggPcellInfo

8.496.1.2 NASPhyCaAggScellDlBw NasGetLTECphyCaInfo::PhyCaAggScellDlBw

8.496.1.3 NASPhyCaAggScellIndex NasGetLTECphyCaInfo::PhyCaAggScellIndex

8.496.1.4 NASPhyCaAggScellIndType NasGetLTECphyCaInfo::PhyCaAggScellIndType

8.496.1.5 NASPhyCaAggScellInfo NasGetLTECphyCaInfo::PhyCaAggScellInfo

8.497 nasGetLTECphyCaResp Struct Reference

Data Fields

- PhyCaAggScellIndType * pPhyCaAggScellIndType
- PhyCaAggScellDlBw * pPhyCaAggScellDlBw
- PhyCaAggScellInfo * pPhyCaAggScellInfo
- PhyCaAggPcellInfo * pPhyCaAggPcellInfo
- PhyCaAggScellIndex * pPhyCaAggScellIndex

8.497.1 Field Documentation

8.497.1.1 PhyCaAggPcellInfo∗ nasGetLTECphyCaResp::pPhyCaAggPcellInfo

8.497.1.2 PhyCaAggScellDlBw∗ nasGetLTECphyCaResp::pPhyCaAggScellDlBw

8.497.1.3 PhyCaAggScellIndex∗ nasGetLTECphyCaResp::pPhyCaAggScellIndex

8.497.1.4 PhyCaAggScellIndType∗ nasGetLTECphyCaResp::pPhyCaAggScellIndType

8.497.1.5 PhyCaAggScellInfo∗ nasGetLTECphyCaResp::pPhyCaAggScellInfo

8.498 nasGetSigInfoResp Struct Reference

Data Fields

- CDMASSInfo ∗ pCDMASSInfo
- HDRSSInfo ∗ pHDRSSInfo
- INT8 ∗ pGSMSSInfo
- CDMASSInfo ∗ pWCDMASSInfo
- LTESSInfo ∗ pLTESSInfo
- INT8 ∗ pTDSCDMASigInfoRscp
- TDSCDMASigInfoExt ∗ pTDSCDMASigInfoExt

8.498.1 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.

8.498.2 Field Documentation

8.498.2.1 CDMASSInfo* nasGetSigInfoResp::pCDMASSInfo

8.498.2.2 INT8* nasGetSigInfoResp::pGSMSSInfo

8.498.2.3 HDRSSInfo* nasGetSigInfoResp::pHDRSSInfo

8.498.2.4 LTESSInfo* nasGetSigInfoResp::pLTESSInfo

8.498.2.5 TDSCDMASigInfoExt* nasGetSigInfoResp::pTDSCDMASigInfoExt

8.498.2.6 INT8* nasGetSigInfoResp::pTDSCDMASigInfoRscp

8.498.2.7 CDMASSInfo* nasGetSigInfoResp::pWCDMASSInfo

8.499 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

8.499.1 Detailed Description

Structure for storing the SLQSNasGetSysInfo response parameters.

Parameters

pCDMASrv-StatusInfo	• See SrvStatusInfo for more information.
pHDRSrvStatus-Info	• See SrvStatusInfo for more information.

pGSMSrvStatus-Info	• See GSMSrvStatusInfo for more information.
pWCDMASrv-StatusInfo	• See GSMSrvStatusInfo for more information.
pLTESrvStatus-Info	• See GSMSrvStatusInfo for more information.
pCDMASysInfo	• See CDMASysInfo for more information.
pHDRSysInfo	• See HDRSysInfo for more information.
pGSMSysInfo	• See GSMSysInfo for more information.
pWCDMASys-Info	• See WCDMASysInfo for more information.
pLTESysInfo	• See LTESysInfo for more information.
pAddCDMASys-Info	• See AddCDMASysInfo for more information.
pAddHDRSys-Info	• 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.
pAddGSMSys-Info	• See AddSysInfo for more information.
pAddWCDMA-SysInfo	• 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

8.499.2 Field Documentation

8.499.2.1 AddCDMASysInfo∗ nasGetSysInfoResp::pAddCDMASysInfo

8.499.2.2 AddSysInfo∗ nasGetSysInfoResp::pAddGSMSysInfo

8.499.2.3 WORD∗ nasGetSysInfoResp::pAddHDRSysInfo

8.499.2.4 WORD∗ nasGetSysInfoResp::pAddLTESysInfo

8.499.2.5 AddSysInfo∗ nasGetSysInfoResp::pAddWCDMASysInfo

8.499.2.6 SrvStatusInfo∗ nasGetSysInfoResp::pCDMASrvStatusInfo

8.499.2.7 CDMASysInfo∗ nasGetSysInfoResp::pCDMASysInfo

8.499.2.8 CallBarringSysInfo∗ nasGetSysInfoResp::pGSMCallBarringSysInfo

8.499.2.9 BYTE∗ nasGetSysInfoResp::pGSMCipherDomainSysInfo

8.499.2.10 GSMSrvStatusInfo∗ nasGetSysInfoResp::pGSMSrvStatusInfo

8.499.2.11 GSMSysInfo∗ nasGetSysInfoResp::pGSMSysInfo

8.499.2.12 SrvStatusInfo∗ nasGetSysInfoResp::pHDRSrvStatusInfo

8.499.2.13 HDRSysInfo∗ nasGetSysInfoResp::pHDRSysInfo

8.499.2.14 GSMSrvStatusInfo∗ nasGetSysInfoResp::pLTESrvStatusInfo

8.499.2.15 LTESysInfo∗ nasGetSysInfoResp::pLTESysInfo

8.499.2.16 BYTE* nasGetSysInfoResp::pLTEVoiceSupportSysInfo

8.499.2.17 CallBarringSysInfo* nasGetSysInfoResp::pWCDMACallBarringSysInfo

8.499.2.18 BYTE* nasGetSysInfoResp::pWCDMACipherDomainSysInfo

8.499.2.19 GSMSrvStatusInfo* nasGetSysInfoResp::pWCDMASrvStatusInfo

8.499.2.20 WCDMASysInfo* nasGetSysInfoResp::pWCDMASysInfo

8.500 nasGetTxRxInfoReq Struct Reference

Data Fields

- BYTE radio_if

8.500.1 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

8.500.2 Field Documentation

8.500.2.1 BYTE nasGetTxRxInfoReq::radio_if

8.501 nasGetTxRxInfoResp Struct Reference

Data Fields

- rxInfo * pRXChain0Info
- rxInfo * pRXChain1Info
- txInfo * pTXInfo

8.501.1 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.

8.501.2 Field Documentation

8.501.2.1 rxInfo∗ nasGetTxRxInfoResp::pRXChain0Info

8.501.2.2 rxInfo∗ nasGetTxRxInfoResp::pRXChain1Info

8.501.2.3 txInfo∗ nasGetTxRxInfoResp::pTXInfo

8.502 NASGWAcqOrderPrefTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t GWAcqOrderPref

8.502.1 Detailed Description

Structure used to store GSM/WCDMA acquisition order preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
GWAcqOrder-Pref	- GSM/WCDMA acquisition order Preference Value

8.502.2 Field Documentation

8.502.2.1 uint32_t NASGWAcqOrderPrefTlv::GWAcqOrderPref

8.502.2.2 uint8_t NASGWAcqOrderPrefTlv::TlvPresent

8.503 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

8.503.1 Detailed Description

This structure contains the SLQSNasIndicationRegisterExt request parameters.

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 – 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.

8.503.2 Field Documentation

8.503.2.1 BYTE* nasIndicationRegisterReq::pDDTMInd

8.503.2.2 BYTE* nasIndicationRegisterReq::pDualStandByPrefInd

8.503.2.3 BYTE* nasIndicationRegisterReq::pErrorRateInd

8.503.2.4 BYTE* nasIndicationRegisterReq::pHDRNewUATIAssInd

8.503.2.5 BYTE* nasIndicationRegisterReq::pHDRSessionCloseInd

8.503.2.6 BYTE* nasIndicationRegisterReq::pLTECphyCa

8.503.2.7 BYTE* nasIndicationRegisterReq::pManagedRoamingInd

8.503.2.8 BYTE* nasIndicationRegisterReq::pNetworkRejectInd

8.503.2.9 BYTE* nasIndicationRegisterReq::pNetworkTimeInd

8.503.2.10 BYTE* nasIndicationRegisterReq::pServingSystemInd

8.503.2.11 BYTE* nasIndicationRegisterReq::pSignalStrengthInd

8.503.2.12 BYTE* nasIndicationRegisterReq::pSubscriptionInfoInd

8.503.2.13 BYTE* nasIndicationRegisterReq::pSuppressSysInfoInd

8.503.2.14 BYTE* nasIndicationRegisterReq::pSysInfoInd

8.503.2.15 BYTE* nasIndicationRegisterReq::pSystemSelectionInd

8.504 nasInitNetworkReg Struct Reference

Data Fields

- ULONG regAction
- MNRInfo * pMNRInfo
- ULONG * pChangeDuration
- BOOL * pMncPcsDigitStatus

8.504.1 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.

8.504.2 Field Documentation

8.504.2.1 ULONG* nasInitNetworkReg::pChangeDuration

8.504.2.2 BOOL* nasInitNetworkReg::pMncPcsDigitStatus

8.504.2.3 MNRInfo* nasInitNetworkReg::pMNRInfo

8.504.2.4 ULONG nasInitNetworkReg::regAction

8.505 NASLTEBandPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t LTEBandPref

8.505.1 Detailed Description

Structure used to store LTE Band Preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
LTEBandPref	- LTE Band Preference Value

8.505.2 Field Documentation

8.505.2.1 uint64_t NASLTEBandPreferenceTlv::LTEBandPref

8.505.2.2 uint8_t NASLTEBandPreferenceTlv::TlvPresent

8.506 NASLteNasReleaseInfoTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t nas_release
- uint8_t nas_major
- uint8_t nas_minor

8.506.1 Detailed Description

This structure contains LTE Nas Release Infomation

Parameters

Tlvresult	• unpack result
nas_release	• LTE NAS release
nas_major	• LTE NAS version major
nas_minor	• LTE NAS version minor

8.506.2 Field Documentation

8.506.2.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

8.506.2.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

8.506.2.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

8.506.2.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

8.507 NASModePreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t ModePref

8.507.1 Detailed Description

Structure used to store Mode Preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
ModePref	- Mode Preference Value

8.507.2 Field Documentation

8.507.2.1 uint16_t NASModePreferenceTlv::ModePref

8.507.2.2 uint8_t NASModePreferenceTlv::TlvPresent

8.508 NASNetSelPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t NetSelPref

8.508.1 Detailed Description

Structure used to store Network Selection Preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
NetSelPref	- Network Selection Preference Value

8.508.2 Field Documentation

8.508.2.1 uint8_t NASNetSelPreferenceTlv::NetSelPref

8.508.2.2 uint8_t NASNetSelPreferenceTlv::TlvPresent

8.509 nasNetworkReject Struct Reference

Data Fields

- BYTE radioIf
- TYPE_OF_SERVICE_DOMAIN serviceDomain
- BYTE rejectCause
- PlmnID * pPlmnId
- CsgId * pCsgId

8.509.1 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.

8.509.2 Field Documentation

8.509.2.1 CsgId* nasNetworkReject::pCsgId

8.509.2.2 PlmnID* nasNetworkReject::pPlmnId

8.509.2.3 BYTE nasNetworkReject::radioIf

8.509.2.4 BYTE nasNetworkReject::rejectCause

8.509.2.5 TYPE_OF_SERVICE_DOMAIN nasNetworkReject::serviceDomain

8.510 nasNetworkTime Struct Reference

Data Fields

- UniversalTime universalTime
- BYTE * pTimeZone
- BYTE * pDayItSavAdj
- BYTE * pRadioInterface

8.510.1 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

8.510.2 Field Documentation

8.510.2.1 BYTE* nasNetworkTime::pDayltSavAdj

8.510.2.2 BYTE* nasNetworkTime::pRadioInterface

8.510.2.3 BYTE* nasNetworkTime::pTimeZone

8.510.2.4 UniversalTime nasNetworkTime::universalTime

8.511 nasOperatorNameResp Struct Reference

Data Fields

- serviceProviderName * pSrvcProviderName
- operatorPLMNList * pOperatorPLMNList
- PLMNNetworkName * pPLMNNetworkName
- operatorNameString * pOperatorNameString
- PLMNNetworkNameData * pNITZInformation

8.511.1 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.

8.511.2 Field Documentation

8.511.2.1 PLMNNetworkNameData∗ nasOperatorNameResp::pNITZInformation

8.511.2.2 operatorNameString∗ nasOperatorNameResp::pOperatorNameString

8.511.2.3 operatorPLMNList∗ nasOperatorNameResp::pOperatorPLMNList

8.511.2.4 PLMNNetworkName∗ nasOperatorNameResp::pPLMNNetworkName

8.511.2.5 serviceProviderName∗ nasOperatorNameResp::pSrvcProviderName

8.512 NASOTAMessageTlv Struct Reference

Data Fields

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

8.512.1 Detailed Description

This structure contains OTA message

Parameters

Tlvresult	• unpack result
message_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_buf	• OTA Message Content

8.512.2 Field Documentation

8.512.2.1 uint8_t NASOTAMessageTlv::data_buf[2048]

8.512.2.2 uint16_t NASOTAMessageTlv::data_len

8.512.2.3 uint32_t NASOTAMessageTlv::message_type

8.512.2.4 uint8_t NASOTAMessageTlv::TlvPresent

8.513 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

8.513.1 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 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information.
scell_state	• Scell state Values. • See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.513.2 Field Documentation

8.513.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value

8.513.2.2 uint32_t NASPhyCaAggPcellInfo::freq

8.513.2.3 uint32_t NASPhyCaAggPcellInfo::iLTEbandValue

8.513.2.4 uint32_t NASPhyCaAggPcellInfo::pci

8.513.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

8.514 NASPhyCaAggScellDlBw Struct Reference

Data Fields

- LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- uint8_t TlvPresent

8.514.1 Detailed Description

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

Parameters

dl_bw_value	• Downlink Bandwidth Values. • See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information.

8.514.2 Field Documentation

8.514.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellDlBw::dl_bw_value

8.514.2.2 uint8_t NASPhyCaAggScellDlBw::TlvPresent

8.515 NASPhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

8.515.1 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.

8.515.2 Field Documentation

8.515.2.1 uint8_t NASPhyCaAggScellIndex::scell_idx

8.515.2.2 uint8_t NASPhyCaAggScellIndex::TlvPresent

8.516 NASPhyCaAggScellIndType Struct Reference

Data Fields

- uint32_t pci
- uint32_t freq
- LITEQMI_NAS_LTE_CPHY_SCELL_STATE scell_state
- uint8_t TlvPresent

8.516.1 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 LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.516.2 Field Documentation

8.516.2.1 uint32_t NASPhyCaAggScellIndType::freq

8.516.2.2 uint32_t NASPhyCaAggScellIndType::pci

8.516.2.3 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

8.516.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

8.517 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

8.517.1 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 LITEQMI_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 LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information.
TlvPresent	• Tlv Present.

8.517.2 Field Documentation

8.517.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

8.517.2.2 uint32_t NASPhyCaAggScellInfo::freq

8.517.2.3 uint32_t NASPhyCaAggScellInfo::iLTEbandValue

8.517.2.4 uint32_t NASPhyCaAggScellInfo::pci

8.517.2.5 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

8.517.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

8.518 nasPLMNNameReq Struct Reference

Data Fields

- WORD mcc
- WORD mnc

- BYTE * pMncPcsStatus

8.518.1 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.

8.518.2 Field Documentation

8.518.2.1 WORD nasPLMNNameReq::mcc

8.518.2.2 WORD nasPLMNNameReq::mnc

8.518.2.3 BYTE* nasPLMNNameReq::pMncPcsStatus

8.519 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]

8.519.1 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

8.519.2 Field Documentation

8.519.2.1 BYTE nasPLMNNameResp::longName[255]

8.519.2.2 BYTE nasPLMNNameResp::longNameCI

8.519.2.3 BYTE nasPLMNNameResp::longNameEn

8.519.2.4 BYTE nasPLMNNameResp::longNameLen

8.519.2.5 BYTE nasPLMNNameResp::longNameSB

8.519.2.6 BYTE nasPLMNNameResp::shortName[255]

8.519.2.7 BYTE nasPLMNNameResp::shortNameCI

8.519.2.8 BYTE nasPLMNNameResp::shortNameEn

8.519.2.9 BYTE nasPLMNNameResp::shortNameLen

8.519.2.10 BYTE nasPLMNNameResp::shortNameSB

8.519.2.11 BYTE nasPLMNNameResp::spn[255]

8.519.2.12 BYTE nasPLMNNameResp::spnEncoding

8.519.2.13 BYTE nasPLMNNameResp::spnLength

8.520 NASPRLPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t PRLPref

8.520.1 Detailed Description

Structure used to store CDMA PRL Preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
PRLPref	- CDMA PRL Preference Value

8.520.2 Field Documentation

8.520.2.1 uint16_t NASPRLPreferenceTlv::PRLPref

8.520.2.2 uint8_t NASPRLPreferenceTlv::TlvPresent

8.521 NASQmiCbkNasSwiOTAMessageInd Struct Reference

Data Fields

- NASOTAMessageTlv otaMsgTlv
- NASLteNasReleaseInfoTlv nasRelInfoTlv
- NASTimeInfoTlv timeTlv

8.521.1 Detailed Description

This structure contains OTA message

Parameters

otaMsgTlv	• See NASOTAMessageTlv for more details
nasRelInfoTlv	• See NASLteNasReleaseInfoTlv for more details
timeTlv	• See NASTimeInfoTlv for more details

8.521.2 Field Documentation

8.521.2.1 NASLteNasReleaseInfoTlv NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv

8.521.2.2 NASOTAMessageTlv NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv

8.521.2.3 NASTimeInfoTlv NASQmiCbkNasSwiOTAMessageInd::timeTlv

8.522 NASQmiCbkNasSystemSelPrefInd Struct Reference

Data Fields

- NASEmergencyModeTlv EMTlv
- NASModePreferenceTlv MPTlv
- NASBandPreferenceTlv BPTlv
- NASPRLPreferenceTlv PRLPTlv
- NASRoamPreferenceTlv RPTlv
- NASLTEBandPreferenceTlv LBPTlv
- NASNetSelPreferenceTlv NSPTlv
- NASServDomainPrefTlv SDPTlv
- NASGWAcqOrderPrefTlv GWAOPTlv

8.522.1 Detailed Description

Structure used to store all QMI Notification parameters.

Parameters

EMTlv	- Emergency mode Tlv • See NASEmergencyModeTlv for more details
MPTlv	- Mode preference Tlv • See NASModePreferenceTlv for more details
BPTlv	- Band preference Tlv • See NASBandPreferenceTlv for more details
PRLPTlv	- PRL preference Tlv • See NASPRLPreferenceTlv for more details
RPTlv	- Roaming preference Tlv • See NASRoamPreferenceTlv for more details
LBPTlv	- LTE band preference Tlv • See NASLTEBandPreferenceTlv for more details
NSPTlv	- Network selection preference Tlv • See NASNetSelPreferenceTlv for more details
SDPTlv	- Service domain preference Tlv • See NASServDomainPrefTlv for more details
GWAOPTlv	- GSM WCDMA Acquisition Order Preference Tlv • See NASGWAcqOrderPrefTlv for more details

Note: None

8.522.2 Field Documentation

8.522.2.1 NASBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::BPTlv

8.522.2.2 NASEmergencyModeTlv NASQmiCbkNasSystemSelPrefInd::EMTlv

8.522.2.3 NASGWAcqOrderPrefTlv NASQmiCbkNasSystemSelPrefInd::GWAOPTlv

8.522.2.4 NASLTEBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::LBPTlv

8.522.2.5 NASModePreferenceTlv NASQmiCbkNasSystemSelPrefInd::MPTlv

8.522.2.6 NASNetSelPreferenceTlv NASQmiCbkNasSystemSelPrefInd::NSPTlv

8.522.2.7 NASPRLPreferenceTlv NASQmiCbkNasSystemSelPrefInd::PRLPTlv

8.522.2.8 NASRoamPreferenceTlv NASQmiCbkNasSystemSelPrefInd::RPTlv

8.522.2.9 NASServDomainPrefTlv NASQmiCbkNasSystemSelPrefInd::SDPTlv

8.523 NASRoamPreferenceTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint16_t RoamPref

8.523.1 Detailed Description

Structure used to store Roaming Preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
RoamPref	- Roaming Preference Value

8.523.2 Field Documentation

8.523.2.1 uint16_t NASRoamPreferenceTlv::RoamPref

8.523.2.2 uint8_t NASRoamPreferenceTlv::TlvPresent

8.524 NASServDomainPrefTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t SrvDomainPref

8.524.1 Detailed Description

Structure used to store Service domain preference TLV Value.

Parameters

TlvPresent	- Boolean indicating the presence of the TLV in the QMI response
SrvDomainPref	- Service Domain Preference Value

8.524.2 Field Documentation

8.524.2.1 uint32_t NASServDomainPrefTlv::SrvDomainPref

8.524.2.2 uint8_t NASServDomainPrefTlv::TlvPresent

8.525 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

8.525.1 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

8.525.2 Field Documentation

8.525.2.1 uint8_t NASServingSystemInfo::csAttachState

8.525.2.2 uint8_t NASServingSystemInfo::hdrPersonality

8.525.2.3 uint8_t NASServingSystemInfo::psAttachState

8.525.2.4 uint8_t NASServingSystemInfo::radioInterfaceList[255]

8.525.2.5 uint8_t NASServingSystemInfo::radioInterfaceNo

8.525.2.6 uint8_t NASServingSystemInfo::registrationState

8.525.2.7 uint8_t NASServingSystemInfo::selectedNetwork

8.526 nasSigInfo Struct Reference

Data Fields

- CDMASSInfo * pCDMASigInfo
- HDRSSInfo * pHDRSigInfo
- INT8 * pGSMSigInfo
- CDMASSInfo * pWCDMASigInfo
- LTESSInfo * pLTESigInfo
- INT8 * pRscp
- TDSCDMASigInfoExt * pTDSCDMASigInfoExt

8.526.1 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.

8.526.2 Field Documentation

8.526.2.1 CDMASSInfo* nasSigInfo::pCDMASigInfo

8.526.2.2 INT8* nasSigInfo::pGSMSigInfo

8.526.2.3 HDRSSInfo* nasSigInfo::pHDRSigInfo

8.526.2.4 LTESSInfo* nasSigInfo::pLTESigInfo

8.526.2.5 INT8* nasSigInfo::pRscp

8.526.2.6 TDSCDMASigInfoExt* nasSigInfo::pTDSCDMASigInfoExt

8.526.2.7 CDMASSInfo* nasSigInfo::pWCDMASigInfo

8.527 nasSwiGetChannelLockResp Struct Reference

Data Fields

- wcdmaUARFCN * pWcdmaUARFCN
- lteEARFCN * pLteEARFCN
- ltePCI * pLtePCI

8.527.1 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

8.527.2 Field Documentation

8.527.2.1 lteEARFCN∗ nasSwiGetChannelLockResp::pLteEARFCN

8.527.2.2 ltePCI∗ nasSwiGetChannelLockResp::pLtePCI

8.527.2.3 wcdmaUARFCN∗ nasSwiGetChannelLockResp::pWcdmaUARFCN

8.528 NasSwiIndReg Struct Reference

Data Fields

- BYTE lteEsmUl
- BYTE lteEsmDl
- BYTE lteEmmUl
- BYTE lteEmmDl
- BYTE gsmUmtsUl
- BYTE gsmUmtsDl
- BYTE ∗ pRankIndicatorInd
- BYTE ∗ pTimer

8.528.1 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

8.528.2 Field Documentation

8.528.2.1 BYTE NasSwiIndReg::gsmUmtsDl

8.528.2.2 BYTE NasSwiIndReg::gsmUmtsUl

8.528.2.3 BYTE NasSwiIndReg::lteEmmDl

8.528.2.4 BYTE NasSwiIndReg::lteEmmUl

8.528.2.5 BYTE NasSwiIndReg::lteEsmDl

8.528.2.6 BYTE NasSwiIndReg::lteEsmUl

8.528.2.7 BYTE* NasSwiIndReg::pRankIndicatorInd

8.528.2.8 BYTE* NasSwiIndReg::pTimer

8.529 nasSwiSetChannelLockReq Struct Reference

Data Fields

- wcdmaUARFCN * pWcdmaUARFCN
- lteEARFCN * pLteEARFCN
- ltePCI * pLtePCI

8.529.1 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

8.529.2 Field Documentation

8.529.2.1 lteEARFCN∗ nasSwiSetChannelLockReq::pLteEARFCN

8.529.2.2 ltePCI∗ nasSwiSetChannelLockReq::pLtePCI

8.529.2.3 wcdmaUARFCN∗ nasSwiSetChannelLockReq::pWcdmaUARFCN

8.530 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

8.530.1 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.

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
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

8.530.2 Field Documentation

8.530.2.1 AddCDMASysInfo∗ nasSysInfo::pAddCDMASysInfo

8.530.2.2 AddSysInfo∗ nasSysInfo::pAddGSMSysInfo

8.530.2.3 WORD∗ nasSysInfo::pAddHDRSysInfo

8.530.2.4 WORD∗ nasSysInfo::pAddLTESysInfo

8.530.2.5 AddSysInfo∗ nasSysInfo::pAddWCDMASysInfo

8.530.2.6 SrvStatusInfo∗ nasSysInfo::pCDMASrvStatusInfo

8.530.2.7 CDMASysInfo∗ nasSysInfo::pCDMASysInfo

8.530.2.8 CallBarringSysInfo∗ nasSysInfo::pGSMCallBarringSysInfo

8.530.2.9 BYTE∗ nasSysInfo::pGSMCipherDomainSysInfo

8.530.2.10 GSMSrvStatusInfo∗ nasSysInfo::pGSMSrvStatusInfo

8.530.2.11 GSMSysInfo∗ nasSysInfo::pGSMSysInfo

8.530.2.12 SrvStatusInfo* nasSysInfo::pHDRSrvStatusInfo

8.530.2.13 HDRSysInfo* nasSysInfo::pHDRSysInfo

8.530.2.14 GSMSrvStatusInfo* nasSysInfo::pLTESrvStatusInfo

8.530.2.15 LTESysInfo* nasSysInfo::pLTESysInfo

8.530.2.16 BYTE* nasSysInfo::pLTEVoiceSupportSysInfo

8.530.2.17 BYTE* nasSysInfo::pSysInfoNoChange

8.530.2.18 CallBarringSysInfo* nasSysInfo::pWCDMACallBarringSysInfo

8.530.2.19 BYTE* nasSysInfo::pWCDMACipherDomainSysInfo

8.530.2.20 GSMSrvStatusInfo* nasSysInfo::pWCDMASrvStatusInfo

8.530.2.21 WCDMASysInfo* nasSysInfo::pWCDMASysInfo

8.531 NASTimeInfoTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t time

8.531.1 Detailed Description

This structure contains LTE Nas time Infomation

Parameters

Tlvresult	• unpack result
time	• Seconds in local time since Jan. 6th 1980 00:00:00 UTC

8.531.2 Field Documentation

8.531.2.1 uint64_t NASTimeInfoTlv::time

8.531.2.2 uint8_t NASTimeInfoTlv::TlvPresent

8.532 nasTimers Struct Reference

Data Fields

- char t3396_apn [101]
- BYTE t3396_plmn_id [3]
- ULONG t3396_val

8.532.1 Detailed Description

This structure contains Network Timer

Parameters

t3396_apn	• APN name
t3396_plmn_id	• PLMN ID
t3396_val	• T3396 timer value

8.532.2 Field Documentation

8.532.2.1 char nasTimers::t3396_apn[101]

8.532.2.2 BYTE nasTimers::t3396_plmn_id[3]

8.532.2.3 ULONG nasTimers::t3396_val

8.533 netSelectionPref Struct Reference

Data Fields

- BYTE netReg
- WORD mcc
- WORD mnc

8.533.1 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

8.533.2 Field Documentation

8.533.2.1 WORD netSelectionPref::mcc

8.533.2.2 WORD netSelectionPref::mnc

8.533.2.3 BYTE netSelectionPref::netReg

8.534 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

8.534.1 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

8.534.2 Field Documentation

8.534.2.1 ULONGLONG NetStats::rx_bytes

8.534.2.2 ULONG NetStats::rx_errors

8.534.2.3 ULONG NetStats::rx_overflows

8.534.2.4 ULONG NetStats::rx_packets

8.534.2.5 ULONGLONG NetStats::tx_bytes

8.534.2.6 ULONG NetStats::tx_errors

8.534.2.7 ULONG NetStats::tx_overflows

8.534.2.8 ULONG NetStats::tx_packets

8.535 NetworkDebugResp Struct Reference

Data Fields

- BYTE * pObjectVer
- NetworkStat1x * pNetworkStat1x
- NetworkStatEVDO * pNetworkStatEVDO
- DeviceConfigDetail * pDeviceConfigDetail
- DataStatusDetail * pDataStatusDetail

8.535.1 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.

8.535.2 Field Documentation

8.535.2.1 DataStatusDetail* NetworkDebugResp::pDataStatusDetail

8.535.2.2 DeviceConfigDetail* NetworkDebugResp::pDeviceConfigDetail

8.535.2.3 NetworkStat1x* NetworkDebugResp::pNetworkStat1x

8.535.2.4 NetworkStatEVDO* NetworkDebugResp::pNetworkStatEVDO

8.535.2.5 BYTE* NetworkDebugResp::pObjectVer

8.536 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

8.536.1 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

8.536.2 Field Documentation

8.536.2.1 BYTE NetworkStat1x::ActSetCnt

8.536.2.2 BYTE NetworkStat1x::NeighborSetCnt

8.536.2.3 ActPilotPNElement* NetworkStat1x::pActPilotPNElements

8.536.2.4 WORD* NetworkStat1x::pNeighborSetPilotPN

8.536.2.5 WORD NetworkStat1x::RX_EC_IO

8.536.2.6 ULONG NetworkStat1x::RX_PWR

8.536.2.7 WORD NetworkStat1x::SO

8.536.2.8 BYTE NetworkStat1x::State

8.536.2.9 ULONG NetworkStat1x::TX_PWR

8.537 NetworkStatEVDO Struct Reference

Data Fields

- BYTE State
- BYTE MACIndex
- BYTE SectorIDLen
- WORD * pSectorID
- WORD RX_PWR
- WORD PER
- WORD PilotEnergy
- BYTE SNR

8.537.1 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)

8.537.2 Field Documentation

8.537.2.1 BYTE NetworkStatEVDO::MACIndex

8.537.2.2 WORD NetworkStatEVDO::PER

8.537.2.3 WORD NetworkStatEVDO::PilotEnergy

8.537.2.4 WORD* NetworkStatEVDO::pSectorID

8.537.2.5 WORD NetworkStatEVDO::RX_PWR

8.537.2.6 BYTE NetworkStatEVDO::SectorIDLen

8.537.2.7 BYTE NetworkStatEVDO::SNR

8.537.2.8 BYTE NetworkStatEVDO::State

8.538 newMTMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSMTMessageInfo MTMessageInfo

8.538.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
MTMessageInfo	• MT Message • See sMSMTMessageInfo for more information

8.538.2 Field Documentation

8.538.2.1 sMSMTMessageInfo newMTMessageTlv::MTMessageInfo

8.538.2.2 uint8_t newMTMessageTlv::TlvPresent

8.539 newPwdData Struct Reference

Data Fields

- BYTE newPwd [4]
- BYTE newPwdAgain [4]

8.539.1 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.

8.539.2 Field Documentation

8.539.2.1 BYTE newPwdData::newPwd[4]

8.539.2.2 BYTE newPwdData::newPwdAgain[4]

8.540 nmrCellInfo Struct Reference

Data Fields

- ULONG nmrCellID

- BYTE nmrPlmn [3]
- WORD nmrLac
- WORD nmrArfcn
- BYTE nmrBsic
- WORD nmrRxLev

8.540.1 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

8.540.2 Field Documentation

8.540.2.1 WORD nmrCellInfo::nmrArfcn

8.540.2.2 BYTE nmrCellInfo::nmrBsic

8.540.2.3 ULONG nmrCellInfo::nmrCellID

8.540.2.4 WORD nmrCellInfo::nmrLac

8.540.2.5 BYTE nmrCellInfo::nmrPlmn[3]

8.540.2.6 WORD nmrCellInfo::nmrRxLev

8.541 NSSAudioCtrl Struct Reference

Data Fields

- BYTE upLink
- BYTE downLink

8.541.1 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].

8.541.2 Field Documentation

8.541.2.1 BYTE NSSAudioCtrl::downLink

8.541.2.2 BYTE NSSAudioCtrl::upLink

8.542 NWProfile Struct Reference

Data Fields

- WORD tech
- BYTE * pProfSz
- WORD * pProfValues

8.542.1 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	

8.542.2 Field Documentation

8.542.2.1 BYTE* NWProfile::pProfSz

8.542.2.2 WORD* NWProfile::pProfValues

8.542.2.3 WORD NWProfile::tech

8.543 omaDmConfigTlv Struct Reference

Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- USHORT alertmsglength
- BYTE alertmsg [256]

8.543.1 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
userInputReq	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject
userInput-Timeout	• Timeout for user input in minutes. A value of 0 means no time-out
alertmsglength	• Length of Alert message string in bytes
alertmsg	• Alert message in UCS2 (Max 256 characters)

8.543.2 Field Documentation

8.543.2.1 BYTE omaDmConfigTlv::alertmsg[256]

8.543.2.2 USHORT omaDmConfigTlv::alertmsglength

8.543.2.3 BYTE omaDmConfigTlv::state

8.543.2.4 BYTE omaDmConfigTlv::userInputReq

8.543.2.5 USHORT omaDmConfigTlv::userInputTimeout

8.544 omaDmConfigTlvExt Struct Reference

Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- USHORT alertmsglength
- BYTE alertmsg [256]

8.544.1 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

8.544.2 Field Documentation

8.544.2.1 BYTE omaDmConfigTlvExt::alertmsg[256]

8.544.2.2 USHORT omaDmConfigTlvExt::alertmsglength

8.544.2.3 BYTE omaDmConfigTlvExt::state

8.544.2.4 BYTE omaDmConfigTlvExt::userInputReq

8.544.2.5 USHORT omaDmConfigTlvExt::userInputTimeout

8.545 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

8.545.1 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

userInputReq	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject
userInput-Timeout	• Timeout for user input in minutes. A value of 0 means no time-out
fwdloadsize	• The size (in bytes) of the firmware update package
fwloadComplete	• The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment.
update-CompleteStatus	• This field should be looked at only when the OMADM session is complete. • See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h for update complete status.
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

8.545.2 Field Documentation

8.545.2.1 BYTE omaDmFotaTlv::description[256]

8.545.2.2 USHORT omaDmFotaTlv::descriptionlength

8.545.2.3 ULONG omaDmFotaTlv::fwdloadsize

8.545.2.4 ULONG omaDmFotaTlv::fwloadComplete

8.545.2.5 USHORT omaDmFotaTlv::namelength

8.545.2.6 BYTE omaDmFotaTlv::package_name[256]

8.545.2.7 BYTE omaDmFotaTlv::sessionType

8.545.2.8 BYTE omaDmFotaTlv::severity

8.545.2.9 BYTE omaDmFotaTlv::state

8.545.2.10 USHORT omaDmFotaTlv::updateCompleteStatus

8.545.2.11 BYTE omaDmFotaTlv::userInputReq

8.545.2.12 USHORT omaDmFotaTlv::userInputTimeout

8.545.2.13 BYTE omaDmFotaTlv::version[256]

8.545.2.14 USHORT omaDmFotaTlv::versionlength

8.546 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]

8.546.1 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)

8.546.2 Field Documentation

8.546.2.1 BYTE omaDmFotaTlvExt::description[256]

8.546.2.2 USHORT omaDmFotaTlvExt::descriptionlength

8.546.2.3 USHORT omaDmFotaTlvExt::fumoResultCode

8.546.2.4 USHORT omaDmFotaTlvExt::namelength

8.546.2.5 BYTE omaDmFotaTlvExt::package_name[256]

8.546.2.6 ULONG omaDmFotaTlvExt::packageSize

8.546.2.7 ULONG omaDmFotaTlvExt::receivedBytes

8.546.2.8 BYTE omaDmFotaTlvExt::reserved

8.546.2.9 BYTE omaDmFotaTlvExt::state

8.546.2.10 USHORT omaDmFotaTlvExt::userInputTimeout

8.546.2.11 BYTE omaDmFotaTlvExt::version[256]

8.546.2.12 USHORT omaDmFotaTlvExt::versionlength

8.547 omaDmNotificationsTlv Struct Reference

Data Fields

- BYTE notification
- USHORT sessionStatus

8.547.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

Parameters

notification	• 0x01 - GPS settings change • 0x02 - Device reset • 0x03 - Device factory reset • 0x04 - CI-DC Session start • 0x05 - CI-DC Session end • 0x06 - CI-PRL Session start • 0x07 - CI-PRL Session end • 0x08 - CI-FUMO Session start • 0x09 - CI-FUMO session end • 0x0A - HFA-DC Session start • 0x0B - HFA-DC Session end • 0x0C - HFA-PRL Session start • 0x0D - HFA-PRL Session end • 0x0E - HFA-FUMO Session start • 0x0F - HFA-FUMO session end • 0x10 - NI Session start • 0x11 - NI session end
sessionStatus	• This field will set to the session status for notifications that occur at the end of a session, zero for all other notifications • See qaGobiApiTableSwiOMADMSessionStatus.h for session completion code.

8.547.2 Field Documentation

8.547.2.1 BYTE omaDmNotificationsTlv::notification

8.547.2.2 USHORT omaDmNotificationsTlv::sessionStatus

8.548 operatorNameString Struct Reference

Data Fields

- BYTE PLMNName [255]

8.548.1 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.

8.548.2 Field Documentation

8.548.2.1 BYTE operatorNameString::PLMNName[255]

8.549 OperatorPLMNData Struct Reference

Data Fields

- BYTE mcc [3]
- BYTE mnc [3]
- WORD lac1
- WORD lac2
- BYTE PLMNRecID

8.549.1 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.

8.549.2 Field Documentation

8.549.2.1 WORD OperatorPLMNData::lac1

8.549.2.2 WORD OperatorPLMNData::lac2

8.549.2.3 BYTE OperatorPLMNData::mcc[3]

8.549.2.4 BYTE OperatorPLMNData::mnc[3]

8.549.2.5 BYTE OperatorPLMNData::PLMNRecID

8.550 operatorPLMNList Struct Reference

Data Fields

- WORD numInstance
- OperatorPLMNData PLMNData [255]

8.550.1 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).

8.550.2 Field Documentation

8.550.2.1 WORD operatorPLMNList::numInstance

8.550.2.2 OperatorPLMNData operatorPLMNList::PLMNData[255]

8.551 pack_dms_ActivateAutomatic_t Struct Reference

Data Fields

- uint8_t actCode [81]

8.551.1 Detailed Description

This structure contains UIM activate automation information

Parameters

actCode	• 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"

8.551.2 Field Documentation

8.551.2.1 uint8_t pack_dms_ActivateAutomatic_t::actCode[81]

8.552 pack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

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

8.552.1 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) • NULL terminated ASCII string.
list_type	• list type requested
Tlvresult	• Pack Result

8.552.2 Field Documentation

8.552.2.1 uint8_t pack_dms_GetCustFeaturesV2_t::cust_id[64+1]

8.552.2.2 uint8_t pack_dms_GetCustFeaturesV2_t::list_type

8.552.2.3 uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult

8.553 pack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- uint8_t spc [6]

8.553.1 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)

8.553.2 Field Documentation

8.553.2.1 uint8_t pack_dms_ResetToFactoryDefaults_t::spc[6]

8.554 pack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- uint8_t activationState

8.554.1 Detailed Description

This structure is used to store Set Service Activation Status callback parameter pack.

Parameters

activationState	• Service activation state. • Values – 0 - Do not report – 1 - Report activation state changes

8.554.2 Field Documentation

8.554.2.1 uint8_t pack_dms_SetActivationStatusCallback_t::activationState

8.555 pack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t crashAction

8.555.1 Detailed Description

This structure is used to store pack_dms_SetCrashAction parameters.

Parameters

crashAction	• 0 - USB Memory Download. Modem will reset after a crash and will stay in USB download mode with only DM port enumerated. • 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

8.555.2 Field Documentation

8.555.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

8.556 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

8.556.1 Detailed Description

This structure contains current settings of custom features pack parameters

Parameters

GpsEnable	• describes if GPS is enabled or disabled • values: – 0x00 - GPS is disabled – 0x01 - GPS is enabled
DisableIMSI	• optional 1 byte parameter • describes if IMSI display is enabled or disabled • values: – 0x00 - Allow display of IMSI – 0x01 - Do not display IMSI
IPFamSupport	• optional 2 byte BitMask • bitmask representing the IP families supported • values: – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - IPv4v6
RMAutoConnect	• optional 1 byte parameter • QMI Mode RM Net Auto Connect Support • values: – 0x00 - Not Supported – 0x01 - Supported
GPSSel	• 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
SMSSupport	• optional 1 byte parameter • SMS support • values: – 0x00 - Not supported – 0x01 - supported • Used to determine whether or not to hide SMS from user

IsVoiceEnabled	• 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
DHCPRelay-Enabled	• optional 1 byte parameter • DHCP Relay support • values: – 0x00 - Disable DHCP relay – 0x01 - Enable DHCP relay
GPSLPM	• optional 1 byte parameter • GPSLPM support • values: – 0x00 - Enable GPS in Low Power Mode – 0x01 - Disable GPS in Low Power Mode

8.556.2 Field Documentation

8.556.2.1 uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled

8.556.2.2 uint8_t pack_dms_SetCustFeature_t::DisableIMSI

8.556.2.3 uint32_t pack_dms_SetCustFeature_t::GpsEnable

8.556.2.4 uint8_t pack_dms_SetCustFeature_t::GPSLPM

8.556.2.5 uint8_t pack_dms_SetCustFeature_t::GPSSel

8.556.2.6 uint16_t pack_dms_SetCustFeature_t::IPFamSupport

8.556.2.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

8.556.2.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

8.556.2.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

8.557 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

8.557.1 Detailed Description

This structure contains customization settings set to modem pack

Parameters

cust_id	• Customization ID (Maximum 64 bytes) • NULL terminated ASCII string.
value_length	• length of cust_value field
cust_value	• Customization Setting Value (Maximum 8 bytes)
Tlvresult	• Pack Result

8.557.2 Field Documentation

8.557.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

8.557.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

8.557.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

8.557.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

8.558 pack_dms_SetEventReport_t Struct Reference

Data Fields

- uint8_t mode

8.558.1 Detailed Description

This structure is used to store pack_dms_SetEventReport parameters.

Parameters

mode	• Operating Mode • 0 - reset • 1 - set • 2 - no change

8.558.2 Field Documentation

8.558.2.1 uint8_t pack_dms_SetEventReport_t::mode

8.559 pack_dms_SetPower_t Struct Reference

Data Fields

- uint32_t mode
- uint16_t Tlvresult

8.559.1 Detailed Description

Structure to store pack the operating mode of the device.

Parameters

mode	• Selected operating mode – See qaGobiApiTablePowerModes.h for power modes
Tlvresult	• pack Tlv Result.

8.559.2 Field Documentation

8.559.2.1 uint32_t pack_dms_SetPower_t::mode

8.559.2.2 uint16_t pack_dms_SetPower_t::Tlvresult

8.560 pack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint8_t USBComp
- uint16_t Tlvresult

8.560.1 Detailed Description

This structure is used to store USB composition information pack paramters

Parameters

USBComp	• 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
Tlvresult	• pack Tlv Result.

8.560.2 Field Documentation

8.560.2.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

8.560.2.2 uint8_t pack_dms_SetUSBComp_t::USBComp

8.561 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint8_t resetInfoInd

8.561.1 Detailed Description

This structure contains set registration state for different indication pack

Parameters

resetInfoInd[IN]	• Values – 0 - Disable – 1 - Enable

8.561.2 Field Documentation

8.561.2.1 uint8_t pack_dms_SLQSDmsSwiIndicationRegister_t::resetInfoInd

8.562 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference

Data Fields

- uint8_t clear

8.562.1 Detailed Description

This structure contains pack_dms_SLQSSwiGetCrashInfo parameters

Parameters

clear	• 0 - do not clear crash data after response • 1 - clear crash data after response

8.562.2 Field Documentation

8.562.2.1 uint8_t pack_dms_SLQSSwiGetCrashInfo_t::clear

8.563 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint8_t * pDestSMSNum
- uint8_t * pDestSMSContent

8.563.1 Detailed Description

This structure is used to store pack_dms_SLQSSwiSetDyingGaspCfg parameters

Parameters

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

8.563.2 Field Documentation

8.563.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

8.563.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

8.564 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

Data Fields

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

8.564.1 Detailed Description

This structure contains SWI set host device info unpack information

Parameters

manString[IN]	• Host Device Manufacturer Name • Null terminated ASCII String
modelString[IN]	• Host Device Model String • Null terminated ASCII string.
swVerString[IN]	• Host Device Software Version String • Null terminated ASCII string
plasmaIDString[IN]	• Host Device Plasma ID String • Null terminated alphanumeric ASCII string

8.564.2 Field Documentation

8.564.2.1 char pack_dms_SLQSSwiSetHostDevInfo_t::manString[255]

8.564.2.2 char pack_dms_SLQSSwiSetHostDevInfo_t::modelString[255]

8.564.2.3 char pack_dms_SLQSSwiSetHostDevInfo_t::plasmaIDString[255]

8.564.2.4 char pack_dms_SLQSSwiSetHostDevInfo_t::swVerString[255]

8.565 pack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

- char nameString [255]
- char versionString [255]

8.565.1 Detailed Description

This structure contains SWI set host OS info pack information

Parameters

nameString[IN]	Host device manufacture String • NULL terminated ASCII string.
VersionString[IN]	Host device model String • NULL terminated ASCII string.

8.565.2 Field Documentation

8.565.2.1 char pack_dms_SLQSSwiSetOSInfo_t::nameString[255]

8.565.2.2 char pack_dms_SLQSSwiSetOSInfo_t::versionString[255]

8.566 pack_dms_UIMChangePIN_t Struct Reference

Data Fields

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

8.566.1 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

8.566.2 Field Documentation

8.566.2.1 uint8_t pack_dms_UIMChangePIN_t::id

8.566.2.2 uint8_t pack_dms_UIMChangePIN_t::newValue[255]

8.566.2.3 uint8_t pack_dms_UIMChangePIN_t::oldValue[255]

8.567 pack_dms_UIMGetControlKeyStatus_t Struct Reference

Data Fields

- uint8_t facility

8.567.1 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)

8.567.2 Field Documentation

8.567.2.1 uint8_t pack_dms_UIMGetControlKeyStatus_t::facility

8.568 pack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.568.1 Detailed Description

This structure contains UIM Get ICCID pack

Parameters

Tlvresult	• Pack result.

8.568.2 Field Documentation

8.568.2.1 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

8.569 pack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

- uint8_t facility
- uint8_t facilityState
- uint8_t facilityCk [8]

8.569.1 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 – 0 - Deactivated – 1 - Activated – 2 - Blocked
facilityCk[IN]	• Facility depersonalization control key, string in ASCII text (maximum 8 bytes)

8.569.2 Field Documentation

8.569.2.1 uint8_t pack_dms_UIMSetControlKeyProtection_t::facility

8.569.2.2 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityCk[8]

8.569.2.3 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityState

8.570 pack_dms_UIMSetPINProtection_t Struct Reference

Data Fields

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

8.570.1 Detailed Description

This structure contains PIN Protection Information

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

8.570.2 Field Documentation

8.570.2.1 uint8_t pack_dms_UIMSetPINProtection_t::bEnable

8.570.2.2 uint8_t pack_dms_UIMSetPINProtection_t::id

8.570.2.3 uint8_t pack_dms_UIMSetPINProtection_t::value[255]

8.571 pack_dms_UIMUnblockControlKey_t Struct Reference

Data Fields

- uint8_t facility
- uint8_t facilityCk [8]

8.571.1 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)

8.571.2 Field Documentation

8.571.2.1 uint8_t pack_dms_UIMUnblockControlKey_t::facility

8.571.2.2 uint8_t pack_dms_UIMUnblockControlKey_t::facilityCk[8]

8.572 pack_dms_UIMUnblockPIN_t Struct Reference

Data Fields

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

8.572.1 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

8.572.2 Field Documentation

8.572.2.1 uint8_t pack_dms_UIMUnblockPIN_t::id

8.572.2.2 uint8_t pack_dms_UIMUnblockPIN_t::newPin[255]

8.572.2.3 uint8_t pack_dms_UIMUnblockPIN_t::pukValue[255]

8.573 pack_dms_UIMVerifyPIN_t Struct Reference

Data Fields

- uint8_t id
- uint8_t value [255]

8.573.1 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

8.573.2 Field Documentation

8.573.2.1 uint8_t pack_dms_UIMVerifyPIN_t::id

8.573.2.2 uint8_t pack_dms_UIMVerifyPIN_t::value[255]

8.574 pack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.574.1 Detailed Description

This structure contains the Get Image Preference information pack

Parameters

Tlvresult	• Pack result

8.574.2 Field Documentation

8.574.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

8.575 pack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.575.1 Detailed Description

This structure contains the Get Stored Images pack

Parameters

Tlvresult	• Pack result

8.575.2 Field Documentation

8.575.2.1 uint16_t pack_fms_GetStoredImages_t::Tlvresult

8.576 pack_fms_SetImagesPreference_t Struct Reference

Data Fields

- uint32_t imageListSize
- FMSPrefImageList * pImageList
- uint32_t bForceDownload
- uint8_t modemindex
- uint16_t Tlvresult

8.576.1 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

8.576.2 Field Documentation

8.576.2.1 uint32_t pack_fms_SetImagesPreference_t::bForceDownload

8.576.2.2 uint32_t pack_fms_SetImagesPreference_t::imageListSize

8.576.2.3 uint8_t pack_fms_SetImagesPreference_t::modemindex

8.576.2.4 FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList

8.576.2.5 uint16_t pack_fms_SetImagesPreference_t::Tlvresult

8.577 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

8.577.1 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.

8.577.2 Field Documentation

8.577.2.1 loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo

8.577.2.2 loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb

8.577.2.3 loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo

8.577.2.4 loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData

8.577.2.5 loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo

8.577.2.6 uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult

8.578 pack_loc_EventRegister_t Struct Reference

Data Fields

- uint64_t eventRegister
- uint16_t Tlvresult

8.578.1 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.

8.578.2 Field Documentation

8.578.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

8.578.2.2 uint16_t pack_loc_EventRegister_t::TlvResult

8.579 pack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint32_t extPowerState
- uint16_t TlvResult

8.579.1 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.

8.579.2 Field Documentation

8.579.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

8.579.2.2 uint16_t pack_loc_SetExtPowerState_t::TlvResult

8.580 pack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint32_t mode
- uint16_t TlvResult

8.580.1 Detailed Description

This structure contains Set Operation Mode pack

Parameters

mode	• 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
Tlvresult	• Pack result.

8.580.2 Field Documentation

8.580.2.1 uint32_t pack_loc_SetOperationMode_t::mode

8.580.2.2 uint16_t pack_loc_SetOperationMode_t::Tlvresult

8.581 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint32_t xid
- uint16_t Tlvresult

8.581.1 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.

8.581.2 Field Documentation

8.581.2.1 uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.581.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

8.582 pack_loc_SLQSLOCGetServer_t Struct Reference

Data Fields

- uint32_t serverType
- int has_serverAddrTypeMask
- uint8_t serverAddrTypeMask

8.582.1 Detailed Description

This structure contains parameters to Gets the location server. Please check has_<Param_Name> field for presence of optional parameters

Parameters

serverType	• Type of server. • Valid values – eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE – eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC – eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP – eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4)- Server type is custom PDE
serverAddrTypeMask	• Optional parameter • Type of address the client wants. If unspecified, the indication will contain all the types of addresses it has for the specified server type. • Valid bitmasks – 0x01 – IPv4 – 0x02 – IPv6 – 0x04 – URL

8.582.2 Field Documentation

8.582.2.1 int pack_loc_SLQSLOCGetServer_t::has_serverAddrTypeMask

8.582.2.2 uint8_t pack_loc_SLQSLOCGetServer_t::serverAddrTypeMask

8.582.2.3 uint32_t pack_loc_SLQSLOCGetServer_t::serverType

8.583 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

8.583.1 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
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
altitudeSrcInfo	• Optional parameter – Pointer to struct altitudeSrcInfo. See altitudeSrcInfo for more information
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.

rawHorUncCircular	• Optional parameter • Horizontal position uncertainty (circular) without any optimization. • Units - Meters
rawHorConfidence	• 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.

8.583.2 Field Documentation

8.583.2.1 altSrcInfo_t pack_loc_SLQSLOCInjectPosition_t::altitudeSrcInfo

8.583.2.2 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtEllipsoid

8.583.2.3 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtMeanSeaLevel

8.583.2.4 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeSrcInfo

8.583.2.5 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtEllipsoid

8.583.2.6 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtMeanSeaLevel

8.583.2.7 int pack_loc_SLQSLOCInjectPosition_t::has_horConfidence

8.583.2.8 int pack_loc_SLQSLOCInjectPosition_t::has_horReliability

8.583.2.9 int pack_loc_SLQSLOCInjectPosition_t::has_horUncCircular

8.583.2.10 int pack_loc_SLQSLOCInjectPosition_t::has_latitude

8.583.2.11 int pack_loc_SLQSLOCInjectPosition_t::has_longitude

8.583.2.12 int pack_loc_SLQSLOCInjectPosition_t::has_positionSrc

8.583.2.13 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorConfidence

8.583.2.14 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorUncCircular

8.583.2.15 int pack_loc_SLQSLOCInjectPosition_t::has_timestampAge

8.583.2.16 int pack_loc_SLQSLOCInjectPosition_t::has_timestampUtc

8.583.2.17 int pack_loc_SLQSLOCInjectPosition_t::has_vertConfidence

8.583.2.18 int pack_loc_SLQSLOCInjectPosition_t::has_vertRelicability

8.583.2.19 int pack_loc_SLQSLOCInjectPosition_t::has_vertUnc

8.583.2.20 uint8_t pack_loc_SLQSLOCInjectPosition_t::horConfidence

8.583.2.21 uint32_t pack_loc_SLQSLOCInjectPosition_t::horReliability

8.583.2.22 float pack_loc_SLQSLOCInjectPosition_t::horUncCircular

8.583.2.23 double pack_loc_SLQSLOCInjectPosition_t::latitude

8.583.2.24 double pack_loc_SLQSLOCInjectPosition_t::longitude

8.583.2.25 uint32_t pack_loc_SLQSLOCInjectPosition_t::positionSrc

8.583.2.26 uint8_t pack_loc_SLQSLOCInjectPosition_t::rawHorConfidence

8.583.2.27 float pack_loc_SLQSLOCInjectPosition_t::rawHorUncCircular

8.583.2.28 uint32_t pack_loc_SLQSLOCInjectPosition_t::timestampAge

8.583.2.29 uint64_t pack_loc_SLQSLOCInjectPosition_t::timestampUtc

8.583.2.30 uint8_t pack_loc_SLQSLOCInjectPosition_t::vertConfidence

8.583.2.31 uint32_t pack_loc_SLQSLOCInjectPosition_t::vertReliability

8.583.2.32 float pack_loc_SLQSLOCInjectPosition_t::vertUnc

8.584 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

8.584.1 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	• 3-Axis Accelerometer Data (Optional parameter) • Pointer to struct sensorData. See sensorData_t for more information
gyroData	• 3-Axis Cyroscope Data (Optional parameter) • Pointer to struct sensorData. See sensorData_t 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	• Accelerometer Temperature Data (Optional parameter) • Pointer to struct tempratureData. See tempData_t for more information
gyroTemp	• Gyroscope Temperature Data (Optional parameter) • Pointer to struct tempratureData. See tempData_t for more information

8.584.2 Field Documentation

8.584.2.1 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData

8.584.2.2 uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc

8.584.2.3 tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp

8.584.2.4 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::gyroData

8.584.2.5 tempData_t pack_loc_SLQSLOCInjectSensorData_t::gyroTemp

8.584.2.6 uint32_t pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc

8.584.2.7 int pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc

8.584.2.8 int pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp

8.584.2.9 int pack_loc_SLQSLOCInjectSensorData_t::has_accleroData

8.584.2.10 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroData

8.584.2.11 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp

8.584.2.12 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc

8.584.2.13 int pack_loc_SLQSLOCInjectSensorData_t::has_opaqueId

8.584.2.14 uint32_t pack_loc_SLQSLOCInjectSensorData_t::opaqueId

8.585 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference

Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec

8.585.1 Detailed Description

Parameters

timeMsec	• The UTC time since Jan. 1, 1970
timeUncMsec	• The time Uncertainty

8.585.2 Field Documentation

8.585.2.1 uint64_t pack_loc_SLQSLOCInjectUTCTime_t::timeMsec

8.585.2.2 uint32_t pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec

8.586 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference

Data Fields

- uint32_t state
- int has_confidence
- uint8_t confidence

8.586.1 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

8.586.2 Field Documentation

8.586.2.1 uint8_t pack_loc_SLQSLOCSetCradleMountConfig_t::confidence

8.586.2.2 int pack_loc_SLQSLOCSetCradleMountConfig_t::has_confidence

8.586.2.3 uint32_t pack_loc_SLQSLOCSetCradleMountConfig_t::state

8.587 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

8.587.1 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	• 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	• 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	• 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	• Optional Parameter • Minimum time interval, specified by the control point, that must elapse between position reports. • Units - Milliseconds • Default - 1000 ms
LocApplication-Info	• Optional Parameter • LOC Application Parameters • See loc_LocApplicationInfo for more information
pConfigAltitude-Assumed	• 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
Tlvresult	• Unpack result.

8.587.2 Field Documentation

8.587.2.1 loc_LocApplicationInfo* pack_loc_Start_t::pApplicationInfo

8.587.2.2 uint32_t* pack_loc_Start_t::pConfigAltitudeAssumed

8.587.2.3 uint32_t* pack_loc_Start_t::pHorizontalAccuracyLvl

8.587.2.4 uint32_t* pack_loc_Start_t::pIntermediateReportState

8.587.2.5 uint32_t* pack_loc_Start_t::pMinIntervalTime

8.587.2.6 uint32_t* pack_loc_Start_t::pRecurrenceType

8.587.2.7 uint8_t pack_loc_Start_t::SessionId

8.587.2.8 uint16_t pack_loc_Start_t::Tlvresult

8.588 pack_loc_Stop_t Struct Reference

Data Fields

- uint8_t SessionId
- uint16_t Tlvresult

8.588.1 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.

8.588.2 Field Documentation

8.588.2.1 uint8_t pack_loc_Stop_t::SessionId

8.588.2.2 uint16_t pack_loc_Stop_t::Tlvresult

8.589 pack_nas_InitiateDomainAttach_t Struct Reference

Data Fields

- uint32_t action

8.589.1 Detailed Description

Pack structure for Initiates a domain attach/detach of the device.

Parameters

action	• Domain action to attempt 1 - Attach 2 - Detach

8.589.2 Field Documentation

8.589.2.1 uint32_t pack_nas_InitiateDomainAttach_t::action

8.590 pack_nas_SetACCOLC_t Struct Reference

Data Fields

- int8_t spc [6]
- uint8_t accolc

8.590.1 Detailed Description

Sets the access overload class (ACCOLC)

Parameters

spc	• service programming code NULL-terminated string of six digit
accolc	• ACCOLC : Valid range is 0 to 15

8.590.2 Field Documentation

8.590.2.1 uint8_t pack_nas_SetACCOLC_t::accolc

8.590.2.2 int8_t pack_nas_SetACCOLC_t::spc[6]

8.591 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

8.591.1 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

8.591.2 Field Documentation

8.591.2.1 uint32_t* pack_nas_SetCDMANetworkParameters_t::pApplication

8.591.2.2 uint32_t* pack_nas_SetCDMANetworkParameters_t::pBroadcast

8.591.2.3 uint8_t* pack_nas_SetCDMANetworkParameters_t::pCustomSCP

8.591.2.4 uint8_t* pack_nas_SetCDMANetworkParameters_t::pForceRev0

8.591.2.5 uint32_t* pack_nas_SetCDMANetworkParameters_t::pProtocol

8.591.2.6 uint32_t* pack_nas_SetCDMANetworkParameters_t::pRoaming

8.591.2.7 char* pack_nas_SetCDMANetworkParameters_t::pSPC

8.592 pack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint32_t TechnologyPref
- uint32_t Duration
- uint16_t Tlvresult

8.592.1 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

8.592.2 Field Documentation

8.592.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

8.592.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

8.592.2.3 uint16_t pack_nas_SetNetworkPreference_t::TlvResult

8.593 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

8.593.1 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

pTDSCDMASIN-RCONFThresh	• TD-SCDMA SINR Threshold List • See nas_TDSCDMASINRCONFThresh for more details

8.593.2 Field Documentation

8.593.2.1 nas_ECIOThresh∗ pack_nas_SLQSConfigSigInfo_t::pECIOThresh

8.593.2.2 nas_HDRSINRThresh∗ pack_nas_SLQSConfigSigInfo_t::pHDRSINRThresh

8.593.2.3 nas_IOThresh∗ pack_nas_SLQSConfigSigInfo_t::pIOThresh

8.593.2.4 nas_LTESigRptCfg∗ pack_nas_SLQSConfigSigInfo_t::pLTESigRptCfg

8.593.2.5 nas_LTESNRThresh∗ pack_nas_SLQSConfigSigInfo_t::pLTESNRThresh

8.593.2.6 nas_RSRPThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRPThresh

8.593.2.7 nas_RSRQThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRQThresh

8.593.2.8 nas_RSSIThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSSIThresh

8.593.2.9 nas_TDSCDMASINRCONFThresh∗ pack_nas_SLQSConfigSigInfo_t::pTDSCDMASINRCONFThresh

8.594 pack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- uint8_t ∗ pMncPcsStatus

8.594.1 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.

8.594.2 Field Documentation

8.594.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

8.594.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

8.594.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

8.595 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference

Data Fields

- uint32_t regAction
- nas_MNRInfo * pMNRInfo
- uint32_t * pChangeDuration
- uint8_t * pMncPcsDigitStatus

8.595.1 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.

8.595.2 Field Documentation

8.595.2.1 uint32_t* pack_nas_SLQSInitiateNetworkRegistration_t::pChangeDuration

8.595.2.2 uint8_t* pack_nas_SLQSInitiateNetworkRegistration_t::pMncPcsDigitStatus

8.595.2.3 nas_MNRInfo* pack_nas_SLQSInitiateNetworkRegistration_t::pMNRInfo

8.595.2.4 uint32_t pack_nas_SLQSInitiateNetworkRegistration_t::regAction

8.596 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

8.596.1 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 nas_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 nas_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.

Note

Mixture of threshold and delta values can be provided in the request. But for each type and RAT,only one of threshold list or delta value is to be provided.

8.596.2 Field Documentation

8.596.2.1 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta

8.596.2.2 nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh

8.596.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta

8.596.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh

8.596.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta

8.596.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh

8.596.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta

8.596.2.8 nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh

8.596.2.9 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta

8.596.2.10 nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh

8.596.2.11 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIDelta

8.596.2.12 nas_HDRRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIThresh

8.596.2.13 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRDelta

8.596.2.14 nas_HDRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRThresh

8.596.2.15 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta

8.596.2.16 nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh

8.596.2.17 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta

8.596.2.18 nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh

8.596.2.19 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta

8.596.2.20 nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh

8.596.2.21 nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig

8.596.2.22 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta

8.596.2.23 nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh

8.596.2.24 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta

8.596.2.25 nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh

8.596.2.26 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta

8.596.2.27 nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh

8.596.2.28 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta

8.596.2.29 nas_TDSCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh

8.596.2.30 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta

8.596.2.31 nas_TDSCDMASINRThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh

8.596.2.32 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIODelta

8.596.2.33 nas_WCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh

8.596.2.34 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta

8.596.2.35 nas_WCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh

8.597 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference

Data Fields

- uint8_t namID

8.597.1 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.

8.597.2 Field Documentation

8.597.2.1 uint8_t pack_nas_SLQSNasGet3GPP2Subscription_t::namID

8.598 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- uint8_t radio_if

8.598.1 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

8.598.2 Field Documentation

8.598.2.1 uint8_t pack_nas_SLQSNasGetTxRxInfo_t::radio_if

8.599 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

8.599.1 Detailed Description

Parameters

pSystemSelectionInd	[Optional] • System Selection Preference indication registration. The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pDDTMInd	[Optional] • DDTM (Data Dedicated Transmission Mode) indication registration. The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pServingSystemInd	[Optional] • Serving System indication registration. The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pDualStandByPrefInd	[Optional] • Dual Standby Preference indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pSubscriptionInfoInd	[Optional] • Subscription Information indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pNetworkTimeInd	[Optional] • Network Time indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pSysInfoInd	[Optional] • System Information indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable

pSignalStrength-Ind	[Optional] • Signal Strength indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pErrorRateInd	[Optional] • Error Rate indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pHDRNewUATI-AssInd	[Optional] • HDR New UATI Assigned indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pHDRSession-CloseInd	[Optional] • HDR Session Closed indication registration.The callbacks would not be invoked if the indication is disabled. – 0x00 - Disable – 0x01 - Enable
pManaged-RoamingInd	[Optional] • Managed Roaming indication registration.The callbacks would not be invoked if the indication is disabled. – 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. – 0x00 - Disable (default value) – 0x01 - Enable

8.599.2 Field Documentation

8.599.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.599.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.599.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.599.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

8.599.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

8.599.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

8.599.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

8.599.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkRejectInd

8.599.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

8.599.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

8.599.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

8.599.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

8.599.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSuppressSysInfoInd

8.599.2.14 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

8.599.2.15 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

8.600 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

8.600.1 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

8.600.2 Field Documentation

8.600.2.1 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsDl

8.600.2.2 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsUl

8.600.2.3 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmDl

8.600.2.4 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmUl

8.600.2.5 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmDl

8.600.2.6 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmUl

8.600.2.7 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pRankIndicatorInd

8.600.2.8 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pTimer

8.601 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference

Data Fields

- nas_wcdmaUARFCN * pWcdmaUARFCN
- nas_lteEARFCN * pLteEARFCN
- nas_ltePCI * pLtePCI

8.601.1 Detailed Description

Structure for ChannelLock pack.

Parameters

pWcdmaUARFCN	[Optional] • See nas_wcdmaUARFCN for more information
pLteEARFCN	[Optional] • See nas_lteEARFCN for more information
pLtePCI	[Optional] • See nas_ltePCI for more information

8.601.2 Field Documentation

8.601.2.1 nas_lteEARFCN∗ pack_nas_SLQSNASSwiSetChannelLock_t::pLteEARFCN

8.601.2.2 nas_ltePCI∗ pack_nas_SLQSNASSwiSetChannelLock_t::pLtePCI

8.601.2.3 nas_wcdmaUARFCN∗ pack_nas_SLQSNASSwiSetChannelLock_t::pWcdmaUARFCN

8.602 pack_nas_SLQSSetBandPreference_t Struct Reference

Data Fields

- uint64_t bandPref

8.602.1 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

8.602.2 Field Documentation

8.602.2.1 uint64_t pack_nas_SLQSSetBandPreference_t::bandPref

8.603 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

Data Fields

- uint8_t bEnable
- nas_SLQSSignalStrengthsIndReq * pSigIndReq

8.603.1 Detailed Description

Parameters

bEnable	0/1 to disable/enable RSSI signal stregth indication
pSigIndReq	parameters to control signal strength indication

8.603.2 Field Documentation

8.603.2.1 uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable

8.603.2.2 nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq

8.604 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

8.604.1 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 nas_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
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 nas_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 nas_CSGID for more information
pRAT	• Optional parameter Radio Access Technology order Preference • Values: – 0x04 - GSM – 0x05 - UMTS – 0x08 - LTE – 0x09 - TDSCDMA

8.604.2 Field Documentation

8.604.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

8.604.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

8.604.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

8.604.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

8.604.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

8.604.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

8.604.2.7 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref

8.604.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat

8.604.2.9 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref

8.604.2.10 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref

8.604.2.11 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref

8.604.2.12 unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT

8.604.2.13 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref

8.604.2.14 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref

8.604.2.15 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction

8.604.2.16 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdscdmaBandPref

8.605 pack_nas_SLQSSwiPSDetach_t Struct Reference

Data Fields

- uint8_t * pDetachAction

8.605.1 Detailed Description

Structure for PS connection detach pack.

Parameters

pDetachAction[IN]	• Values – 2- Initiates an immediate packet domain detach.

8.605.2 Field Documentation

8.605.2.1 uint8_t* pack_nas_SLQSSwiPSDetach_t::pDetachAction

8.606 pack_pds_PDSInjectTimeReference_t Struct Reference

Data Fields

- uint64_t systemTime
- uint16_t systemDiscontinuities

8.606.1 Detailed Description

Structure contain the parameter for PDS Inject time.

Parameters

systemTime	• System time(milliseconds)
systemDiscontinuities	• Number of system time discontinuities

8.606.2 Field Documentation

8.606.2.1 uint16_t pack_pds_PDSInjectTimeReference_t::systemDiscontinuities

8.606.2.2 uint64_t pack_pds_PDSInjectTimeReference_t::systemTime

8.607 pack_pds_ResetPDSData_t Struct Reference

Data Fields

- uint32_t * pGPSDataMask
- uint32_t * pCellDataMask

8.607.1 Detailed Description

Structure contain the parameter to Resets the specified PDS data.

Parameters

pGPSDataMask	• 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	• 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

8.607.2 Field Documentation

8.607.2.1 uint32_t* pack_pds_ResetPDSData_t::pCellDataMask

8.607.2.2 uint32_t* pack_pds_ResetPDSData_t::pGPSDataMask

8.608 pack_pds_SetEventReportCallback_t Struct Reference

Data Fields

- uint8_t posDataNmea
- uint8_t rptPosData

8.608.1 Detailed Description

Structure contain the parameter for the automatic tracking configuration for the NMEA COM port.

Parameters

posDataNmea	• Report new position data in NMEA format. • Values – 0x00 - Disable – 0x01 - Enable
rptPosData	• Report new position data in raw format. • Values – 0x00 - Disable – 0x01 - Enable

8.608.2 Field Documentation

8.608.2.1 uint8_t pack_pds_SetEventReportCallback_t::posDataNmea

8.608.2.2 uint8_t pack_pds_SetEventReportCallback_t::rptPosData

8.609 pack_pds_SetPDSDefaults_t Struct Reference

Data Fields

- uint32_t operation
- uint8_t timeout
- uint32_t interval
- uint32_t accuracy

8.609.1 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)

8.609.2 Field Documentation

8.609.2.1 uint32_t pack_pds_SetPDSDefaults_t::accuracy

8.609.2.2 uint32_t pack_pds_SetPDSDefaults_t::interval

8.609.2.3 uint32_t pack_pds_SetPDSDefaults_t::operation

8.609.2.4 uint8_t pack_pds_SetPDSDefaults_t::timeout

8.610 pack_pds_SetPDSState_t Struct Reference

Data Fields

- uint32_t enable

8.610.1 Detailed Description

Structure contain PDS state parameters.

Parameters

enable	• Desired PDS state – Zero - disable – Non-Zero - enable

8.610.2 Field Documentation

8.610.2.1 uint32_t pack_pds_SetPDSState_t::enable

8.611 pack_pds_SetPortAutomaticTracking_t Struct Reference

Data Fields

- uint32_t bAuto

8.611.1 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

8.611.2 Field Documentation

8.611.2.1 uint32_t pack_pds_SetPortAutomaticTracking_t::bAuto

8.612 pack_pds_SetServiceAutomaticTracking_t Struct Reference

Data Fields

- uint32_t bAuto

8.612.1 Detailed Description

Structure contain the parameter to Sets the automatic tracking state.

Parameters

bAuto	• Automatic tracking session started for service – 0x00 - Disabled – 0x01 - Enabled

8.612.2 Field Documentation

8.612.2.1 uint32_t pack_pds_SetServiceAutomaticTracking_t::bAuto

8.613 pack_pds_SetXTRAAutomaticDownload_t Struct Reference

Data Fields

- uint32_t bEnabled
- uint16_t interval

8.613.1 Detailed Description

Structure contain the parameter to set the XTRA automatic database download configuration.

Parameters

bEnabled	• Automatic XTRA download status – 0 - Disabled – 1 - Enabled

interval	• Interval (hours) between XTRA downloads

8.613.2 Field Documentation

8.613.2.1 uint32_t pack_pds_SetXTRAAutomaticDownload_t::bEnabled

8.613.2.2 uint16_t pack_pds_SetXTRAAutomaticDownload_t::interval

8.614 pack_pds_SetXTRANetwork_t Struct Reference

Data Fields

- uint32_t preference

8.614.1 Detailed Description

Structure contain the parameter to Sets the XTRA WWAN network preference

Parameters

preference	• 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

8.614.2 Field Documentation

8.614.2.1 uint32_t pack_pds_SetXTRANetwork_t::preference

8.615 pack_pds_SLQSGetAGPSConfig_t Struct Reference

Data Fields

- uint8_t * pNetworkMode

8.615.1 Detailed Description

Structure contain the parameter to PDS AGPS (MS-based) configuration pack.

Parameters

pNetworkMode	• Network Mode of AGPS Server [optional - should be present in Multimode Systems] – 0x00 - UMTS – 0x01 - CDMA

8.615.2 Field Documentation

8.615.2.1 uint8_t* pack_pds_SLQSGetAGPSConfig_t::pNetworkMode

8.616 pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference

Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec
- uint8_t timeBase
- uint8_t forceFlag

8.616.1 Detailed Description

Structure contain the parameter to Injects a absolute time reference.

Parameters

timeMsec	• Represents the number of milliseconds elapsed since either a GPS or UTC time base. If the time base is UTC, this value should NOT include leap seconds
timeUncMsec	• Time uncertainty in milliseconds
timeBase	• Time base – 0x00 - GPS (midnight, Jan 6, 1980) – 0x01 - UTC (midnight, Jan 1, 1970)
forceFlag	• Force acceptance of data

8.616.2 Field Documentation

8.616.2.1 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::forceFlag

8.616.2.2 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeBase

8.616.2.3 uint64_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeMsec

8.616.2.4 uint32_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeUncMsec

8.617 pack_pds_SLQSPDSInjectPositionData_t Struct Reference

Data Fields

- uint64_t * pTimeStamp
- uint64_t * pLatitude
- uint64_t * pLongitude
- uint32_t * pAltitudeWrtEllipsoid

- uint32_t * pAltitudeWrtSealevel
- uint32_t * pHorizontalUncCircular
- uint32_t * pVerticalUnc
- uint8_t * pHorizontalConfidence
- uint8_t * pVerticalConfidence
- uint8_t * pPositionSource
- uint8_t * pTimeType

8.617.1 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

8.617.2 Field Documentation

8.617.2.1 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtEllipsoid

8.617.2.2 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtSealevel

8.617.2.3 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalConfidence

8.617.2.4 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalUncCircular

8.617.2.5 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLatitude

8.617.2.6 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLongitude

8.617.2.7 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pPositionSource

8.617.2.8 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeStamp

8.617.2.9 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeType

8.617.2.10 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalConfidence

8.617.2.11 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalUnc

8.618 pack_pds_SLQSSetAGPSConfig_t Struct Reference

Data Fields

- uint32_t * pServerAddress
- uint32_t * pServerPort
- uint8_t * pServerURL
- uint8_t * pServerURLLength
- uint8_t * pNetworkMode

8.618.1 Detailed Description

Structure contain the parameter to Sets the PDS AGPS (MS-based) configuration.

Parameters

pServerAddress	• IPv4 address of AGPS server [optional]
pServerPort	• Port number of AGPS server [optional - should be present when pServerAddress is present]
pServerURL	• URL of the AGPS server [optional]
pServerURL-Length	• URL length of AGPS server [optional - should be present when pServerURL is present]
pNetworkMode	• Network Mode of AGPS Server [optional - should be present in Multimode Systems] – 0x00 - UMTS – 0x01 - CDMA

8.618.2 Field Documentation

8.618.2.1 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pNetworkMode

8.618.2.2 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerAddress

8.618.2.3 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerPort

8.618.2.4 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURL

8.618.2.5 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURLLength

8.619 pack_pds_SLQSSetPositionMethodState_t Struct Reference

Data Fields

- uint8_t * pXtraTimeState
- uint8_t * pXtraDataState
- uint8_t * pWifiState

8.619.1 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

8.619.2 Field Documentation

8.619.2.1 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pWifiState

8.619.2.2 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraDataState

8.619.2.3 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraTimeState

8.620 pack_pds_StartPDSTrackingSessionExt_t Struct Reference

Data Fields

- uint8_t sessionControl
- uint8_t sessionType
- uint8_t sessionOperation
- uint8_t sessionServerOption
- uint8_t fixTimeout
- uint32_t fixInterval
- uint32_t fixCount
- uint32_t fixAccuracy

8.620.1 Detailed Description

Structure contain the parameter for PDS tracking session.

Parameters

sessionControl	• Control method: – 0x0 - Manual

sessionType	• Type: – 0x0 - New
session-Operation	• Operating mode: – 0x00 - Standalone – 0x01 - MS-based
sessionServer-Option	• Location server option: – 0x0 - Default
fixTimeout	• Maximum time to work on each fix (in seconds, max 255)
fixCount	• Count of position fix requests for this session (must be at least 1)
fixInterval	• interval between position fix requests (in seconds)
fixAccuracy	• Preferred accuracy threshold(in meters)

8.620.2 Field Documentation

8.620.2.1 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixAccuracy

8.620.2.2 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixCount

8.620.2.3 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixInterval

8.620.2.4 uint8_t pack_pds_StartPDSTrackingSessionExt_t::fixTimeout

8.620.2.5 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionControl

8.620.2.6 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionOperation

8.620.2.7 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionServerOption

8.620.2.8 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionType

8.621 pack_qmi_t Struct Reference

Data Fields

- uint16_t xid
- int timeout
- uint16_t msgid
- uint8_t svc

8.621.1 Detailed Description

qmi request context

Parameters

in	*xid*	transaction id
out	*timeout*	timeout recommended in seconds
out	*msgid*	message id
out	*svc*	qmi service

8.621.2 Field Documentation

8.621.2.1 uint16_t pack_qmi_t::msgid

8.621.2.2 uint8_t pack_qmi_t::svc

8.621.2.3 int pack_qmi_t::timeout

8.621.2.4 uint16_t pack_qmi_t::xid

8.622 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

Data Fields

- uint32_t apnId

8.622.1 Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

Parameters

apnId[IN]	• APN id

8.622.2 Field Documentation

8.622.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.623 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

Data Fields

- uint32_t apnId

8.623.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

Parameters

apnId[IN]	• APN id

8.623.2 Field Documentation

8.623.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

8.624 pack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- uint8_t enable

8.624.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

Parameters

`in`	*enable*	• 1 - Enable QoS event reporting • 0 - Disable QoS event reporting

8.624.2 Field Documentation

8.624.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

8.625 pack_rms_SetSMSWake_t Struct Reference

Data Fields

- uint32_t enabled
- uint32_t wake_mask

8.625.1 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)

8.625.2 Field Documentation

8.625.2.1 uint32_t pack_rms_SetSMSWake_t::enabled

8.625.2.2 uint32_t pack_rms_SetSMSWake_t::wake_mask

8.626 pack_sar_SLQSSetRfSarState_t Struct Reference

Data Fields

- uint32_t RfSarState

8.626.1 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

8.626.2 Field Documentation

8.626.2.1 uint32_t pack_sar_SLQSSetRfSarState_t::RfSarState

8.627 pack_sms_SaveSMS_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage

8.627.1 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

8.627.2 Field Documentation

8.627.2.1 uint32_t pack_sms_SaveSMS_t::messageFormat

8.627.2.2 uint32_t pack_sms_SaveSMS_t::messageSize

8.627.2.3 uint8_t* pack_sms_SaveSMS_t::pMessage

8.627.2.4 uint32_t pack_sms_SaveSMS_t::storageType

8.628 pack_sms_SendSMS_t Struct Reference

Data Fields

- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage
- uint8_t * pLinktimer

8.628.1 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

8.628.2 Field Documentation

8.628.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

8.628.2.2 uint32_t pack_sms_SendSMS_t::messageSize

8.628.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

8.628.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

8.629 pack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- enum eqmiCbkSetStatus status

8.629.1 Detailed Description

Parameters

status	callback parameter

8.629.2 Field Documentation

8.629.2.1 enum eqmiCbkSetStatus pack_sms_SetNewSMSCallback_t::status

8.630 pack_sms_SetSMSCAddress_t Struct Reference

Data Fields

- uint8_t * pSMSCAddress
- uint8_t * pSMSCType

8.630.1 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).

8.630.2 Field Documentation

8.630.2.1 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCAddress

8.630.2.2 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCType

8.631 pack_sms_SLQSDeleteSMS_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t * pMessageIndex
- uint32_t * pMessageTag
- uint8_t * pMessageMode

8.631.1 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)

8.631.2 Field Documentation

8.631.2.1 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageIndex

8.631.2.2 uint8_t* pack_sms_SLQSDeleteSMS_t::pMessageMode

8.631.2.3 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageTag

8.631.2.4 uint32_t pack_sms_SLQSDeleteSMS_t::storageType

8.632 pack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint8_t * pMessageMode

8.632.1 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)

8.632.2 Field Documentation

8.632.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

8.632.2.2 uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode

8.632.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

8.633 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

Data Fields

- uint8_t mode

8.633.1 Detailed Description

Parameters

mode	• Mode – 0x00 - CDMA, LTE (if network type is CDMA) – 0x01 - GW, LTE (if network type is UMTS)

8.633.2 Field Documentation

8.633.2.1 uint8_t pack_sms_SLQSGetSmsBroadcastConfig_t::mode

8.634 pack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t * pRequestedTag
- uint8_t * pMessageMode

8.634.1 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)

8.634.2 Field Documentation

8.634.2.1 uint8_t* pack_sms_SLQSGetSMSList_t::pMessageMode

8.634.2.2 uint32_t* pack_sms_SLQSGetSMSList_t::pRequestedTag

8.634.2.3 uint32_t pack_sms_SLQSGetSMSList_t::storageType

8.635 pack_sms_SLQSModifySMSStatus_t Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint32_t messageTag
- uint8_t * pMessageMode

8.635.1 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)

8.635.2 Field Documentation

8.635.2.1 uint32_t pack_sms_SLQSModifySMSStatus_t::messageIndex

8.635.2.2 uint32_t pack_sms_SLQSModifySMSStatus_t::messageTag

8.635.2.3 uint8_t* pack_sms_SLQSModifySMSStatus_t::pMessageMode

8.635.2.4 uint32_t pack_sms_SLQSModifySMSStatus_t::storageType

8.636 pack_sms_SLQSSendAsyncSMS_t Struct Reference

Data Fields

- sms_sendAsyncsmsParams * pSendSmsParams

8.636.1 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

8.636.2 Field Documentation

8.636.2.1 sms_sendAsyncsmsParams* pack_sms_SLQSSendAsyncSMS_t::pSendSmsParams

8.637 pack_sms_SLQSSetIndicationRegister_t Struct Reference

Data Fields

- sms_setIndicationReg * pSetIndicationRegReq

8.637.1 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

8.637.2 Field Documentation

8.637.2.1 sms_setIndicationReg∗ pack_sms_SLQSSetIndicationRegister_t::pSetIndicationRegReq

8.638 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference

Data Fields

- uint8_t mode
- uint8_t broadcastActivate

8.638.1 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

8.638.2 Field Documentation

8.638.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::broadcastActivate

8.638.2.2 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::mode

8.639 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference

Data Fields

- uint8_t mode
- sms_qaQmi3GPPBroadcastCfgInfo ∗ pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo ∗ pCDMABroadcastConfig

8.639.1 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).

8.639.2 Field Documentation

8.639.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastConfig_t::mode

8.639.2.2 sms_qaQmi3GPPBroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pBroadcastConfig

8.639.2.3 sms_qaQmi3GPP2BroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pCDMABroadcastConfig

8.640 pack_sms_SLQSSetSmsStorage_t Struct Reference

Data Fields

- uint8_t smsStorage

8.640.1 Detailed Description

Structure contain Parameter to set the SMS Storage on the device

Parameters

smsStorage	• SMS Storage – 0x01 - device's permanent memory – 0x02 - UICC

8.640.2 Field Documentation

8.640.2.1 uint8_t pack_sms_SLQSSetSmsStorage_t::smsStorage

8.641 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

Data Fields

- sms_maxStorageSizeReq * pMaxStorageSizeReq

8.641.1 Detailed Description

Parameters

pMaxStorage-SizeReq	• Request parameters for SmsSLQSGetMaxStorageSize – See sms_maxStorageSizeReq for more information

8.641.2 Field Documentation

8.641.2.1 sms_maxStorageSizeReq∗ pack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeReq

8.642 pack_sms_SLQSSmsSetRoutes_t Struct Reference

Data Fields

- sms_setRoutesReq ∗ pSetRoutesReq

8.642.1 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

8.642.2 Field Documentation

8.642.2.1 sms_setRoutesReq∗ pack_sms_SLQSSmsSetRoutes_t::pSetRoutesReq

8.643 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

8.643.1 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

8.643.2 Field Documentation

8.643.2.1 uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate

8.643.2.2 uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type

8.643.2.3 uint8_t pack_swiloc_SwiLocSetAutoStart_t::function

8.643.2.4 uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist

8.643.2.5 uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time

8.643.2.6 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate

8.643.2.7 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type

8.643.2.8 int pack_swiloc_SwiLocSetAutoStart_t::set_function

8.643.2.9 int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist

8.643.2.10 int pack_swiloc_SwiLocSetAutoStart_t::set_max_time

8.644 pack_swioma_SLQSOMADMCancelSession_t Struct Reference

Data Fields

- uint32_t sessionType

8.644.1 Detailed Description

Structure that contains the session type for OMA cancel session command

Parameters

in	*sessionType*	• Session type – 0x01 - FOTA, to check availability of FW Update – 0xFF - Cancel any active OMADM session

8.644.2 Field Documentation

8.644.2.1 uint32_t pack_swioma_SLQSOMADMCancelSession_t::sessionType

8.645 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint32_t SessionType

8.645.1 Detailed Description

Structure that contains the session type for OMA get session info command

Parameters

in	*SessionType*	• Session type – 0x01 - FOTA – 0xFF - Any active OMADM session. If no active sessions are available, then previous OMADM session info is returned

8.645.2 Field Documentation

8.645.2.1 uint32_t pack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

8.646 pack_swioma_SLQSOMADMSendSelection_t Struct Reference

Data Fields

- uint32_t selection
- uint32_t * pDeferTime
- uint32_t * pRejectReason

8.646.1 Detailed Description

Structure containing the OMA DM selection

Parameters

in	*selection*	• OMA-DM NIA Selection – 0x01 - Accept – 0x02 - Reject – 0x03 - Defer
in	*pDeferTime*	• 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.
in	*pRejectReason*	• 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.

8.646.2 Field Documentation

8.646.2.1 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pDeferTime

8.646.2.2 uint32_t* pack_swioma_SLQSOMADMSendSelection_t::pRejectReason

8.646.2.3 uint32_t pack_swioma_SLQSOMADMSendSelection_t::selection

8.647 pack_swioma_SLQSOMADMSetSettings_t Struct Reference

Data Fields

- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t * pAutosdm
- uint8_t * pFwAutoCheck

8.647.1 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	• Optional 1 byte parameter indicating OMA Automatic UI Alert Response – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject
pFwAutoCheck	• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response – 0x00 - Disabled – 0x01 - Enabled

8.647.2 Field Documentation

8.647.2.1 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAdownload

8.647.2.2 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAUpdate

8.647.2.3 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pAutosdm

8.647.2.4 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pFwAutoCheck

8.648 pack_swioma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t sessionType

8.648.1 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

8.648.2 Field Documentation

8.648.2.1 uint32_t pack_swioma_SLQSOMADMStartSession_t::sessionType

8.649 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

8.649.1 Detailed Description

This structure contains information of the request parameters associated with a Change PIN API.

Parameters

sessionInfo	• See uim_sessionInformation for more information.
changePIN	• See uim_changeUIMPIN for more information.
pKeyReferenceID(optional)	• Indicates the PIN key reference ID. • 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.

8.649.2 Field Documentation

8.649.2.1 uim_changeUIMPIN pack_uim_ChangePin_t::changePIN

8.649.2.2 uim_encryptedPIN1 pack_uim_ChangePin_t::EncryptedPIN1

8.649.2.3 uint32_t* pack_uim_ChangePin_t::pIndicationToken

8.649.2.4 uint8_t* pack_uim_ChangePin_t::pKeyReferenceID

8.649.2.5 uim_sessionInformation pack_uim_ChangePin_t::sessionInfo

8.649.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

8.650 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

8.650.1 Detailed Description

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

Parameters

sessionInfo	• See uim_sessionInformation for more information.
fileIndex	• See uim_fileInfo for more information.
readTransparent	• See uim_readTransparentInfo for more information.
pIndication-Token(optional)	• Response in Indication. • 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.

8.650.2 Field Documentation

8.650.2.1 uim_FileInfo pack_uim_ReadTransparent_t::fileIndex

8.650.2.2 uint8_t* pack_uim_ReadTransparent_t::pEncryptData

8.650.2.3 uint32_t* pack_uim_ReadTransparent_t::pIndicationToken

8.650.2.4 uim_readTransparentInfo pack_uim_ReadTransparent_t::readTransparent

8.650.2.5 uim_sessionInformation pack_uim_ReadTransparent_t::sessionInfo

8.650.2.6 uint16_t pack_uim_ReadTransparent_t::TlvResult

8.651 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

8.651.1 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.

8.651.2 Field Documentation

8.651.2.1 uim_encryptedPIN1 pack_uim_SetPinProtection_t::EncryptedPIN1

8.651.2.2 uint32_t* pack_uim_SetPinProtection_t::pIndicationToken

8.651.2.3 uim_setPINProtection pack_uim_SetPinProtection_t::pinProtection

8.651.2.4 uint8_t* pack_uim_SetPinProtection_t::pKeyReferenceID

8.651.2.5 uim_sessionInformation pack_uim_SetPinProtection_t::sessionInfo

8.651.2.6 uint16_t pack_uim_SetPinProtection_t::Tlvresult

8.652 pack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_authenticationData authData
- uint32_t * pIndicationToken

8.652.1 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.

8.652.2 Field Documentation

8.652.2.1 uim_authenticationData pack_uim_SLQSUIMAuthenticate_t::authData

8.652.2.2 uint32_t* pack_uim_SLQSUIMAuthenticate_t::pIndicationToken

8.652.2.3 uim_sessionInformation pack_uim_SLQSUIMAuthenticate_t::sessionInfo

8.653 pack_uim_SLQSUIMDepersonalization_t Struct Reference

Data Fields

- uim_depersonalizationInformation depersonilisationInfo

8.653.1 Detailed Description

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

Parameters

depersonilisation-Info	• See uim_depersonalizationInformation for more information.

8.653.2 Field Documentation

8.653.2.1 uim_depersonalizationInformation pack_uim_SLQSUIMDepersonalization_t::depersonilisationInfo

8.654 pack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t eventMask

8.654.1 Detailed Description

Parameters

eventMask	- bit 1 - card status • bit 4 - physical slot status

8.654.2 Field Documentation

8.654.2.1 uint32_t pack_uim_SLQSUIMEventRegister_t::eventMask

8.655 pack_uim_SLQSUIMGetConfiguration_t Struct Reference

Data Fields

- uint32_t * pConfigurationMask

8.655.1 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.

8.655.2 Field Documentation

8.655.2.1 uint32_t* pack_uim_SLQSUIMGetConfiguration_t::pConfigurationMask

8.656 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_fileInfo fileIndex
- uint32_t * pIndicationToken

8.656.1 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.

8.656.2 Field Documentation

8.656.2.1 uim_fileInfo pack_uim_SLQSUIMGetFileAttributes_t::fileIndex

8.656.2.2 uint32_t* pack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

8.656.2.3 uim_sessionInformation pack_uim_SLQSUIMGetFileAttributes_t::sessionInfo

8.657 pack_uim_SLQSUIMPowerDown_t Struct Reference

Data Fields

- uint8_t slot

8.657.1 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

8.657.2 Field Documentation

8.657.2.1 uint8_t pack_uim_SLQSUIMPowerDown_t::slot

8.658 pack_uim_SLQSUIMPowerUp_t Struct Reference

Data Fields

- uint8_t slot
- uint8_t * pIgnoreHotSwapSwitch

8.658.1 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

8.658.2 Field Documentation

8.658.2.1 uint8_t* pack_uim_SLQSUIMPowerUp_t::pIgnoreHotSwapSwitch

8.658.2.2 uint8_t pack_uim_SLQSUIMPowerUp_t::slot

8.659 pack_uim_SLQSUIMRefreshComplete_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uint8_t refreshComplete

8.659.1 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

8.659.2 Field Documentation

8.659.2.1 uint8_t pack_uim_SLQSUIMRefreshComplete_t::refreshComplete

8.659.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshComplete_t::sessionInfo

8.660 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo

8.660.1 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.

8.660.2 Field Documentation

8.660.2.1 uim_sessionInformation pack_uim_SLQSUIMRefreshGetLastEvent_t::sessionInfo

8.661 pack_uim_SLQSUIMRefreshOK_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uint8_t OKtoRefresh

8.661.1 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

8.661.2 Field Documentation

8.661.2.1 uint8_t pack_uim_SLQSUIMRefreshOK_t::OKtoRefresh

8.661.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshOK_t::sessionInfo

8.662 pack_uim_SLQSUIMRefreshRegister_t Struct Reference

Data Fields

- uim_sessionInformation sessionInfo
- uim_registerRefresh regRefresh

8.662.1 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

8.662.2 Field Documentation

8.662.2.1 uim_registerRefresh pack_uim_SLQSUIMRefreshRegister_t::regRefresh

8.662.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshRegister_t::sessionInfo

8.663 pack_uim_SLQSUIMSwitchSlot_t Struct Reference

Data Fields

- uint8_t bLogicalSlot
- uint32_t ulPhysicalSlot

8.663.1 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

8.663.2 Field Documentation

8.663.2.1 uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot

8.663.2.2 uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot

8.664 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

8.664.1 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

8.664.2 Field Documentation

8.664.2.1 uim_encryptedPIN1 pack_uim_UnblockPin_t::EncryptedPIN1

8.664.2.2 uint32_t* pack_uim_UnblockPin_t::pIndicationToken

8.664.2.3 uim_unblockUIMPIN pack_uim_UnblockPin_t::pinProtection

8.664.2.4 uint8_t* pack_uim_UnblockPin_t::pKeyReferenceID

8.664.2.5 uim_sessionInformation pack_uim_UnblockPin_t::sessionInfo

8.664.2.6 uint16_t pack_uim_UnblockPin_t::Tlvresult

8.665 pack_uim_VerifyPin_t Struct Reference

Data Fields

- uim_encryptedPIN1 * pEncryptedPIN1
- uint32_t * pIndicationToken
- uint8_t * pKeyReferenceID
- uim_sessionInformation sessionInfo
- uim_verifyUIMPIN verifyPIN
- uint16_t Tlvresult

8.665.1 Detailed Description

This structure contains information of the request parameters associated with a verify PIN API.

Parameters

sessionInfo	• See uim_sessionInformation for more information.
verifyPIN	• See uim_verifyUIMPIN for more information.
pEncryptedPIN1(optional)	• See uim_encryptedPIN1 for more information.
pKeyReferenceID(optional)	• Indicates the PIN key reference ID. • 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.

8.665.2 Field Documentation

8.665.2.1 uim_encryptedPIN1* pack_uim_VerifyPin_t::pEncryptedPIN1

8.665.2.2 uint32_t* pack_uim_VerifyPin_t::pIndicationToken

8.665.2.3 uint8_t* pack_uim_VerifyPin_t::pKeyReferenceID

8.665.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

8.665.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

8.665.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

8.666 pack_voice_AnswerUSSD_t Struct Reference

Data Fields

- uint8_t * pInfo

8.666.1 Detailed Description

Parameters

pInfo	• USS information • See voice_USSInfo for more details

8.666.2 Field Documentation

8.666.2.1 uint8_t* pack_voice_AnswerUSSD_t::pInfo

8.667 pack_voice_OriginateUSSD_t Struct Reference

Data Fields

- uint8_t * pInfo

8.667.1 Detailed Description

Parameters

pInfo	• USS information • See voice_USSInfo for more details

8.667.2 Field Documentation

8.667.2.1 uint8_t* pack_voice_OriginateUSSD_t::pInfo

8.668 pack_voice_SLQSOriginateUSSD_t Struct Reference

Data Fields

- uint8_t ussDCS
- uint8_t ussLen

- uint8_t ussData [182]

8.668.1 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

8.668.2 Field Documentation

8.668.2.1 uint8_t pack_voice_SLQSOriginateUSSD_t::ussData[182]

8.668.2.2 uint8_t pack_voice_SLQSOriginateUSSD_t::ussDCS

8.668.2.3 uint8_t pack_voice_SLQSOriginateUSSD_t::ussLen

8.669 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference

Data Fields

- uint8_t lineValue

8.669.1 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

8.669.2 Field Documentation

8.669.2.1 uint8_t pack_voice_SLQSVoiceALSSelectLine_t::lineValue

8.670 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference

Data Fields

- uint8_t switchOption

8.670.1 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

8.670.2 Field Documentation

8.670.2.1 uint8_t pack_voice_SLQSVoiceALSSetLineSwitching_t::switchOption

8.671 pack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.671.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

pCallId	• Unique call identifier for the call that must be answered.

8.671.2 Field Documentation

8.671.2.1 uint8_t* pack_voice_SLQSVoiceAnswerCall_t::pCallId

8.672 pack_voice_SLQSVoiceBindSubscription_t Struct Reference

Data Fields

- uint8_t subsType

8.672.1 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

8.672.2 Field Documentation

8.672.2.1 uint8_t pack_voice_SLQSVoiceBindSubscription_t::subsType

8.673 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference

Data Fields

- voice_burstDTMFInfo BurstDTMFInfo
- voice_DTMFLengths * pBurstDTMFLengths

8.673.1 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

8.673.2 Field Documentation

8.673.2.1 voice_burstDTMFInfo pack_voice_SLQSVoiceBurstDTMF_t::BurstDTMFInfo

8.673.2.2 voice_DTMFLengths* pack_voice_SLQSVoiceBurstDTMF_t::pBurstDTMFLengths

8.674 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

8.674.1 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

8.674.2 Field Documentation

8.674.2.1 uint8_t pack_voice_SLQSVoiceDialCall_t::callNumber[81]

8.674.2.2 voice_calledPartySubAdd* pack_voice_SLQSVoiceDialCall_t::pCallPartySubAdd

8.674.2.3 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCallType

8.674.2.4 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCLIRType

8.674.2.5 voice_CUGInfo* pack_voice_SLQSVoiceDialCall_t::pCUGInfo

8.674.2.6 uint8_t* pack_voice_SLQSVoiceDialCall_t::pEmergencyCategory

8.674.2.7 uint8_t* pack_voice_SLQSVoiceDialCall_t::pSvcType

8.674.2.8 voice_UUSInfo* pack_voice_SLQSVoiceDialCall_t::pUUSInFo

8.675 pack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.675.1 Detailed Description

Parameters

pCallId	• Unique call identifier for the call that must be ended

8.675.2 Field Documentation

8.675.2.1 uint8_t* pack_voice_SLQSVoiceEndCall_t::pCallId

8.676 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference

Data Fields

- uint8_t reason

- uint8_t * pSvcClass

8.676.1 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

8.676.2 Field Documentation

8.676.2.1 uint8_t* pack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

8.676.2.2 uint8_t pack_voice_SLQSVoiceGetCallBarring_t::reason

8.677 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference

Data Fields

- uint8_t Reason
- uint8_t * pSvcClass

8.677.1 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.

8.677.2 Field Documentation

8.677.2.1 uint8_t* pack_voice_SLQSVoiceGetCallForwardingStatus_t::pSvcClass

8.677.2.2 uint8_t pack_voice_SLQSVoiceGetCallForwardingStatus_t::Reason

8.678 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference

Data Fields

- uint8_t callID

8.678.1 Detailed Description

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

Parameters

callID	• Call identifier for the call queried for information.

8.678.2 Field Documentation

8.678.2.1 uint8_t pack_voice_SLQSVoiceGetCallInfo_t::callID

8.679 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

Data Fields

- uint8_t * pSvcClass

8.679.1 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

8.679.2 Field Documentation

8.679.2.1 uint8_t* pack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

8.680 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

8.680.1 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.

8.680.2 Field Documentation

8.680.2.1 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAirTimer

8.680.2.2 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAMRStatus

8.680.2.3 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAutoAnswer

8.680.2.4 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pNamID

8.680.2.5 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoicePrivacy

8.680.2.6 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoiceSO

8.680.2.7 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pRoamTimer

8.680.2.8 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pTTYMode

8.680.2.9 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pVoiceDomainPref

8.681 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference

Data Fields

- uint8_t * pRegDTMFEvents
- uint8_t * pRegVoicePrivacyEvents
- uint8_t * pSuppsNotifEvents

8.681.1 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.

8.681.2 Field Documentation

8.681.2.1 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegDTMFEvents

8.681.2.2 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegVoicePrivacyEvents

8.681.2.3 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pSuppsNotifEvents

8.682 pack_voice_SLQSVoiceManageCalls_t Struct Reference

Data Fields

- uint8_t SUPSType

- uint8_t * pCallID

8.682.1 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

8.682.2 Field Documentation

8.682.2.1 uint8_t* pack_voice_SLQSVoiceManageCalls_t::pCallID

8.682.2.2 uint8_t pack_voice_SLQSVoiceManageCalls_t::SUPSType

8.683 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference

Data Fields

- struct voice_USSInfo USSInformation

8.683.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

Parameters

USSInformation	• See voice_USSInfo for more information.

8.683.2 Field Documentation

8.683.2.1 struct voice_USSInfo pack_voice_SLQSVoiceOrigUSSDNoWait_t::USSInformation

8.684 pack_voice_SLQSVoiceSendFlash_t Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t * pFlashPayLd
- uint8_t * pFlashType

8.684.1 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

8.684.2 Field Documentation

8.684.2.1 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pCallID

8.684.2.2 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashPayLd

8.684.2.3 uint8_t* pack_voice_SLQSVoiceSendFlash_t::pFlashType

8.685 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

Data Fields

- uint8_t Reason
- uint8_t oldPasswd [4]
- uint8_t newPasswd [4]

- uint8_t newPasswdAgain [4]

8.685.1 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.

8.685.2 Field Documentation

8.685.2.1 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswd[4]

8.685.2.2 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswdAgain[4]

8.685.2.3 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::oldPasswd[4]

8.685.2.4 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::Reason

8.686 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

8.686.1 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 voice_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.

8.686.2 Field Documentation

8.686.2.1 voice_airTimer* pack_voice_SLQSVoiceSetConfig_t::pAirTimerConfig

8.686.2.2 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pAutoAnswer

8.686.2.3 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceDomain

8.686.2.4 voice_prefVoiceSO* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSO

8.686.2.5 voice_roamTimer* pack_voice_SLQSVoiceSetConfig_t::pRoamTimerConfig

8.686.2.6 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pTTYMode

8.687 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference

Data Fields

- uint8_t privacyPref

8.687.1 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

8.687.2 Field Documentation

8.687.2.1 uint8_t pack_voice_SLQSVoiceSetPreferredPrivacy_t::privacyPref

8.688 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

8.688.1 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 liteServiceClassInformation for more information
pCallBarring-Passwd	• Password is required if call barring is provisioned using a password. Password consists of 4 ASCII digits. Range: 0000 to 9999 (optional)

pCallForwarding-Number	• Call forwarding number to be registered with the network. This has to be included in the request only when the service is set to VOICE_SERVICE_REGISTER. NULL terminated ASCII string. (optional)
pTimerVal	• Call forwarding no reply timer value in seconds. This has to be included in the request only when the service is set to VOICE_SERVICE_REGISTER and the reason is QMI_VOICE_REASON_FWD_NOREPLY. (optional) – 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

8.688.2 Field Documentation

8.688.2.1 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallBarringPasswd

8.688.2.2 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallForwardingNumber

8.688.2.3 voice_callFwdTypeAndPlan* pack_voice_SLQSVoiceSetSUPSService_t::pCallFwdTypeAndPlan

8.688.2.4 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pServiceClass

8.688.2.5 uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pTimerVal

8.688.2.6 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::reason

8.688.2.7 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::voiceSvc

8.689 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t DTMFdigit

8.689.1 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.

8.689.2 Field Documentation

8.689.2.1 uint8_t pack_voice_SLQSVoiceStartContDTMF_t::DTMFdigit

8.689.2.2 uint8_t* pack_voice_SLQSVoiceStartContDTMF_t::pCallID

8.690 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference

Data Fields

- uint8_t callID

8.690.1 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

8.690.2 Field Documentation

8.690.2.1 uint8_t pack_voice_SLQSVoiceStopContDTMF_t::callID

8.691 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference

Data Fields

- uint8_t * pEnableNotification

8.691.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

Parameters

pEnable-Notification	• Enable Notification or not

8.691.2 Field Documentation

8.691.2.1 uint8_t* pack_wds_DHCPv4ClientLeaseChange_t::pEnableNotification

8.692 pack_wds_GetDefaultProfile_t Struct Reference

Data Fields

- uint32_t profiletype

8.692.1 Detailed Description

Parameters

profiletype	profile type

8.692.2 Field Documentation

8.692.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

8.693 pack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t type
- uint8_t family

8.693.1 Detailed Description

Parameters

type	profile type • 0 - 3GPP • 1 - 3GPP2
type	profile family • 0 - Embedded • 1 - Tethered

8.693.2 Field Documentation

8.693.2.1 uint8_t pack_wds_GetDefaultProfileNum_t::family

8.693.2.2 uint8_t pack_wds_GetDefaultProfileNum_t::type

8.694 pack_wds_GetDormancyState_t Struct Reference

8.694.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.695 pack_wds_GetLastMobileIPError_t Struct Reference

8.695.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.696 pack_wds_GetMobileIP_t Struct Reference

8.696.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.697 pack_wds_GetMobileIPProfile_t Struct Reference

Data Fields

- uint8_t index

8.697.1 Detailed Description

Parameters

index	• Mobile IP profile ID

8.697.2 Field Documentation

8.697.2.1 uint8_t pack_wds_GetMobileIPProfile_t::index

8.698 pack_wds_GetPacketStatistics_t Struct Reference

Data Fields

- uint32_t * pStatMask

8.698.1 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

8.698.2 Field Documentation

8.698.2.1 uint32_t* pack_wds_GetPacketStatistics_t::pStatMask

8.699 pack_wds_GetPacketStatus_t Struct Reference

Data Fields

- uint32_t statmask

8.699.1 Detailed Description

Parameters

statmask	packet statistics mask

8.699.2 Field Documentation

8.699.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

8.700 pack_wds_GetSessionDuration_t Struct Reference

8.700.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.701 pack_wds_RMSetTransferStatistics_t Struct Reference

Data Fields

- rmTrasnferStaticsReq RmTrasnferStaticsReq

8.701.1 Detailed Description

Parameters

RmTrasnfer-StaticsReq[IN]	• See rmTrasnferStaticsReq for more information

8.701.2 Field Documentation

8.701.2.1 rmTrasnferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq

8.702 pack_wds_SetAutoconnect_t Struct Reference

Data Fields

- uint8_t acsetting
- uint8_t acroamsetting

8.702.1 Detailed Description

auto connect data session parameters.

Parameters

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

8.702.2 Field Documentation

8.702.2.1 uint8_t pack_wds_SetAutoconnect_t::acroamsetting

8.702.2.2 uint8_t pack_wds_SetAutoconnect_t::acsetting

8.703 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

8.703.1 Detailed Description

Writes the default profile settings to the device. The default profile is used to establish an autoconnect data session.

Parameters

profileType	• Type of profile – 0 - 3GPP – 1 - 3GPP2
pdpType	• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile (optional) – 0 - PDP-IP (IPv4) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
ipAddress	• Preferred IPv4 addr to be assigned to device (optional)
primaryDNS	• Primary DNS Ipv4 address preference (optional)
secondaryDNS	• Secondary DNS Ipv4 address preference (optional)
authentication	• Bitmap that indicates authentication algorithm preference (optional) – 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	• profile Name (optional)
pAPNName	• 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	• Username used during network authentication (optional)
pPassword	• Password used during network authentication (optional)

8.703.2 Field Documentation

8.703.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

8.703.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

8.703.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

8.703.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

8.703.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

8.703.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

8.703.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

8.703.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

8.703.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

8.703.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

8.704 pack_wds_SetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t type
- uint8_t family
- uint8_t index

8.704.1 Detailed Description

This structure to hold Set default profile number

Parameters

type	• 0 - 3GPP • 1 - 3GPP2
family	• 0 - Embedded • 1 - Tethered
index	

8.704.2 Field Documentation

8.704.2.1 uint8_t pack_wds_SetDefaultProfileNum_t::family

8.704.2.2 uint8_t pack_wds_SetDefaultProfileNum_t::index

8.704.2.3 uint8_t pack_wds_SetDefaultProfileNum_t::type

8.705 pack_wds_SetMobileIP_t Struct Reference

Data Fields

- uint32_t mode

8.705.1 Detailed Description

Parameters

mode	• Mobile IP setting – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 2 - Mobile IP only

8.705.2 Field Documentation

8.705.2.1 uint32_t pack_wds_SetMobileIP_t::mode

8.706 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

8.706.1 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

8.706.2 Field Documentation

8.706.2.1 uint8_t* pack_wds_SetMobileIPParameters_t::pHA2002bis

8.706.2.2 uint8_t* pack_wds_SetMobileIPParameters_t::pHAAuthenticator

8.706.2.3 uint32_t* pack_wds_SetMobileIPParameters_t::pMode

8.706.2.4 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegPeriod

8.706.2.5 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegTraffic

8.706.2.6 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryInterval

8.706.2.7 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryLimit

8.706.2.8 char* pack_wds_SetMobileIPParameters_t::pSPC

8.707 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

8.707.1 Detailed Description

Sets the mobile IP parameters.

Parameters

spc	• Six digit service programming code string
index	• Index of the profile to modify
pEnabled	• (Optional) Enable profile? 0 - Disabled Nonzero - Enabled
pAddress	• (Optional) Home IPv4 address
pPrimaryHA	• (Optional) Primary home agent IPv4 address
pSecondaryHA	• (Optional) Secondary home agent IPv4 address
pRevTunneling	• (Optional) Enable reverse tunneling? 0 - Disabled Nonzero - Enabled
pNAI	• (Optional) Network access identifier string
pHASPI	• (Optional) Home agent security parameter index
pAAASPI	• (Optional) AAA server security parameter index
pMNHA	• (Optional) MN-HA key string

pMNAAA	• (Optional) MN-AAA key string

8.707.2 Field Documentation

8.707.2.1 uint8_t pack_wds_SetMobileIPProfile_t::index

8.707.2.2 uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI

8.707.2.3 uint32_t* pack_wds_SetMobileIPProfile_t::pAddress

8.707.2.4 uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled

8.707.2.5 uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI

8.707.2.6 int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA

8.707.2.7 int8_t* pack_wds_SetMobileIPProfile_t::pMNHA

8.707.2.8 int8_t* pack_wds_SetMobileIPProfile_t::pNAI

8.707.2.9 uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA

8.707.2.10 uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling

8.707.2.11 uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA

8.707.2.12 int8_t pack_wds_SetMobileIPProfile_t::spc[10]

8.708 pack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

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

8.708.1 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

8.708.2 Field Documentation

8.708.2.1 wds_profileInfo* pack_wds_SLQSCreateProfile_t::pCurProfile

8.708.2.2 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileId

8.708.2.3 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileType

8.709 pack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileIndex

8.709.1 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)

8.709.2 Field Documentation

8.709.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

8.709.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

8.710 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

8.710.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.711 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

8.711.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.712 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

8.712.1 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

8.712.2 Field Documentation

8.712.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

8.712.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

8.712.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

8.712.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

8.712.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

8.712.2.6 transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

8.713 pack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- uint8_t ProfileId
- uint8_t ProfileType

8.713.1 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

8.713.2 Field Documentation

8.713.2.1 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId

8.713.2.2 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType

8.714 pack_wds_SLQSGetRuntimeSettings_t Struct Reference

Data Fields

- uint32_t * pReqSettings

8.714.1 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 (Not Supported on MC/EM73xx)

8.714.2 Field Documentation

8.714.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

8.715 pack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint8_t * pProfileId
- uint8_t * pProfileType
- wds_profileInfo curProfile

8.715.1 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

8.715.2 Field Documentation

8.715.2.1 wds_profileInfo pack_wds_SLQSModifyProfile_t::curProfile

8.715.2.2 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileId

8.715.2.3 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileType

8.716 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

8.716.1 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

8.716.2 Field Documentation

8.716.2.1 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

8.716.2.2 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease

8.716.2.3 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled

8.716.2.4 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile

8.716.2.5 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList

8.716.2.6 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pProfileList

8.717 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint8_t IPFamilyPreference

8.717.1 Detailed Description

This structure contains set IP family preference information

Parameters

IPFamily-Preference	IP Family preference • PACK_WDS_IPV4 IP Version 4 • PACK_WDS_IPV6 IP Version 6

8.717.2 Field Documentation

8.717.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

8.718 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

8.718.1 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

8.718.2 Field Documentation

8.718.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

8.718.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

8.718.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearerTechExt

8.718.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

8.718.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

8.718.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

8.718.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

8.718.2.8 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

8.719 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wdsDhcpv4ProfileId * pProfileId

8.719.1 Detailed Description

Parameters

pProfileId	• pointer to Profile Id structure

8.719.2 Field Documentation

8.719.2.1 wdsDhcpv4ProfileId* pack_wds_SLQSSGetDHCPv4ClientConfig_t::pProfileId

8.720 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wds_DHCPv4ProfileId * pProfileId
- wds_DHCPv4HWConfig * pHwConfig
- wds_DHCPv4OptionList * pRequestOptionList

8.720.1 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

8.720.2 Field Documentation

8.720.2.1 wds_DHCPv4HWConfig∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pHwConfig

8.720.2.2 wds_DHCPv4ProfileId∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pProfileId

8.720.2.3 wds_DHCPv4OptionList∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pRequestOptionList

8.721 pack_wds_SLQSSSetLoopback_t Struct Reference

Data Fields

- uint8_t loopbackMode
- uint8_t loopbackMultiplier

8.721.1 Detailed Description

Parameters

loopbackMode	• Loopback Mode. – 0 - Disable – 1 - Enable
loopback-Multiplier	• Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.721.2 Field Documentation

8.721.2.1 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMode

8.721.2.2 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMultiplier

8.722 pack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

- uint8_t ∗ pTech
- uint32_t ∗ pprofileid3gpp
- uint32_t ∗ pprofileid3gpp2
- uint32_t ∗ pAuth
- char ∗ pUser
- char ∗ pPass

8.722.1 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 3GPP2 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

8.722.2 Field Documentation

8.722.2.1 uint32_t* pack_wds_SLQSStartDataSession_t::pAuth

8.722.2.2 char* pack_wds_SLQSStartDataSession_t::pPass

8.722.2.3 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp

8.722.2.4 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp2

8.722.2.5 uint8_t* pack_wds_SLQSStartDataSession_t::pTech

8.722.2.6 char* pack_wds_SLQSStartDataSession_t::pUser

8.723 pack_wds_SLQSStopDataSession_t Struct Reference

Data Fields

- uint32_t * psid

8.723.1 Detailed Description

Parameters

sid	session id

8.723.2 Field Documentation

8.723.2.1 uint32_t* pack_wds_SLQSStopDataSession_t::psid

8.724 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

8.724.1 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.

8.724.2 Field Documentation

8.724.2.1 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd

8.724.2.2 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd

8.724.2.3 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd

8.724.2.4 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd

8.724.2.5 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd

8.724.2.6 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd

8.724.2.7 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd

8.724.2.8 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd

8.724.2.9 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd

8.724.2.10 wds_TrStatInd* pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd

8.725 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

- uint8_t contextId
- uint8_t contextType

8.725.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

Parameters

contextId	• Context Identifier
contextType	• Identifies technology type – 0 - 3GPP – 1 - 3GPP2

8.725.2 Field Documentation

8.725.2.1 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.725.2.2 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType

8.726 PackCreateProfileOut Struct Reference

Data Fields

- uint8_t ProfileType
- uint8_t ProfileIndex
- uint16_t ExtErrorCode

8.726.1 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
ExtErrorCode	• The extended error code received from DS Profile subsystem

8.726.2 Field Documentation

8.726.2.1 uint16_t PackCreateProfileOut::ExtErrorCode

8.726.2.2 uint8_t PackCreateProfileOut::ProfileIndex

8.726.2.3 uint8_t PackCreateProfileOut::ProfileType

8.727 packgetDyingGaspCfg Struct Reference

Data Fields

- uint8_t * pDestSMSNum
- uint8_t * pDestSMSContent

8.727.1 Detailed Description

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

Parameters

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

8.727.2 Field Documentation

8.727.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

8.727.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

8.728 packgetDyingGaspStatistics Struct Reference

Data Fields

- uint32_t * pTimeStamp
- uint8_t * pSMSAttemptedFlag

8.728.1 Detailed Description

This structure contains Get Dysing GASP Statistics

Parameters

pTimeStamp[OUT]	• Timestamp of the last time power loss was detected and Dying Gasp feature was triggered. • UTC time in seconds since Jan 06, 1980 (GPS Epoch).

pSMS-AttemptedFlag[-OUT]	• Indicates whether device attempted to send SMS in the last power loss event. • 0 - SMS not attempted • 1 - SMS attempted • This only indicates device sent the SMS but does not guarantee network delivery.

8.728.2 Field Documentation

8.728.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

8.728.2.2 uint32_t* packgetDyingGaspStatistics::pTimeStamp

8.729 PCMparams Struct Reference

Data Fields

- BYTE iFaceTabLen
- BYTE iFaceTab [255]

8.729.1 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

8.729.2 Field Documentation

8.729.2.1 BYTE PCMparams::iFaceTab[255]

8.729.2.2 BYTE PCMparams::iFaceTabLen

8.730 PCSCFFQDNAddress Struct Reference

Data Fields

- WORD fqdnLen
- CHAR fqdnAddr [256]

8.730.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

Parameters

fqdnLen	• length of the received FQDN address
fqdnAddr	• FQDN address(Max 256 characters)

8.730.2 Field Documentation

8.730.2.1 CHAR PCSCFFQDNAddress::fqdnAddr[256]

8.730.2.2 WORD PCSCFFQDNAddress::fqdnLen

8.731 PCSCFFQDNAddressList Struct Reference

Data Fields

- BYTE numInstances
- struct PCSCFFQDNAddress pcsfFQDNAddress [10]

8.731.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

Parameters

numInstances	• Number of FQDN addresses received
pcsfFQDN-Address	• FQDN address information(Max 10 addresses)

8.731.2 Field Documentation

8.731.2.1 BYTE PCSCFFQDNAddressList::numInstances

8.731.2.2 struct PCSCFFQDNAddress PCSCFFQDNAddressList::pcsfFQDNAddress[10]

8.732 PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- BYTE numInstances
- ULONG pscsfIPv4Addr [64]

8.732.1 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)

8.732.2 Field Documentation

8.732.2.1 BYTE PCSCFIPv4ServerAddressList::numInstances

8.732.2.2 ULONG PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

8.733 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

8.733.1 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

8.733.2 Field Documentation

8.733.2.1 ULONG* PDSPositionData::pAltitudeWrtEllipsoid

8.733.2.2 ULONG* PDSPositionData::pAltitudeWrtSealevel

8.733.2.3 BYTE* PDSPositionData::pHorizontalConfidence

8.733.2.4 ULONG* PDSPositionData::pHorizontalUncCircular

8.733.2.5 ULONGLONG* PDSPositionData::pLatitude

8.733.2.6 ULONGLONG* PDSPositionData::pLongitude

8.733.2.7 BYTE* PDSPositionData::pPositionSource

8.733.2.8 ULONGLONG* PDSPositionData::pTimeStamp

8.733.2.9 BYTE* PDSPositionData::pTimeType

8.733.2.10 BYTE* PDSPositionData::pVerticalConfidence

8.733.2.11 ULONG* PDSPositionData::pVerticalUnc

8.734 PDSPosMethodStateReq Struct Reference

Data Fields

- BYTE * pXtraTimeState
- BYTE * pXtraDataState
- BYTE * pWifiState

8.734.1 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

8.734.2 Field Documentation

8.734.2.1 BYTE* PDSPosMethodStateReq::pWifiState

8.734.2.2 BYTE* PDSPosMethodStateReq::pXtraDataState

8.734.2.3 BYTE* PDSPosMethodStateReq::pXtraTimeState

8.735 peerNumberInfo Struct Reference

Data Fields

- BYTE callID
- BYTE numPI
- BYTE numSI
- BYTE numType
- BYTE numPlan
- BYTE numLen
- BYTE number [81]

8.735.1 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.

8.735.2 Field Documentation

8.735.2.1 BYTE peerNumberInfo::callID

8.735.2.2 BYTE peerNumberInfo::number[81]

8.735.2.3 BYTE peerNumberInfo::numLen

8.735.2.4 BYTE peerNumberInfo::numPI

8.735.2.5 BYTE peerNumberInfo::numPlan

8.735.2.6 BYTE peerNumberInfo::numSI

8.735.2.7 BYTE peerNumberInfo::numType

8.736 personalizationStatus Struct Reference

Data Fields

- BYTE numFeatures
- BYTE feature [12]
- BYTE verifyLeft [12]
- BYTE unblockLeft [12]

8.736.1 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.

8.736.2 Field Documentation

8.736.2.1 BYTE personalizationStatus::feature[12]

8.736.2.2 BYTE personalizationStatus::numFeatures

8.736.2.3 BYTE personalizationStatus::unblockLeft[12]

8.736.2.4 BYTE personalizationStatus::verifyLeft[12]

8.737 PhyCaAggPcellInfo Struct Reference

Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- int iLTEbandValue
- BYTE TlvPresent

8.737.1 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.

8.737.2 Field Documentation

8.737.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggPcellInfo::dl_bw_value

8.737.2.2 int PhyCaAggPcellInfo::freq

8.737.2.3 int PhyCaAggPcellInfo::iLTEbandValue

8.737.2.4 int PhyCaAggPcellInfo::pci

8.737.2.5 BYTE PhyCaAggPcellInfo::TlvPresent

8.738 PhyCaAggScellDlBw Struct Reference

Data Fields

- NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- BYTE TlvPresent

8.738.1 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.

8.738.2 Field Documentation

8.738.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellDlBw::dl_bw_value

8.738.2.2 BYTE PhyCaAggScellDlBw::TlvPresent

8.739 PhyCaAggScellIndex Struct Reference

Data Fields

- BYTE scell_idx
- BYTE TlvPresent

8.739.1 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.

8.739.2 Field Documentation

8.739.2.1 BYTE PhyCaAggScellIndex::scell_idx

8.739.2.2 BYTE PhyCaAggScellIndex::TlvPresent

8.740 PhyCaAggScellIndType Struct Reference

Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_SCELL_STATE scell_state
- BYTE TlvPresent

8.740.1 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.

8.740.2 Field Documentation

8.740.2.1 int PhyCaAggScellIndType::freq

8.740.2.2 int PhyCaAggScellIndType::pci

8.740.2.3 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellIndType::scell_state

8.740.2.4 BYTE PhyCaAggScellIndType::TlvPresent

8.741 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

8.741.1 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.

8.741.2 Field Documentation

8.741.2.1 NAS_LTE_CPHY_CA_BW_NRB PhyCaAggScellInfo::dl_bw_value

8.741.2.2 int PhyCaAggScellInfo::freq

8.741.2.3 int PhyCaAggScellInfo::iLTEbandValue

8.741.2.4 int PhyCaAggScellInfo::pci

8.741.2.5 NAS_LTE_CPHY_SCELL_STATE PhyCaAggScellInfo::scell_state

8.741.2.6 BYTE PhyCaAggScellInfo::TlvPresent

8.742 PilotSetData Struct Reference

Data Fields

- BYTE NumPilots
- PilotSetParams * pPilotSetInfo

8.742.1 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.

8.742.2 Field Documentation

8.742.2.1 BYTE PilotSetData::NumPilots

8.742.2.2 PilotSetParams* PilotSetData::pPilotSetInfo

8.743 PilotSetParams Struct Reference

Data Fields

- ULONG PilotType
- WORD PilotPN
- WORD PilotStrength

8.743.1 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)

8.743.2 Field Documentation

8.743.2.1 WORD PilotSetParams::PilotPN

8.743.2.2 WORD PilotSetParams::PilotStrength

8.743.2.3 ULONG PilotSetParams::PilotType

8.744 pktErrRate Struct Reference

Data Fields

- WORD multiplier
- WORD exponent

8.744.1 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)

8.744.2 Field Documentation

8.744.2.1 WORD pktErrRate::exponent

8.744.2.2 WORD pktErrRate::multiplier

8.745 PlmnID Struct Reference

Data Fields

- WORD mcc
- WORD mnc
- BOOL pcsDigit
- BYTE TlvPresent

8.745.1 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

8.745.2 Field Documentation

8.745.2.1 WORD PlmnID::mcc

8.745.2.2 WORD PlmnID::mnc

8.745.2.3 BOOL PlmnID::pcsDigit

8.745.2.4 BYTE PlmnID::TlvPresent

8.746 PLMNNetworkName Struct Reference

Data Fields

- BYTE numInstance
- PLMNNetworkNameData PLMNNetName [255]

8.746.1 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).

8.746.2 Field Documentation

8.746.2.1 BYTE PLMNNetworkName::numInstance

8.746.2.2 PLMNNetworkNameData PLMNNetworkName::PLMNNetName[255]

8.747 PLMNNetworkNameData Struct Reference

Data Fields

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

8.747.1 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.

8.747.2 Field Documentation

8.747.2.1 BYTE PLMNNetworkNameData::codingScheme

8.747.2.2 BYTE PLMNNetworkNameData::countryInitials

8.747.2.3 BYTE PLMNNetworkNameData::longName[255]

8.747.2.4 BYTE PLMNNetworkNameData::longNameLen

8.747.2.5 BYTE PLMNNetworkNameData::longNameSpareBits

8.747.2.6 BYTE PLMNNetworkNameData::shortName[255]

8.747.2.7 BYTE PLMNNetworkNameData::shortNameLen

8.747.2.8 BYTE PLMNNetworkNameData::shortNameSpareBits

8.748 Port Struct Reference

Data Fields

- WORD port
- WORD range

8.748.1 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

8.748.2 Field Documentation

8.748.2.1 WORD Port::port

8.748.2.2 WORD Port::range

8.749 precisionDilution_s Struct Reference

Data Fields

- ULONG PDOP
- ULONG HDOP
- ULONG VDOP

8.749.1 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)

8.749.2 Field Documentation

8.749.2.1 ULONG precisionDilution_s::HDOP

8.749.2.2 ULONG precisionDilution_s::PDOP

8.749.2.3 ULONG precisionDilution_s::VDOP

8.750 PrefImageList Struct Reference

Data Fields

- BYTE listSize
- struct ImageElement listEntries [2]

8.750.1 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

8.750.2 Field Documentation

8.750.2.1 struct ImageElement PrefImageList::listEntries[2]

8.750.2.2 BYTE PrefImageList::listSize

8.751 prefVoiceSO Struct Reference

Data Fields

- BYTE namID
- BYTE evrcCapability
- WORD homePageVoiceSO
- WORD homeOrigVoiceSO
- WORD roamOrigVoiceSO

8.751.1 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

8.751.2 Field Documentation

8.751.2.1 BYTE prefVoiceSO::evrcCapability

8.751.2.2 WORD prefVoiceSO::homeOrigVoiceSO

8.751.2.3 WORD prefVoiceSO::homePageVoiceSO

8.751.2.4 BYTE prefVoiceSO::namID

8.751.2.5 WORD prefVoiceSO::roamOrigVoiceSO

8.752 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

8.752.1 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().

8.752.2 Field Documentation

8.752.2.1 BYTE* Profile3GPP::pAddrAllocPref

8.752.2.2 BYTE* Profile3GPP::pAPNClass

8.752.2.3 BYTE* Profile3GPP::pAPNDisabledFlag

8.752.2.4 CHAR* Profile3GPP::pAPNName

8.752.2.5 WORD* Profile3GPP::pAPNnameSize

8.752.2.6 BYTE* Profile3GPP::pAuthenticationPref

8.752.2.7 struct GPRSRequestedQoS* Profile3GPP::pGPRSMinimumQoS

8.752.2.8 struct GPRSRequestedQoS* Profile3GPP::pGPRSRequestedQos

8.752.2.9 BYTE* Profile3GPP::pImCnFlag

8.752.2.10 ULONG* Profile3GPP::pIPv4AddrPref

8.752.2.11 USHORT* Profile3GPP::pIPv6AddPref

8.752.2.12 CHAR* Profile3GPP::pPassword

8.752.2.13 WORD* Profile3GPP::pPasswordSize

8.752.2.14 BYTE* Profile3GPP::pPcscfAddrUsingDhcp

8.752.2.15 BYTE* Profile3GPP::pPcscfAddrUsingPCO

8.752.2.16 ULONG* Profile3GPP::pPDNInactivTimeout

8.752.2.17 BYTE* Profile3GPP::pPdpAccessConFlag

8.752.2.18 BYTE* Profile3GPP::pPdpContext

8.752.2.19 BYTE* Profile3GPP::pPdpDataCompType

8.752.2.20 BYTE* Profile3GPP::pPdpHdrCompType

8.752.2.21 BYTE* Profile3GPP::pPDPtype

8.752.2.22 ULONG* Profile3GPP::pPriDNSIPv4AddPref

8.752.2.23 USHORT* Profile3GPP::pPriDNSIPv6addpref

8.752.2.24 **BYTE*** Profile3GPP::pPrimaryID

8.752.2.25 **CHAR*** Profile3GPP::pProfilename

8.752.2.26 **WORD*** Profile3GPP::pProfilenameSize

8.752.2.27 struct **QosClassID*** Profile3GPP::pQosClassID

8.752.2.28 **ULONG*** Profile3GPP::pSecDNSIPv4AddPref

8.752.2.29 **USHORT*** Profile3GPP::pSecDNSIPv6addpref

8.752.2.30 **BYTE*** Profile3GPP::pSecondaryFlag

8.752.2.31 struct **TFTIDParams*** Profile3GPP::pTFTID1Params

8.752.2.32 struct **TFTIDParams*** Profile3GPP::pTFTID2Params

8.752.2.33 struct **UMTSQoS*** Profile3GPP::pUMTSMinQoS

8.752.2.34 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSMinQosSigInd

8.752.2.35 struct **UMTSQoS*** Profile3GPP::pUMTSReqQoS

8.752.2.36 struct **UMTSReqQoSSigInd*** Profile3GPP::pUMTSReqQoSSigInd

8.752.2.37 **CHAR*** Profile3GPP::pUsername

8.752.2.38 **WORD*** Profile3GPP::pUsernameSize

8.753 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

8.753.1 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().

8.753.2 Field Documentation

8.753.2.1 BYTE* Profile3GPP2::pAllowLinger

8.753.2.2 BYTE* Profile3GPP2::pAPNClass3GPP2

8.753.2.3 BYTE* Profile3GPP2::pAPNEnabled3GPP2

8.753.2.4 CHAR* Profile3GPP2::pApnString

8.753.2.5 WORD* Profile3GPP2::pApnStringSize

8.753.2.6 BYTE* Profile3GPP2::pAppPriority

8.753.2.7 ULONG* Profile3GPP2::pAppType

8.753.2.8 CHAR* Profile3GPP2::pAuthPassword

8.753.2.9 WORD* Profile3GPP2::pAuthPasswordSize

8.753.2.10 **BYTE*** Profile3GPP2::pAuthProtocol

8.753.2.11 **BYTE*** Profile3GPP2::pAuthRetryCount

8.753.2.12 **USHORT*** Profile3GPP2::pAuthTimeout

8.753.2.13 **BYTE*** Profile3GPP2::pDataMode

8.753.2.14 **BYTE*** Profile3GPP2::pDataRate

8.753.2.15 **USHORT*** Profile3GPP2::plpcpAckTimeout

8.753.2.16 **BYTE*** Profile3GPP2::plpcpCreqRetryCount

8.753.2.17 **BYTE*** Profile3GPP2::plsPcscfAddressNedded

8.753.2.18 **USHORT*** Profile3GPP2::pLcpAckTimeout

8.753.2.19 **BYTE*** Profile3GPP2::pLcpCreqRetryCount

8.753.2.20 **BYTE*** Profile3GPP2::pNegoDnsSrvrPref

8.753.2.21 **ULONG*** Profile3GPP2::pPDNInactivTimeout3GPP2

8.753.2.22 **BYTE*** Profile3GPP2::pPdnType

8.753.2.23 **ULONG*** Profile3GPP2::pPppSessCloseTimer1x

8.753.2.24 **ULONG*** Profile3GPP2::pPppSessCloseTimerDO

8.753.2.25 **ULONG*** Profile3GPP2::pPrimaryV4DnsAddress

8.753.2.26 **USHORT*** Profile3GPP2::pPriV6DnsAddress

8.753.2.27 **BYTE*** Profile3GPP2::pRATType

8.753.2.28 **ULONG*** Profile3GPP2::pSecondaryV4DnsAddress

8.753.2.29 **USHORT*** Profile3GPP2::pSecV6DnsAddress

8.753.2.30 **CHAR*** Profile3GPP2::pUserId

8.753.2.31 **WORD*** Profile3GPP2::pUserIdSize

8.754 ProfileIdentifier Struct Reference

Data Fields

- BYTE profileType
- BYTE profileIndex

8.754.1 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

8.754.2 Field Documentation

8.754.2.1 BYTE ProfileIdentifier::profileIndex

8.754.2.2 BYTE ProfileIdentifier::profileType

8.755 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

8.755.1 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

8.755.2 Field Documentation

8.755.2.1 WORD protocolSubtypeElement::AccessMac

8.755.2.2 WORD protocolSubtypeElement::AuthProt

8.755.2.3 WORD protocolSubtypeElement::ControlMac

8.755.2.4 WORD protocolSubtypeElement::EncryptProt

8.755.2.5 WORD protocolSubtypeElement::ForwardMac

8.755.2.6 WORD protocolSubtypeElement::IdleState

8.755.2.7 WORD protocolSubtypeElement::KeyExchange

8.755.2.8 WORD protocolSubtypeElement::MultDisc

8.755.2.9 WORD protocolSubtypeElement::PhysicalLayer

8.755.2.10 WORD protocolSubtypeElement::ReverseMac

8.755.2.11 WORD protocolSubtypeElement::SecProt

8.755.2.12 WORD protocolSubtypeElement::VirtStream

8.756 PSDetachReq Struct Reference

Data Fields

- BYTE * pDetachAction

8.756.1 Detailed Description

This structure contains information about the SLQSSwiPSDetach request parameters.

Parameters

pDetachAction[IN]	• Values – 2- Initiates an immediate packet domain detach.

8.756.2 Field Documentation

8.756.2.1 BYTE* PSDetachReq::pDetachAction

8.757 qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- BYTE leapSeconds
- BYTE localTimeOffset
- BYTE daylightSavings

8.757.1 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)

8.757.2 Field Documentation

8.757.2.1 BYTE qaQmi3Gpp2TimeZone::daylightSavings

8.757.2.2 BYTE qaQmi3Gpp2TimeZone::leapSeconds

8.757.2.3 BYTE qaQmi3Gpp2TimeZone::localTimeOffset

8.758 qaQmiInterfaceInfo Struct Reference

Data Fields

- BYTE qaQmiinstanceid
- eQaQMIService qaQmisvctype
- ULONG v4sessionId
- ULONG v6sessionId

8.758.1 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

8.758.2 Field Documentation

8.758.2.1 BYTE qaQmiInterfaceInfo::qaQmiinstanceid

8.758.2.2 eQaQMIService qaQmiInterfaceInfo::qaQmisvctype

8.758.2.3 ULONG qaQmiInterfaceInfo::v4sessionId

8.758.2.4 ULONG qaQmiInterfaceInfo::v6sessionId

8.759 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

8.759.1 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

8.759.2 Field Documentation

8.759.2.1 WORD qaQmiServingSystemParam::BasestationID

8.759.2.2 ULONG qaQmiServingSystemParam::BasestationLatitude

8.759.2.3 ULONG qaQmiServingSystemParam::BasestationLongitude

8.759.2.4 callBarStatus qaQmiServingSystemParam::CallBarStatus

8.759.2.5 BYTE qaQmiServingSystemParam::CDMA_P_Rev

8.759.2.6 CDMASysInfoExt qaQmiServingSystemParam::CDMASystemInfoExt

8.759.2.7 ULONG qaQmiServingSystemParam::CellID

8.759.2.8 BYTE qaQmiServingSystemParam::concSvcInfo

8.759.2.9 currentPLMN qaQmiServingSystemParam::CurrentPLMN

8.759.2.10 dataSrvCapabilities qaQmiServingSystemParam::DataSrvCapabilities

8.759.2.11 BYTE qaQmiServingSystemParam::defaultRoamInd

8.759.2.12 detailSvcInfo qaQmiServingSystemParam::DetailedSvcInfo

8.759.2.13 BYTE qaQmiServingSystemParam::DTMInd

8.759.2.14 qaQmi3Gpp2TimeZone qaQmiServingSystemParam::Gpp2TimeZone

8.759.2.15 BYTE qaQmiServingSystemParam::GppNetworkDSTAdjustment

8.759.2.16 BYTE qaQmiServingSystemParam::GppTimeZone

8.759.2.17 BYTE qaQmiServingSystemParam::hdrPersonality

8.759.2.18 WORD qaQmiServingSystemParam::Lac

8.759.2.19 WORD qaQmiServingSystemParam::NetworkID

8.759.2.20 BYTE qaQmiServingSystemParam::PRLInd

8.759.2.21 BYTE qaQmiServingSystemParam::roamIndicatorVal

8.759.2.22 roamIndList qaQmiServingSystemParam::RoamingIndicatorList

8.759.2.23 servSystem qaQmiServingSystemParam::ServingSystem

8.759.2.24 WORD qaQmiServingSystemParam::SystemID

8.759.2.25 WORD qaQmiServingSystemParam::trackAreaCode

8.760 QmiCbkCatEventStatusReportInd Struct Reference

Data Fields

- BYTE event_Index
- struct CatCommonEventTlv CCETlv [11]

8.760.1 Field Documentation

8.760.1.1 struct CatCommonEventTlv QmiCbkCatEventStatusReportInd::CCETlv[11]

8.760.1.2 BYTE QmiCbkCatEventStatusReportInd::event_Index

8.761 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

8.761.1 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

8.761.2 Field Documentation

8.761.2.1 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtEllipsoid

8.761.2.2 ULONG* QmiCbkLocBestAvailPosInd::pAltitudeWrtMeanSeaLevel

8.761.2.3 gpsTime* QmiCbkLocBestAvailPosInd::pGpsTime

8.761.2.4 ULONG* QmiCbkLocBestAvailPosInd::pHeading

8.761.2.5 ULONG* QmiCbkLocBestAvailPosInd::pHeadingUnc

8.761.2.6 BYTE* QmiCbkLocBestAvailPosInd::pHorCirConf

8.761.2.7 BYTE* QmiCbkLocBestAvailPosInd::pHorEllpConf

8.761.2.8 ULONG* QmiCbkLocBestAvailPosInd::pHorReliability

8.761.2.9 ULONG* QmiCbkLocBestAvailPosInd::pHorUncCircular

8.761.2.10 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseOrientAzimuth

8.761.2.11 ULONG* QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMajor

8.761.2.12 **ULONG*** QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMinor

8.761.2.13 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pLatitude

8.761.2.14 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pLongitude

8.761.2.15 **ULONG*** QmiCbkLocBestAvailPosInd::pMagneticDeviation

8.761.2.16 **precisionDilution*** QmiCbkLocBestAvailPosInd::pPrecisionDilution

8.761.2.17 **sensorDataUsage*** QmiCbkLocBestAvailPosInd::pSensorDataUsage

8.761.2.18 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedHorizontal

8.761.2.19 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedUnc

8.761.2.20 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedVertical

8.761.2.21 **ULONG*** QmiCbkLocBestAvailPosInd::pSpeedVerticalUnc

8.761.2.22 **svUsedforFix*** QmiCbkLocBestAvailPosInd::pSvUsedforFix

8.761.2.23 **ULONG*** QmiCbkLocBestAvailPosInd::pTechnologyMask

8.761.2.24 **ULONG*** QmiCbkLocBestAvailPosInd::pTimeSrc

8.761.2.25 **ULONGLONG*** QmiCbkLocBestAvailPosInd::pTimestampUtc

8.761.2.26 **ULONG*** QmiCbkLocBestAvailPosInd::pTimeUnc

8.761.2.27 **BYTE*** QmiCbkLocBestAvailPosInd::pVertConfidence

8.761.2.28 **ULONG*** QmiCbkLocBestAvailPosInd::pVertReliability

8.761.2.29 **ULONG*** QmiCbkLocBestAvailPosInd::pVertUnc

8.761.2.30 **ULONG*** QmiCbkLocBestAvailPosInd::pXid

8.761.2.31 **ULONG** QmiCbkLocBestAvailPosInd::status

8.762 QmiCbkLocCradleMountInd Struct Reference

Data Fields

- ULONG cradleMountConfigStatus

8.762.1 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

8.762.2 Field Documentation

8.762.2.1 ULONG QmiCbkLocCradleMountInd::cradleMountConfigStatus

8.763 QmiCbkLocEngineStateInd Struct Reference

Data Fields

- ULONG engineState

8.763.1 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

8.763.2 Field Documentation

8.763.2.1 ULONG QmiCbkLocEngineStateInd::engineState

8.764 QmiCbkLocEventTimeSyncInd Struct Reference

Data Fields

- ULONG timeSyncRefCounter

8.764.1 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.

8.764.2 Field Documentation

8.764.2.1 ULONG QmiCbkLocEventTimeSyncInd::timeSyncRefCounter

8.765 QmiCbkLocGetServerInd Struct Reference

Data Fields

- ULONG serverStatus
- ULONG serverType
- IPv4Info * pIPv4AddrInfo
- IPv6Info * pIPv6AddrInfo
- urlAddr * pURL

8.765.1 Detailed Description

This structure contains LOC Get Server field.

Parameters

serverStatus	• Status of the Get Server 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

serverType	• Type of server. • Valid values – eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE – eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC – eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP – eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) - Server type is custom PDE
pIPv4AddrInfo	• See IPv4Info for more information.
pIPv6AddrInfo	• See IPv6Info for more information.
pURL	• See urlAddr for more information.

8.765.2 Field Documentation

8.765.2.1 IPv4Info∗ QmiCbkLocGetServerInd::pIPv4AddrInfo

8.765.2.2 IPv6Info∗ QmiCbkLocGetServerInd::pIPv6AddrInfo

8.765.2.3 urlAddr∗ QmiCbkLocGetServerInd::pURL

8.765.2.4 ULONG QmiCbkLocGetServerInd::serverStatus

8.765.2.5 ULONG QmiCbkLocGetServerInd::serverType

8.766 QmiCbkLocInjectPositionInd Struct Reference

Data Fields

- ULONG status

8.766.1 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

8.766.2 Field Documentation

8.766.2.1 ULONG QmiCbkLocInjectPositionInd::status

8.767 QmiCbkLocInjectSensorDataInd Struct Reference

Data Fields

- ULONG injectSensorDataStatus
- ULONG * pOpaqueIdentifier
- BYTE * pAccelSamplesAccepted
- BYTE * pGyroSamplesAccepted
- BYTE * pAccelTempSamplesAccepted
- BYTE * pGyroTempSamplesAccepted

8.767.1 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.

8.767.2 Field Documentation

8.767.2.1 ULONG QmiCbkLocInjectSensorDataInd::injectSensorDataStatus

8.767.2.2 BYTE* QmiCbkLocInjectSensorDataInd::pAccelSamplesAccepted

8.767.2.3 BYTE* QmiCbkLocInjectSensorDataInd::pAccelTempSamplesAccepted

8.767.2.4 BYTE* QmiCbkLocInjectSensorDataInd::pGyroSamplesAccepted

8.767.2.5 BYTE* QmiCbkLocInjectSensorDataInd::pGyroTempSamplesAccepted

8.767.2.6 ULONG* QmiCbkLocInjectSensorDataInd::pOpaqueIdentifier

8.768 QmiCbkLocInjectTimeInd Struct Reference

Data Fields

- ULONG injectTimeSyncStatus

8.768.1 Detailed Description

This structure contains LOC Inject Time Sync Data Status

Parameters

injectTimeSyncStatus	• 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

8.768.2 Field Documentation

8.768.2.1 ULONG QmiCbkLocInjectTimeInd::injectTimeSyncStatus

8.769 QmiCbkLocInjectUTCTimeInd Struct Reference

Data Fields

- ULONG status

8.769.1 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

8.769.2 Field Documentation

8.769.2.1 ULONG QmiCbkLocInjectUTCTimeInd::status

8.770 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

8.770.1 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

8.770.2 Field Documentation

8.770.2.1 BYTE* QmiCbkLocPositionReportInd::pAltitudeAssumed

8.770.2.2 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtEllipsoid

8.770.2.3 ULONG* QmiCbkLocPositionReportInd::pAltitudeWrtMeanSeaLevel

8.770.2.4 ULONG* QmiCbkLocPositionReportInd::pFixId

8.770.2.5 gpsTime* QmiCbkLocPositionReportInd::pGpsTime

8.770.2.6 **ULONG*** QmiCbkLocPositionReportInd::pHeading

8.770.2.7 **ULONG*** QmiCbkLocPositionReportInd::pHeadingUnc

8.770.2.8 **BYTE*** QmiCbkLocPositionReportInd::pHorConfidence

8.770.2.9 **ULONG*** QmiCbkLocPositionReportInd::pHorReliability

8.770.2.10 **ULONG*** QmiCbkLocPositionReportInd::pHorUncCircular

8.770.2.11 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseOrientAzimuth

8.770.2.12 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMajor

8.770.2.13 **ULONG*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMinor

8.770.2.14 **ULONGLONG*** QmiCbkLocPositionReportInd::pLatitude

8.770.2.15 **BYTE*** QmiCbkLocPositionReportInd::pLeapSeconds

8.770.2.16 **ULONGLONG*** QmiCbkLocPositionReportInd::pLongitude

8.770.2.17 **ULONG*** QmiCbkLocPositionReportInd::pMagneticDeviation

8.770.2.18 **precisionDilution*** QmiCbkLocPositionReportInd::pPrecisionDilution

8.770.2.19 **sensorDataUsage*** QmiCbkLocPositionReportInd::pSensorDataUsage

8.770.2.20 **ULONG*** QmiCbkLocPositionReportInd::pSpeedHorizontal

8.770.2.21 **ULONG*** QmiCbkLocPositionReportInd::pSpeedUnc

8.770.2.22 **ULONG*** QmiCbkLocPositionReportInd::pSpeedVertical

8.770.2.23 **svUsedforFix*** QmiCbkLocPositionReportInd::pSvUsedforFix

8.770.2.24 **ULONG*** QmiCbkLocPositionReportInd::pTechnologyMask

8.770.2.25 **ULONG*** QmiCbkLocPositionReportInd::pTimeSrc

8.770.2.26 **ULONGLONG*** QmiCbkLocPositionReportInd::pTimestampUtc

8.770.2.27 **ULONG*** QmiCbkLocPositionReportInd::pTimeUnc

8.770.2.28 **BYTE*** QmiCbkLocPositionReportInd::pVertConfidence

8.770.2.29 **ULONG*** QmiCbkLocPositionReportInd::pVertReliability

8.770.2.30 **ULONG*** QmiCbkLocPositionReportInd::pVertUnc

8.770.2.31 **BYTE** QmiCbkLocPositionReportInd::sessionId

8.770.2.32 **ULONG** QmiCbkLocPositionReportInd::sessionStatus

8.771 QmiCbkLocSensorStreamingInd Struct Reference

Data Fields

- accelAcceptReady * pAccelAcceptReady
- gyroAcceptReady * pGyroAcceptReady
- accelTempAcceptReady * pAccelTempAcceptReady
- gyroTempAcceptReady * pGyroTempAcceptReady

8.771.1 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

8.771.2 Field Documentation

8.771.2.1 accelAcceptReady* QmiCbkLocSensorStreamingInd::pAccelAcceptReady

8.771.2.2 accelTempAcceptReady* QmiCbkLocSensorStreamingInd::pAccelTempAcceptReady

8.771.2.3 gyroAcceptReady* QmiCbkLocSensorStreamingInd::pGyroAcceptReady

8.771.2.4 gyroTempAcceptReady* QmiCbkLocSensorStreamingInd::pGyroTempAcceptReady

8.772 QmiCbkLocSetExtPowerConfigInd Struct Reference

Data Fields

- ULONG status

8.772.1 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

8.772.2 Field Documentation

8.772.2.1 ULONG QmiCbkLocSetExtPowerConfigInd::status

8.773 QmiCbkNasLTECphyCaInfo Struct Reference

Data Fields

- PhyCaAggScellIndType sPhyCaAggScellIndType
- PhyCaAggScellDlBw sPhyCaAggScellDlBw
- PhyCaAggScellInfo sPhyCaAggScellInfo
- PhyCaAggPcellInfo sPhyCaAggPcellInfo
- PhyCaAggScellIndex sPhyCaAggScellIndex

8.773.1 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.

8.773.2 Field Documentation

8.773.2.1 PhyCaAggPcellInfo QmiCbkNasLTEphyCaInfo::sPhyCaAggPcellInfo

8.773.2.2 PhyCaAggScellDlBw QmiCbkNasLTEphyCaInfo::sPhyCaAggScellDlBw

8.773.2.3 PhyCaAggScellIndex QmiCbkNasLTEphyCaInfo::sPhyCaAggScellIndex

8.773.2.4 PhyCaAggScellIndType QmiCbkNasLTEphyCaInfo::sPhyCaAggScellIndType

8.773.2.5 PhyCaAggScellInfo QmiCbkNasLTEphyCaInfo::sPhyCaAggScellInfo

8.774 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference

Data Fields

- struct sessionInfoTlv SITlv

8.774.1 Field Documentation

8.774.1.1 struct sessionInfoTlv QmiCbkSwiOmaDmEventStatusReportInd::SITlv

8.775 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference

Data Fields

- struct sessionInfoTlvExt SITlv

8.775.1 Field Documentation

8.775.1.1 struct sessionInfoTlvExt QmiCbkSwiOmaDmEventStatusReportIndExt::SITlv

8.776 QmiCbkTmdMitiLvlRptInd Struct Reference

Data Fields

- _MitigationDevInfo MitigationDevInfo
- BYTE currentMitigationLvl

8.776.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

Parameters

MitigationDevInfo	• See MitigationDevInfo for more information.
currentMitigationLvl	• Current Thermal Mitigation Level

8.776.2 Field Documentation

8.776.2.1 BYTE QmiCbkTmdMitiLvlRptInd::currentMitigationLvl

8.776.2.2 _MitigationDevInfo QmiCbkTmdMitiLvlRptInd::MitigationDevInfo

8.777 QmiCbkWdsStatisticsIndState Struct Reference

Data Fields

- DataULongTlv TxOkConutTlv
- DataULongTlv RxOkConutTlv
- DataULongLongTlv TxOkByteCountTlv
- DataULongLongTlv RxOkByteCountTlv
- DataULongTlv TxDropConutTlv
- DataULongTlv RxDropConutTlv

8.777.1 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.

8.777.2 Field Documentation

8.777.2.1 DataULongTlv QmiCbkWdsStatisticsIndState::RxDropConutTlv

8.777.2.2 DataULongLongTlv QmiCbkWdsStatisticsIndState::RxOkByteCountTlv

8.777.2.3 DataULongTlv QmiCbkWdsStatisticsIndState::RxOkConutTlv

8.777.2.4 DataULongTlv QmiCbkWdsStatisticsIndState::TxDropConutTlv

8.777.2.5 DataULongLongTlv QmiCbkWdsStatisticsIndState::TxOkByteCountTlv

8.777.2.6 DataULongTlv QmiCbkWdsStatisticsIndState::TxOkConutTlv

8.778 qmifwinfo_s Struct Reference

Data Fields

- union {
  struct fwinfo_s g
  struct slqsfwinfo_s s
  } dev

8.778.1 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

8.778.2 Field Documentation

8.778.2.1 union { ... } qmifwinfo_s::dev

8.778.2.2 struct fwinfo_s qmifwinfo_s::g

8.778.2.3 struct slqsfwinfo_s qmifwinfo_s::s

8.779 QmiNas3GppNetworkInfo Struct Reference

Data Fields

- WORD pMCC
- WORD pMNC
- ULONG pInUse
- ULONG pRoaming
- ULONG pForbidden
- ULONG pPreferred
- CHAR pDesription [255]

8.779.1 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

8.779.2 Field Documentation

8.779.2.1 CHAR QmiNas3GppNetworkInfo::pDesription[255]

8.779.2.2 ULONG QmiNas3GppNetworkInfo::pForbidden

8.779.2.3 ULONG QmiNas3GppNetworkInfo::pInUse

8.779.2.4 WORD QmiNas3GppNetworkInfo::pMCC

8.779.2.5 WORD QmiNas3GppNetworkInfo::pMNC

8.779.2.6 ULONG QmiNas3GppNetworkInfo::pPreferred

8.779.2.7 ULONG QmiNas3GppNetworkInfo::pRoaming

8.780 QmiNasGetRFBandInfoResp Struct Reference

Data Fields

- struct qmTlvResult results
- BYTE * pInstancesSize
- struct RFBandInfoElements * pRFBandInfoElements

8.780.1 Field Documentation

8.780.1.1 BYTE* QmiNasGetRFBandInfoResp::pInstancesSize

8.780.1.2 struct RFBandInfoElements* QmiNasGetRFBandInfoResp::pRFBandInfoElements

8.780.1.3 struct qmTlvResult QmiNasGetRFBandInfoResp::results

8.781 QmiNasPerformNetworkScanResp Struct Reference

Data Fields

- struct qmTlvResult results
- BYTE * pInstanceSize
- struct QmiNas3GppNetworkInfo * pInstances

8.781.1 Field Documentation

8.781.1.1 struct QmiNas3GppNetworkInfo* QmiNasPerformNetworkScanResp::pInstances

8.781.1.2 BYTE* QmiNasPerformNetworkScanResp::pInstanceSize

8.781.1.3 struct qmTlvResult QmiNasPerformNetworkScanResp::results

8.782 qmiSmsMessageList Struct Reference

Data Fields

- uint32_t messageIndex
- uint32_t messageTag

8.782.1 Detailed Description

Parameters

messageIndex	• Message index of each matched message
messageTag	• Messagetag

8.782.2 Field Documentation

8.782.2.1 uint32_t qmiSmsMessageList::messageIndex

8.782.2.2 uint32_t qmiSmsMessageList::messageTag

8.783 qmiWDSDataBearerTechnology Struct Reference

Data Fields

- uint8_t currentNetwork
- uint32_t ratMask
- uint32_t soMask

8.783.1 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

8.783.2 Field Documentation

8.783.2.1 uint8_t qmiWDSDataBearerTechnology::currentNetwork

8.783.2.2 uint32_t qmiWDSDataBearerTechnology::ratMask

8.783.2.3 uint32_t qmiWDSDataBearerTechnology::soMask

8.784 QmiWdsIpAddressInfo Struct Reference

Data Fields

- ULONG * pIPAddressV4
- USHORT * pIPAddressV6
- BYTE * pIPv6prefixlen

8.784.1 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

8.784.2 Field Documentation

8.784.2.1 ULONG* QmiWdsIpAddressInfo::pIPAddressV4

8.784.2.2 USHORT* QmiWdsIpAddressInfo::pIPAddressV6

8.784.2.3 BYTE* QmiWdsIpAddressInfo::pIPv6prefixlen

8.785 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

8.785.1 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
pMtu	• MTU
pDomainList	• Domain-Name List
pIPFamilyPreference	• IP family – 0x04 - IPV4 ADDR – 0x06 - IPV6 ADDR
pIMCNflag	• IM CN Flag – 0x00 - FALSE – 0x01 - TRUE
pTechnology	• Technology – CDMA - 0x8001 – UMTS - 0x8004
pIPV6AddressInfo	• IPV6 Address Information
pIPV6GWAddressInfo	• IPV6 Gateway Address Information

8.785.2 Field Documentation

8.785.2.1 CHAR* qmiWdsRunTimeSettings::pAPNName

8.785.2.2 ULONG* qmiWdsRunTimeSettings::pAuthentication

8.785.2.3 struct DomainNameList* qmiWdsRunTimeSettings::pDomainList

8.785.2.4 struct GPRSQoS* qmiWdsRunTimeSettings::pGPRSGrantedQoS

8.785.2.5 ULONG* qmiWdsRunTimeSettings::pGWAddressV4

8.785.2.6 BYTE* qmiWdsRunTimeSettings::pIMCNflag

8.785.2.7 ULONG* qmiWdsRunTimeSettings::pIPAddressV4

8.785.2.8 BYTE* qmiWdsRunTimeSettings::pIPFamilyPreference

8.785.2.9 struct IPV6AddressInfo* qmiWdsRunTimeSettings::pIPV6AddrInfo

8.785.2.10 struct IPV6GWAddressInfo* qmiWdsRunTimeSettings::pIPV6GWAddrInfo

8.785.2.11 ULONG* qmiWdsRunTimeSettings::pMtu

8.785.2.12 BYTE* qmiWdsRunTimeSettings::pPCSCFAddrPCO

8.785.2.13 struct PCSCFFQDNAddressList* qmiWdsRunTimeSettings::pPCSCFFQDNAddrList

8.785.2.14 ULONG* qmiWdsRunTimeSettings::pPDPType

8.785.2.15 ULONG* qmiWdsRunTimeSettings::pPrimaryDNSV4

8.785.2.16 USHORT* qmiWdsRunTimeSettings::pPrimaryDNSV6

8.785.2.17 struct ProfileIdentifier* qmiWdsRunTimeSettings::pProfileID

8.785.2.18 CHAR* qmiWdsRunTimeSettings::pProfileName

8.785.2.19 ULONG* qmiWdsRunTimeSettings::pSecondaryDNSV4

8.785.2.20 USHORT* qmiWdsRunTimeSettings::pSecondaryDNSV6

8.785.2.21 struct PCSCFIPv4ServerAddressList* qmiWdsRunTimeSettings::pServerAddrList

8.785.2.22 ULONG* qmiWdsRunTimeSettings::pSubnetMaskV4

8.785.2.23 WORD* qmiWdsRunTimeSettings::pTechnology

8.785.2.24 struct UMTSQoS* qmiWdsRunTimeSettings::pUMTSGrantedQoS

8.785.2.25 CHAR* qmiWdsRunTimeSettings::pUsername

8.786 QosClassID Struct Reference

Data Fields

- BYTE QCI
- ULONG gDlBitRate
- ULONG maxDlBitRate
- ULONG gUlBitRate

- ULONG maxUlBitRate

8.786.1 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

8.786.2 Field Documentation

8.786.2.1 ULONG QosClassID::gDlBitRate

8.786.2.2 ULONG QosClassID::gUlBitRate

8.786.2.3 ULONG QosClassID::maxDlBitRate

8.786.2.4 ULONG QosClassID::maxUlBitRate

8.786.2.5 BYTE QosClassID::QCI

8.787 QosEventInfo Struct Reference

Data Fields

- ULONG * pDataBearer
- ULONG * pPacketsCountTX
- ULONG * pPacketsCountRX
- ULONGLONG * pTotalBytesTX
- ULONGLONG * pTotalBytesRX

8.787.1 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)

8.787.2 Field Documentation

8.787.2.1 ULONG* QosEventInfo::pDataBearer

8.787.2.2 ULONG* QosEventInfo::pPacketsCountRX

8.787.2.3 ULONG* QosEventInfo::pPacketsCountTX

8.787.2.4 ULONGLONG* QosEventInfo::pTotalBytesRX

8.787.2.5 ULONGLONG* QosEventInfo::pTotalBytesTX

8.788 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

8.788.1 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.

8.788.2 Field Documentation

8.788.2.1 BYTE* QosFlowInfo::pBearerID

8.788.2.2 QosFlowInfoState* QosFlowInfo::pQFlowState

8.788.2.3 swiQosFilter* QosFlowInfo::pRxQFilter[MAX_QOS_FILTER_TLV]

8.788.2.4 swiQosFlow* QosFlowInfo::pRxQFlowGranted

8.788.2.5 swiQosFilter* QosFlowInfo::pTxQFilter[MAX_QOS_FILTER_TLV]

8.788.2.6 swiQosFlow* QosFlowInfo::pTxQFlowGranted

8.789 QosFlowInfoState Struct Reference

Data Fields

- ULONG id
- BYTE isNewFlow
- BYTE state

8.789.1 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

8.789.2 Field Documentation

8.789.2.1 ULONG QosFlowInfoState::id

8.789.2.2 BYTE QosFlowInfoState::isNewFlow

8.789.2.3 BYTE QosFlowInfoState::state

8.790 QosMap Struct Reference

Data Fields

- BYTE dscp
- ULONG qos_id
- BYTE state

8.790.1 Detailed Description

This structure contains the SLQSQosDumpMap Information

Parameters

dscp	• Differential Service Code Point(DSCP) value
qos_id	• QoS identifier
state	• QoS Flow state

8.790.2 Field Documentation

8.790.2.1 BYTE QosMap::dscp

8.790.2.2 ULONG QosMap::qos_id

8.790.2.3 BYTE QosMap::state

8.791 RankIndicatorInd Struct Reference

Data Fields

- WORD Count1
- WORD Count2

8.791.1 Field Documentation

8.791.1.1 WORD RankIndicatorInd::Count1

8.791.1.2 WORD RankIndicatorInd::Count2

8.792 readResult Struct Reference

Data Fields

- WORD contentLen
- BYTE content [255+1]

8.792.1 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.

8.792.2 Field Documentation

8.792.2.1 BYTE readResult::content[255+1]

8.792.2.2 WORD readResult::contentLen

8.793 readTransparentInfo Struct Reference

Data Fields

- WORD offset
- WORD length

8.793.1 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.

8.793.2 Field Documentation

8.793.2.1 WORD readTransparentInfo::length

8.793.2.2 WORD readTransparentInfo::offset

8.794 redirNumInfo Struct Reference

Data Fields

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

8.794.1 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.

8.794.2 Field Documentation

8.794.2.1 BYTE redirNumInfo::number[255]

8.794.2.2 BYTE redirNumInfo::numLen

8.794.2.3 BYTE redirNumInfo::numPlan

8.794.2.4 BYTE redirNumInfo::numType

8.794.2.5 BYTE redirNumInfo::PI

8.794.2.6 BYTE redirNumInfo::reason

8.794.2.7 BYTE redirNumInfo::SI

8.795 registerRefresh Struct Reference

Data Fields

- BYTE registerFlag
- BYTE voteForInit
- WORD numFiles
- fileInfo arrfileInfo [255]

8.795.1 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

8.795.2 Field Documentation

8.795.2.1 fileInfo registerRefresh::arrfileInfo[255]

8.795.2.2 WORD registerRefresh::numFiles

8.795.2.3 BYTE registerRefresh::registerFlag

8.795.2.4 BYTE registerRefresh::voteForInit

8.796 remainingRetries Struct Reference

Data Fields

- BYTE verifyLeft
- BYTE unblockLeft

8.796.1 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.

8.796.2 Field Documentation

8.796.2.1 BYTE remainingRetries::unblockLeft

8.796.2.2 BYTE remainingRetries::verifyLeft

8.797 remotePartyName Struct Reference

Data Fields

- BYTE namePI
- BYTE codingScheme
- BYTE nameLen
- BYTE callerName [255]

8.797.1 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.

8.797.2 Field Documentation

8.797.2.1 BYTE remotePartyName::callerName[255]

8.797.2.2 BYTE remotePartyName::codingScheme

8.797.2.3 BYTE remotePartyName::nameLen

8.797.2.4 BYTE remotePartyName::namePI

8.798 remotePartyNum Struct Reference

Data Fields

- BYTE presentationInd
- BYTE numLen
- BYTE remPartyNumber [81]

8.798.1 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.

8.798.2 Field Documentation

8.798.2.1 BYTE remotePartyNum::numLen

8.798.2.2 BYTE remotePartyNum::presentationInd

8.798.2.3 BYTE remotePartyNum::remPartyNumber[81]

8.799 ReqFieldsList Struct Reference

Data Fields

- BYTE requestFieldsLen
- BYTE requestFields [256]

8.799.1 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.

8.799.2 Field Documentation

8.799.2.1 BYTE ReqFieldsList::requestFields[256]

8.799.2.2 BYTE ReqFieldsList::requestFieldsLen

8.800 RespFieldsList Struct Reference

Data Fields

- BYTE responseFieldsLen
- BYTE responseFields [256]

8.800.1 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.

8.800.2 Field Documentation

8.800.2.1 BYTE RespFieldsList::responseFields[256]

8.800.2.2 BYTE RespFieldsList::responseFieldsLen

8.801 RFBandInfoElements Struct Reference

Data Fields

- BYTE radioInterface
- WORD activeBandClass
- WORD activeChannel
- uint8_t radioInterface
- uint16_t activeBandClass
- uint16_t activeChannel

8.801.1 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)

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)

8.801.2 Field Documentation

8.801.2.1 WORD RFBandInfoElements::activeBandClass

8.801.2.2 uint16_t RFBandInfoElements::activeBandClass

8.801.2.3 WORD RFBandInfoElements::activeChannel

8.801.2.4 uint16_t RFBandInfoElements::activeChannel

8.801.2.5 BYTE RFBandInfoElements::radioInterface

8.801.2.6 uint8_t RFBandInfoElements::radioInterface

8.802 rmTrasnferStaticsReq Struct Reference

Data Fields

- uint8_t bResetStatistics
- uint32_t ulMask

8.802.1 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

8.802.2 Field Documentation

8.802.2.1 uint8_t rmTrasnferStaticsReq::bResetStatistics

8.802.2.2 uint32_t rmTrasnferStaticsReq::ulMask

8.803 roamIndList Struct Reference

Data Fields

- BYTE numInstances
- BYTE radioInterface [0x0A]
- BYTE roamIndicator [0x0A]

8.803.1 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

8.803.2 Field Documentation

8.803.2.1 BYTE roamIndList::numInstances

8.803.2.2 BYTE roamIndList::radioInterface[0x0A]

8.803.2.3 BYTE roamIndList::roamIndicator[0x0A]

8.804 RoamingInfo Struct Reference

Data Fields

- BYTE TlvPresent
- BYTE roaming_ind

8.804.1 Field Documentation

8.804.1.1 BYTE RoamingInfo::roaming_ind

8.804.1.2 BYTE RoamingInfo::TlvPresent

8.805 roamTimer Struct Reference

Data Fields

- BYTE namID
- ULONG roamTimerValue

8.805.1 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.

8.805.2 Field Documentation

8.805.2.1 BYTE roamTimer::namID

8.805.2.2 ULONG roamTimer::roamTimerValue

8.806 RSRPThresh Struct Reference

Data Fields

- BYTE RSRPThresListLen
- SHORT * pRSRPThresList

8.806.1 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

8.806.2 Field Documentation

8.806.2.1 SHORT* RSRPThresh::pRSRPThresList

8.806.2.2 BYTE RSRPThresh::RSRPThresListLen

8.807 rsrqInformation Struct Reference

Data Fields

- INT8 rsrq
- BYTE radioIf

8.807.1 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

8.807.2 Field Documentation

8.807.2.1 BYTE rsrqInformation::radioIf

8.807.2.2 INT8 rsrqInformation::rsrq

8.808 RSRQThresh Struct Reference

Data Fields

- BYTE RSRQThresListLen
- INT8 * pRSRQThresList

8.808.1 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

8.808.2 Field Documentation

8.808.2.1 INT8* RSRQThresh::pRSRQThresList

8.808.2.2 BYTE RSRQThresh::RSRQThresListLen

8.809 RSSIThresh Struct Reference

Data Fields

- BYTE RSSIThresListLen
- INT8 * pRSSIThresList

8.809.1 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.

8.809.2 Field Documentation

8.809.2.1 INT8* RSSIThresh::pRSSIThresList

8.809.2.2 BYTE RSSIThresh::RSSIThresListLen

8.810 RXAGCList Struct Reference

Data Fields

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

8.810.1 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

8.810.2 Field Documentation

8.810.2.1 WORD* RXAGCList::pRXAIG

8.810.2.2 WORD* RXAGCList::pRXComprSlope

8.810.2.3 WORD* RXAGCList::pRXComprThres

8.810.2.4 WORD* RXAGCList::pRXExpSlope

8.810.2.5 WORD* RXAGCList::pRXExpThres

8.810.2.6 WORD* RXAGCList::pRXStaticGain

8.811 RXAVCList Struct Reference

Data Fields

- WORD * pAVRXAVCSens
- WORD * pAVRXAVCHeadroom

8.811.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXAVCLIST.

Parameters

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

8.811.2 Field Documentation

8.811.2.1 WORD* RXAVCList::pAVRXAVCHeadroom

8.811.2.2 WORD* RXAVCList::pAVRXAVCSens

8.812 rxInfo Struct Reference

Data Fields

- BYTE isRadioTuned
- INT32 rxPower
- INT32 ecio
- INT32 rscp
- INT32 rsrp
- ULONG phase

8.812.1 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.

8.812.2 Field Documentation

8.812.2.1 INT32 rxInfo::ecio

8.812.2.2 BYTE rxInfo::isRadioTuned

8.812.2.3 ULONG rxInfo::phase

8.812.2.4 INT32 rxInfo::rscp

8.812.2.5 INT32 rxInfo::rsrp

8.812.2.6 INT32 rxInfo::rxPower

8.813 RXPCMIIRFltr Struct Reference

Data Fields

- WORD * pFlag
- WORD * pStageCnt
- BYTE * pStage0Val
- BYTE * pStage1Val
- BYTE * pStage2Val
- BYTE * pStage3Val
- BYTE * pStage4Val

8.813.1 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

8.813.2 Field Documentation

8.813.2.1 WORD* RXPCMIIRFltr::pFlag

8.813.2.2 BYTE* RXPCMIIRFltr::pStage0Val

8.813.2.3 BYTE* RXPCMIIRFltr::pStage1Val

8.813.2.4 BYTE* RXPCMIIRFltr::pStage2Val

8.813.2.5 BYTE* RXPCMIIRFltr::pStage3Val

8.813.2.6 BYTE* RXPCMIIRFltr::pStage4Val

8.813.2.7 WORD* RXPCMIIRFltr::pStageCnt

8.814 RxSigInfo Struct Reference

Data Fields

- BYTE rxChainIndex
- BYTE isRadioTuned
- INT32 rxPower
- INT32 rsrp

8.814.1 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

8.814.2 Field Documentation

8.814.2.1 BYTE RxSigInfo::isRadioTuned

8.814.2.2 INT32 RxSigInfo::rsrp

8.814.2.3 BYTE RxSigInfo::rxChainIndex

8.814.2.4 INT32 RxSigInfo::rxPower

8.815 rxSignalStrengthListElement Struct Reference

Data Fields

- SHORT rxSignalStrength
- BYTE radioIf

8.815.1 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.

8.815.2 Field Documentation

8.815.2.1 BYTE rxSignalStrengthListElement::radioIf

8.815.2.2 SHORT rxSignalStrengthListElement::rxSignalStrength

8.816 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

8.816.1 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

8.816.2 Field Documentation

8.816.2.1 BYTE sApnExtraParams::ambr_dl

8.816.2.2 BYTE sApnExtraParams::ambr_dl_ext

8.816.2.3 BYTE sApnExtraParams::ambr_dl_ext2

8.816.2.4 BYTE sApnExtraParams::ambr_ul

8.816.2.5 BYTE sApnExtraParams::ambr_ul_ext

8.816.2.6 BYTE sApnExtraParams::ambr_ul_ext2

8.816.2.7 ULONG sApnExtraParams::apnId

8.817 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

8.817.1 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

8.817.2 Field Documentation

8.817.2.1 FLOAT satelliteInfo::azimuth

8.817.2.2 FLOAT satelliteInfo::elevation

8.817.2.3 WORD satelliteInfo::gnssSvId

8.817.2.4 BYTE satelliteInfo::healthStatus

8.817.2.5 FLOAT satelliteInfo::snr

8.817.2.6 BYTE satelliteInfo::svInfoMask

8.817.2.7 BYTE satelliteInfo::svListLen

8.817.2.8 ULONG satelliteInfo::svStatus

8.817.2.9 ULONG satelliteInfo::system

8.817.2.10 ULONG satelliteInfo::validMask

8.818 SccRxInfo Struct Reference

Data Fields

- INT32 rsrq
- SHORT snr
- BYTE numInstances
- RxSigInfo sigInfo [255]
- BYTE TlvPresent

8.818.1 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.

8.818.2 Field Documentation

8.818.2.1 BYTE SccRxInfo::numInstances

8.818.2.2 INT32 SccRxInfo::rsrq

8.818.2.3 RxSigInfo SccRxInfo::sigInfo[255]

8.818.2.4 SHORT SccRxInfo::snr

8.818.2.5 BYTE SccRxInfo::TlvPresent

8.819 sensorData Struct Reference

Data Fields

- ULONG timeOfFirstSample
- BYTE flags
- BYTE sensorDataLen
- WORD timeOffset [64]
- ULONG xAxis [64]
- ULONG yAxis [64]
- ULONG zAxis [64]

8.819.1 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

8.819.2 Field Documentation

8.819.2.1 BYTE sensorData::flags

8.819.2.2 BYTE sensorData::sensorDataLen

8.819.2.3 ULONG sensorData::timeOfFirstSample

8.819.2.4 WORD sensorData::timeOffset[64]

8.819.2.5 ULONG sensorData::xAxis[64]

8.819.2.6 ULONG sensorData::yAxis[64]

8.819.2.7 ULONG sensorData::zAxis[64]

8.820 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]

8.820.1 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

8.820.2 Field Documentation

8.820.2.1 uint8_t sensorData_t::flags

8.820.2.2 uint8_t sensorData_t::sensorDataLen

8.820.2.3 uint32_t sensorData_t::timeOfFirstSample

8.820.2.4 uint16_t sensorData_t::timeOffset[64]

8.820.2.5 uint32_t sensorData_t::xAxis[64]

8.820.2.6 uint32_t sensorData_t::yAxis[64]

8.820.2.7 uint32_t sensorData_t::zAxis[64]

8.821 sensorDataUsage_s Struct Reference

Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

8.821.1 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

8.821.2 Field Documentation

8.821.2.1 ULONG sensorDataUsage_s::aidingIndicatorMask

8.821.2.2 ULONG sensorDataUsage_s::usageMask

8.822 serialNumbersInfo Struct Reference

Data Fields

- BYTE esnSize
- CHAR * pESNString
- BYTE imeiSize
- CHAR * pIMEIString
- BYTE meidSize
- CHAR * pMEIDString
- BYTE imeiSvnSize
- CHAR * pImeiSvnString

8.822.1 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.

8.822.2 Field Documentation

8.822.2.1 BYTE serialNumbersInfo::esnSize

8.822.2.2 BYTE serialNumbersInfo::imeiSize

8.822.2.3 BYTE serialNumbersInfo::imeiSvnSize

8.822.2.4 BYTE serialNumbersInfo::meidSize

8.822.2.5 CHAR* serialNumbersInfo::pESNString

8.822.2.6 CHAR* serialNumbersInfo::pIMEIString

8.822.2.7 CHAR* serialNumbersInfo::pImeiSvnString

8.822.2.8 CHAR* serialNumbersInfo::pMEIDString

8.823 serviceProviderName Struct Reference

Data Fields

- BYTE displayCondition
- BYTE spnLength
- BYTE spn [255]

8.823.1 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.

8.823.2 Field Documentation

8.823.2.1 BYTE serviceProviderName::displayCondition

8.823.2.2 BYTE serviceProviderName::spn[255]

8.823.2.3 BYTE serviceProviderName::spnLength

8.824 ServingSystemInfo Struct Reference

Data Fields

- BYTE registrationState
- BYTE csAttachState
- BYTE psAttachState
- BYTE selectedNetwork
- BYTE radioInterfaceNo
- BYTE radioInterfaceList [255]
- BYTE hdrPersonality

8.824.1 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

8.824.2 Field Documentation

8.824.2.1 BYTE ServingSystemInfo::csAttachState

8.824.2.2 BYTE ServingSystemInfo::hdrPersonality

8.824.2.3 BYTE ServingSystemInfo::psAttachState

8.824.2.4 BYTE ServingSystemInfo::radioInterfaceList[255]

8.824.2.5 BYTE ServingSystemInfo::radioInterfaceNo

8.824.2.6 BYTE ServingSystemInfo::registrationState

8.824.2.7 BYTE ServingSystemInfo::selectedNetwork

8.825 servSystem Struct Reference

Data Fields

- BYTE regState
- BYTE csAttachState
- BYTE psAttachState
- BYTE selNetwork
- BYTE numRadioInterfaces
- BYTE radioInterface [0x0A]

8.825.1 Detailed Description

This structure contains the Serving System parameters

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

Parameters

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

8.825.2 Field Documentation

8.825.2.1 BYTE servSystem::csAttachState

8.825.2.2 BYTE servSystem::numRadioInterfaces

8.825.2.3 BYTE servSystem::psAttachState

8.825.2.4 BYTE servSystem::radioInterface[0x0A]

8.825.2.5 BYTE servSystem::regState

8.825.2.6 BYTE servSystem::selNetwork

8.826 sessionInfo Union Reference

Data Fields

- struct omaDmFotaTlv omaDmFota
- struct omaDmConfigTlv omaDmConfig
- struct omaDmNotificationsTlv omaDmNotifications

8.826.1 Detailed Description

This union sessionInfo consist of omaDmFotaTlv, omaDmConfigTlv and omaDmNotificationsTlv, out of which one will be unpacked against pEventFields.

8.826.2 Field Documentation

8.826.2.1 struct omaDmConfigTlv sessionInfo::omaDmConfig

8.826.2.2 struct omaDmFotaTlv sessionInfo::omaDmFota

8.826.2.3 struct omaDmNotificationsTlv sessionInfo::omaDmNotifications

8.827 sessionInfoExt Union Reference

Data Fields

- struct omaDmFotaTlvExt omaDmFota
- struct omaDmConfigTlvExt omaDmConfig

8.827.1 Detailed Description

This union sessionInfo consist of omaDmFotaTlv and omaDmConfigTlv, out of which one will be unpacked against pEventFields.

8.827.2 Field Documentation

8.827.2.1 struct omaDmConfigTlvExt sessionInfoExt::omaDmConfig

8.827.2.2 struct omaDmFotaTlvExt sessionInfoExt::omaDmFota

8.828 sessionInfoTlv Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformation sessionInfo

8.828.1 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

8.828.2 Field Documentation

8.828.2.1 sessionInformation sessionInfoTlv::sessionInfo

8.828.2.2 ULONG sessionInfoTlv::sessionType

8.828.2.3 BYTE sessionInfoTlv::TlvPresent

8.829 sessionInfoTlvExt Struct Reference

Data Fields

- BYTE TlvPresent
- ULONG sessionType
- sessionInformationExt sessionInfo

8.829.1 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

8.829.2 Field Documentation

8.829.2.1 sessionInformationExt sessionInfoTlvExt::sessionInfo

8.829.2.2 ULONG sessionInfoTlvExt::sessionType

8.829.2.3 BYTE sessionInfoTlvExt::TlvPresent

8.830 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

8.830.1 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

8.830.2 Field Documentation

8.830.2.1 WORD* SetAudioPathConfigReq::pCodecSTGain

8.830.2.2 WORD* SetAudioPathConfigReq::pDTMFTXGain

8.830.2.3 BYTE* SetAudioPathConfigReq::pECMode

8.830.2.4 BYTE* SetAudioPathConfigReq::pNSEnable

8.830.2.5 BYTE SetAudioPathConfigReq::Profile

8.830.2.6 RXAGCList* SetAudioPathConfigReq::pRXAGCList

8.830.2.7 BYTE* SetAudioPathConfigReq::pRXAVCAGCSwitch

8.830.2.8 RXAVCList* SetAudioPathConfigReq::pRXAVCList

8.830.2.9 RXPCMIIRFltr* SetAudioPathConfigReq::pRXPCMIIRFltr

8.830.2.10 TXAGCList* SetAudioPathConfigReq::pTXAGCList

8.830.2.11 BYTE* SetAudioPathConfigReq::pTXAVCSwitch

8.830.2.12 WORD* SetAudioPathConfigReq::pTXGain

8.830.2.13 TXPCMIIRFltr* SetAudioPathConfigReq::pTXPCMIIRFltr

8.831 SetAudioProfileReq Struct Reference

Data Fields

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

8.831.1 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

8.831.2 Field Documentation

8.831.2.1 BYTE SetAudioProfileReq::EarMute

8.831.2.2 BYTE SetAudioProfileReq::Generator

8.831.2.3 BYTE SetAudioProfileReq::MicMute

8.831.2.4 BYTE SetAudioProfileReq::Profile

8.831.2.5 BYTE SetAudioProfileReq::Volume

8.832 SetAudioVolTLBConfigReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Volume
- BYTE Item
- WORD VolValue

8.832.1 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

8.832.2 Field Documentation

8.832.2.1 BYTE SetAudioVolTLBConfigReq::Generator

8.832.2.2 BYTE SetAudioVolTLBConfigReq::Item

8.832.2.3 BYTE SetAudioVolTLBConfigReq::Profile

8.832.2.4 BYTE SetAudioVolTLBConfigReq::Volume

8.832.2.5 WORD SetAudioVolTLBConfigReq::VolValue

8.833 SetAudioVolTLBConfigResp Struct Reference

Data Fields

- WORD ResCode

8.833.1 Detailed Description

This structure contains the SLQSSetAudioVolTLBConfig response parameters.

Parameters

ResCode	• Result of requested item

8.833.2 Field Documentation

8.833.2.1 WORD SetAudioVolTLBConfigResp::ResCode

8.834 setCustomSettingV2 Struct Reference

Data Fields

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

8.834.1 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)

8.834.2 Field Documentation

8.834.2.1 CHAR setCustomSettingV2::cust_id[64+1]

8.834.2.2 BYTE setCustomSettingV2::cust_value[8+1]

8.834.2.3 WORD setCustomSettingV2::value_length

8.835 setDyingGaspCfg Struct Reference

Data Fields

- BYTE * pDestSMSNum
- BYTE * pDestSMSContent

8.835.1 Detailed Description

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

Parameters

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

8.835.2 Field Documentation

8.835.2.1 BYTE* setDyingGaspCfg::pDestSMSContent

8.835.2.2 BYTE* setDyingGaspCfg::pDestSMSNum

8.836 SetIMSSMSConfigReq Struct Reference

Data Fields

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

8.836.1 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

8.836.2 Field Documentation

8.836.2.1 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURI

8.836.2.2 BYTE* SetIMSSMSConfigReq::pPhoneCtxtURILen

8.836.2.3 BYTE* SetIMSSMSConfigReq::pSMSFormat

8.836.2.4 BYTE* SetIMSSMSConfigReq::pSMSOverIPNwInd

8.837 SetIMSSMSConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.837.1 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

8.837.2 Field Documentation

8.837.2.1 BYTE* SetIMSSMSConfigResp::pSettingResp

8.838 SetIMSUserConfigReq Struct Reference

Data Fields

- BYTE * pIMSDomainLen
- BYTE * pIMSDomain

8.838.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig request parameters.

Parameters

pIMSDomainLen	• Length of IMS Domain Name to follow
pIMSDomain	• IMS domain name

8.838.2 Field Documentation

8.838.2.1 BYTE* SetIMSUserConfigReq::pIMSDomain

8.838.2.2 BYTE* SetIMSUserConfigReq::pIMSDomainLen

8.839 SetIMSUserConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.839.1 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

8.839.2 Field Documentation

8.839.2.1 BYTE* SetIMSUserConfigResp::pSettingResp

8.840 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

8.840.1 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.

8.840.2 Field Documentation

8.840.2.1 BYTE* SetIMSVoIPConfigReq::pAmrMode

8.840.2.2 BYTE* SetIMSVoIPConfigReq::pAmrOctetAligned

8.840.2.3 BYTE* SetIMSVoIPConfigReq::pAmrWbEnable

8.840.2.4 WORD* SetIMSVoIPConfigReq::pAmrWBMode

8.840.2.5 BYTE* SetIMSVoIPConfigReq::pAmrWBOctetAligned

8.840.2.6 WORD* SetIMSVoIPConfigReq::pMinSessionExpiryTimer

8.840.2.7 WORD* SetIMSVoIPConfigReq::pRingBackTimer

8.840.2.8 WORD* SetIMSVoIPConfigReq::pRingingTimer

8.840.2.9 WORD* SetIMSVoIPConfigReq::pRTPRTCPInactTimer

8.840.2.10 BYTE* SetIMSVoIPConfigReq::pScrAmrEnable

8.840.2.11 BYTE* SetIMSVoIPConfigReq::pScrAmrWbEnable

8.840.2.12 WORD* SetIMSVoIPConfigReq::pSessionExpiryTimer

8.841 SetIMSVoIPConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.841.1 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

8.841.2 Field Documentation

8.841.2.1 BYTE* SetIMSVoIPConfigResp::pSettingResp

8.842 SetM2MAudioAVCFGReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Device
- BYTE PIFACEId
- PCMparams * pPCMParams

8.842.1 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

8.842.2 Field Documentation

8.842.2.1 BYTE SetM2MAudioAVCFGReq::Device

8.842.2.2 BYTE SetM2MAudioAVCFGReq::PIFACEId

8.842.2.3 PCMparams* SetM2MAudioAVCFGReq::pPCMParams

8.842.2.4 BYTE SetM2MAudioAVCFGReq::Profile

8.843 SetM2MAudioLPBKReq Struct Reference

Data Fields

- BYTE Enable

8.843.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK request parameters.

Parameters

Enable	• Operation to be performed – 0 - stop – 1 - VOCODER loop – 2 - internal codec loop

8.843.2 Field Documentation

8.843.2.1 BYTE SetM2MAudioLPBKReq::Enable

8.844 SetM2MAudioProfileReq Struct Reference

Data Fields

- BYTE Profile
- BYTE * pEarMute
- BYTE * pMicMute
- BYTE * pGenerator
- BYTE * pVolume
- BYTE * pCwtMute

8.844.1 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

8.844.2 Field Documentation

8.844.2.1 BYTE* SetM2MAudioProfileReq::pCwtMute

8.844.2.2 BYTE* SetM2MAudioProfileReq::pEarMute

8.844.2.3 BYTE* SetM2MAudioProfileReq::pGenerator

8.844.2.4 BYTE* SetM2MAudioProfileReq::pMicMute

8.844.2.5 BYTE SetM2MAudioProfileReq::Profile

8.844.2.6 BYTE* SetM2MAudioProfileReq::pVolume

8.845 SetM2MAudioVolumeReq Struct Reference

Data Fields

- BYTE Profile
- BYTE Generator
- BYTE Level

8.845.1 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

8.845.2 Field Documentation

8.845.2.1 BYTE SetM2MAudioVolumeReq::Generator

8.845.2.2 BYTE SetM2MAudioVolumeReq::Level

8.845.2.3 BYTE SetM2MAudioVolumeReq::Profile

8.846 SetM2MAVMuteReq Struct Reference

Data Fields

- BYTE Profile
- BYTE EarMute
- BYTE MicMute
- BYTE * pCwtMute

8.846.1 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

8.846.2 Field Documentation

8.846.2.1 BYTE SetM2MAVMuteReq::EarMute

8.846.2.2 BYTE SetM2MAVMuteReq::MicMute

8.846.2.3 BYTE* SetM2MAVMuteReq::pCwtMute

8.846.2.4 BYTE SetM2MAVMuteReq::Profile

8.847 SetM2MSpkrGainReq Struct Reference

Data Fields

- BYTE Profile
- WORD Value

8.847.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain request parameters.

Parameters

Profile	• Audio Profile Number – 0-5
Value	• RX speakerphone gain – 0x0 - 0x7fff

8.847.2 Field Documentation

8.847.2.1 BYTE SetM2MSpkrGainReq::Profile

8.847.2.2 WORD SetM2MSpkrGainReq::Value

8.848 setPINProtection Struct Reference

Data Fields

- BYTE pinID
- BYTE pinOperation
- BYTE pinLength
- BYTE pinValue [255]

8.848.1 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.

8.848.2 Field Documentation

8.848.2.1 BYTE setPINProtection::pinID

8.848.2.2 BYTE setPINProtection::pinLength

8.848.2.3 BYTE setPINProtection::pinOperation

8.848.2.4 BYTE setPINProtection::pinValue[255]

8.849 SetRegMgrConfigReq Struct Reference

Data Fields

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

8.849.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig request parameters.

Parameters

pPriCSCFPort	• Primary call session control function port
pCSCFPortNameLen	• 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

8.849.2 Field Documentation

8.849.2.1 BYTE* SetRegMgrConfigReq::pCSCFPortName

8.849.2.2 BYTE* SetRegMgrConfigReq::pCSCFPortNameLen

8.849.2.3 BYTE* SetRegMgrConfigReq::pIMSTestMode

8.849.2.4 WORD* SetRegMgrConfigReq::pPriCSCFPort

8.850 SetRegMgrConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.850.1 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

8.850.2 Field Documentation

8.850.2.1 BYTE* SetRegMgrConfigResp::pSettingResp

8.851 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

8.851.1 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.

8.851.2 Field Documentation

8.851.2.1 WORD* setSignalStrengthInfo::pCDMAECIODelta

8.851.2.2 CDMAECIOThresh* setSignalStrengthInfo::pCDMAECIOThresh

8.851.2.3 WORD* setSignalStrengthInfo::pCDMARSSIDelta

8.851.2.4 CDMARSSIThresh* setSignalStrengthInfo::pCDMARSSIThresh

8.851.2.5 WORD* setSignalStrengthInfo::pGSMRSSIDelta

8.851.2.6 GSMRSSIThresh* setSignalStrengthInfo::pGSMRSSIThresh

8.851.2.7 WORD* setSignalStrengthInfo::pHDRECIODelta

8.851.2.8 HDRECIOThresh* setSignalStrengthInfo::pHDRECIOThresh

8.851.2.9 WORD* setSignalStrengthInfo::pHDRIODelta

8.851.2.10 HDRIOThresh* setSignalStrengthInfo::pHDRIOThresh

8.851.2.11 WORD* setSignalStrengthInfo::pHDRRSSIDelta

8.851.2.12 HDRRSSIThresh* setSignalStrengthInfo::pHDRRSSIThresh

8.851.2.13 WORD* setSignalStrengthInfo::pHDRSINRDelta

8.851.2.14 HDRSINRThreshold* setSignalStrengthInfo::pHDRSINRThresh

8.851.2.15 WORD* setSignalStrengthInfo::pLTERSRPDelta

8.851.2.16 LTERSRPThresh* setSignalStrengthInfo::pLTERSRPThresh

8.851.2.17 WORD* setSignalStrengthInfo::pLTERSRQDelta

8.851.2.18 **LTERSRQThresh**∗ setSignalStrengthInfo::pLTERSRQThresh

8.851.2.19 **WORD**∗ setSignalStrengthInfo::pLTERSSIDelta

8.851.2.20 **LTERSSIThresh**∗ setSignalStrengthInfo::pLTERSSIThresh

8.851.2.21 **LTESigRptConfig**∗ setSignalStrengthInfo::pLTESigRptConfig

8.851.2.22 **WORD**∗ setSignalStrengthInfo::pLTESNRDelta

8.851.2.23 **LTESNRThreshold**∗ setSignalStrengthInfo::pLTESNRThresh

8.851.2.24 **ULONG**∗ setSignalStrengthInfo::pTDSCDMAECIODelta

8.851.2.25 **TDSCDMAECIOThresh**∗ setSignalStrengthInfo::pTDSCDMAECIOThresh

8.851.2.26 **WORD**∗ setSignalStrengthInfo::pTDSCDMARSCPDelta

8.851.2.27 **TDSCDMARSCPThresh**∗ setSignalStrengthInfo::pTDSCDMARSCPThresh

8.851.2.28 **ULONG**∗ setSignalStrengthInfo::pTDSCDMARSSIDelta

8.851.2.29 **TDSCDMARSSIThresh**∗ setSignalStrengthInfo::pTDSCDMARSSIThresh

8.851.2.30 **ULONG**∗ setSignalStrengthInfo::pTDSCDMASINRDelta

8.851.2.31 **TDSCDMASINRThresh**∗ setSignalStrengthInfo::pTDSCDMASINRThresh

8.851.2.32 **WORD**∗ setSignalStrengthInfo::pWCDMAECIODelta

8.851.2.33 **WCDMAECIOThresh**∗ setSignalStrengthInfo::pWCDMAECIOThresh

8.851.2.34 **WORD**∗ setSignalStrengthInfo::pWCDMARSSIDelta

8.851.2.35 **WCDMARSSIThresh**∗ setSignalStrengthInfo::pWCDMARSSIThresh

8.852 SetSIPConfigReq Struct Reference

Data Fields

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

8.852.1 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

8.852.2 Field Documentation

8.852.2.1 BYTE* SetSIPConfigReq::pSigCompEnabled

8.852.2.2 WORD* SetSIPConfigReq::pSIPLocalPort

8.852.2.3 ULONG* SetSIPConfigReq::pSubscribeTimer

8.852.2.4 ULONG* SetSIPConfigReq::pTimerSIPReg

8.852.2.5 ULONG* SetSIPConfigReq::pTimerT1

8.852.2.6 ULONG* SetSIPConfigReq::pTimerT2

8.852.2.7 ULONG* SetSIPConfigReq::pTimerTf

8.853 SetSIPConfigResp Struct Reference

Data Fields

- BYTE * pSettingResp

8.853.1 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

8.853.2 Field Documentation

8.853.2.1 BYTE* SetSIPConfigResp::pSettingResp

8.854 sGetDeviceSeriesResult Struct Reference

Data Fields

- enum eGobiDeviceSeries eDevice
- ULONG uResult

8.854.1 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

8.854.2 Field Documentation

8.854.2.1 enum eGobiDeviceSeries sGetDeviceSeriesResult::eDevice

8.854.2.2 ULONG sGetDeviceSeriesResult::uResult

8.855 sidNid Struct Reference

Data Fields

- WORD nid
- WORD sid

8.855.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

nid	• Network ID
sid	• System ID

8.855.2 Field Documentation

8.855.2.1 WORD sidNid::nid

8.855.2.2 WORD sidNid::sid

8.856 sigInfo Struct Reference

Data Fields

- RSSIThresh * pRSSIThresh
- ECIOThresh * pECIOThresh
- HDRSINRThresh * pHDRSINRThresh
- LTESNRThresh * pLTESNRThresh
- IOThresh * pIOThresh
- RSRQThresh * pRSRQThresh
- RSRPThresh * pRSRPThresh
- LTESigRptCfg * pLTESigRptCfg
- TDSCDMASINRCONFThresh * pTDSCDMASINRCONFThresh

8.856.1 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

8.856.2 Field Documentation

8.856.2.1 ECIOThresh∗ sigInfo::pECIOThresh

8.856.2.2 HDRSINRThresh∗ sigInfo::pHDRSINRThresh

8.856.2.3 IOThresh∗ sigInfo::pIOThresh

8.856.2.4 LTESigRptCfg∗ sigInfo::pLTESigRptCfg

8.856.2.5 LTESNRThresh∗ sigInfo::pLTESNRThresh

8.856.2.6 RSRPThresh∗ sigInfo::pRSRPThresh

8.856.2.7 RSRQThresh∗ sigInfo::pRSRQThresh

8.856.2.8 RSSIThresh∗ sigInfo::pRSSIThresh

8.856.2.9 TDSCDMASINRCONFThresh∗ sigInfo::pTDSCDMASINRCONFThresh

8.857 signalInfo Struct Reference

Data Fields

- BYTE signalType
- BYTE alertPitch
- BYTE signal

8.857.1 Detailed Description

This structure contains Signal Information

Parameters

signalType	• Call identifier for the call.
alertPitch	• Signal Information
signal	• Caller ID Information

8.857.2 Field Documentation

8.857.2.1 BYTE signalInfo::alertPitch

8.857.2.2 BYTE signalInfo::signal

8.857.2.3 BYTE signalInfo::signalType

8.858 SignalStrengthDataType Struct Reference

Data Fields

- BYTE thresholdsSize
- INT8 thresholds [5]

8.858.1 Field Documentation

8.858.1.1 INT8 SignalStrengthDataType::thresholds[5]

8.858.1.2 BYTE SignalStrengthDataType::thresholdsSize

8.859 slot_t Struct Reference

Data Fields

- uint32_t uPhyCardStatus
- uint32_t uPhySlotStatus
- uint8_t bLogicalSlot
- uint8_t bICCIDLength
- uint8_t bICCID [255]

8.859.1 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.

8.859.2 Field Documentation

8.859.2.1 uint8_t slot_t::bICCID[255]

8.859.2.2 uint8_t slot_t::bICCIDLength

8.859.2.3 uint8_t slot_t::bLogicalSlot

8.859.2.4 uint32_t slot_t::uPhyCardStatus

8.859.2.5 uint32_t slot_t::uPhySlotStatus

8.860 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]

8.860.1 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.

8.860.2 Field Documentation

8.860.2.1 appStats slotInf::AppStatus[10]

8.860.2.2 uint8_t slotInf::cardState

8.860.2.3 uint8_t slotInf::errorState

8.860.2.4 uint8_t slotInf::numApp

8.860.2.5 uint8_t slotInf::upinRetries

8.860.2.6 uint8_t slotInf::upinState

8.860.2.7 uint8_t slotInf::upukRetries

8.861 slotInfo Struct Reference

Data Fields

- BYTE cardState
- BYTE upinState
- BYTE upinRetries
- BYTE upukRetries
- BYTE errorState
- BYTE numApp
- appStatus AppStatus [10]

8.861.1 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.

8.861.2 Field Documentation

8.861.2.1 appStatus slotInfo::AppStatus[10]

8.861.2.2 BYTE slotInfo::cardState

8.861.2.3 BYTE slotInfo::errorState

8.861.2.4 BYTE slotInfo::numApp

8.861.2.5 BYTE slotInfo::upinRetries

8.861.2.6 BYTE slotInfo::upinState

8.861.2.7 BYTE slotInfo::upukRetries

8.862 slots_t Struct Reference

Data Fields

- slot_t uimSlotStatus [255]

8.862.1 Field Documentation

8.862.1.1 slot_t slots_t::uimSlotStatus[255]

8.863 slqsautoconnect Struct Reference

Data Fields

- BOOL action
- BYTE acsetting
- BYTE acroamsetting

8.863.1 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

8.863.2 Field Documentation

8.863.2.1 BYTE slqsautoconnect::acroamsetting

8.863.2.2 BYTE slqsautoconnect::acsetting

8.863.2.3 BOOL slqsautoconnect::action

8.864 SLQSDeleteProfileParams Struct Reference

Data Fields

- BYTE profileType
- BYTE profileIndex

8.864.1 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)

8.864.2 Field Documentation

8.864.2.1 BYTE SLQSDeleteProfileParams::profileIndex

8.864.2.2 BYTE SLQSDeleteProfileParams::profileType

8.865 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]

8.865.1 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

8.865.2 Field Documentation

8.865.2.1 CHAR slqsfwinfo_s::appversion_str[85]

8.865.2.2 CHAR slqsfwinfo_s::bootversion_str[85]

8.865.2.3 CHAR slqsfwinfo_s::carrier_str[20]

8.865.2.4 CHAR slqsfwinfo_s::cur_carr_name[17]

8.865.2.5 CHAR slqsfwinfo_s::cur_carr_rev[13]

8.865.2.6 CHAR slqsfwinfo_s::modelid_str[20]

8.865.2.7 CHAR slqsfwinfo_s::packageid_str[85]

8.865.2.8 CHAR slqsfwinfo_s::priversion_str[16]

8.865.2.9 CHAR slqsfwinfo_s::sku_str[15]

8.866 SlqsNas3GppNetworkInfo Struct Reference

Data Fields

- WORD MCC
- WORD MNC
- ULONG InUse
- ULONG Roaming
- ULONG Forbidden
- ULONG Preferred
- CHAR Description [255]

8.866.1 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.

8.866.2 Field Documentation

8.866.2.1 CHAR SlqsNas3GppNetworkInfo::Description[255]

8.866.2.2 ULONG SlqsNas3GppNetworkInfo::Forbidden

8.866.2.3 ULONG SlqsNas3GppNetworkInfo::InUse

8.866.2.4 WORD SlqsNas3GppNetworkInfo::MCC

8.866.2.5 WORD SlqsNas3GppNetworkInfo::MNC

8.866.2.6 ULONG SlqsNas3GppNetworkInfo::Preferred

8.866.2.7 ULONG SlqsNas3GppNetworkInfo::Roaming

8.867 SlqsNasPcsDigit Struct Reference

Data Fields

- WORD MCC
- WORD MNC
- BYTE includes_pcs_digit

8.867.1 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

8.867.2 Field Documentation

8.867.2.1 BYTE SlqsNasPcsDigit::includes_pcs_digit

8.867.2.2 WORD SlqsNasPcsDigit::MCC

8.867.2.3 WORD SlqsNasPcsDigit::MNC

8.868 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

8.868.1 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 ULONG slqssendasyncsmsparams_s::messageFormat

8.868.2.2 ULONG slqssendasyncsmsparams_s::messageSize

8.868.2.3 BYTE* slqssendasyncsmsparams_s::pFollowOnDC

8.868.2.4 BYTE* slqssendasyncsmsparams_s::pForceOnDC

8.868.2.5 BYTE* slqssendasyncsmsparams_s::pLinktimer

8.868.2.6 BYTE* slqssendasyncsmsparams_s::pMessage

8.868.2.7 BYTE* slqssendasyncsmsparams_s::pRetryMessage

8.868.2.8 ULONG* slqssendasyncsmsparams_s::pRetryMessageId

8.868.2.9 BYTE* slqssendasyncsmsparams_s::pServiceOption

8.868.2.10 BYTE* slqssendasyncsmsparams_s::pSmsOnIms

8.868.2.11 ULONG* slqssendasyncsmsparams_s::pUserData

8.869 slqssendsmsparams_s Struct Reference

Data Fields

- ULONG messageFormat
- ULONG messageSize
- BYTE * pMessage
- USHORT messageID
- ULONG messageFailureCode
- BYTE * pLinktimer
- BYTE * pSmsOnIms

8.869.1 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

8.869.2 Field Documentation

8.869.2.1 ULONG slqssendsmsparams_s::messageFailureCode

8.869.2.2 ULONG slqssendsmsparams_s::messageFormat

8.869.2.3 USHORT slqssendsmsparams_s::messageID

8.869.2.4 ULONG slqssendsmsparams_s::messageSize

8.869.2.5 BYTE* slqssendsmsparams_s::pLinktimer

8.869.2.6 BYTE* slqssendsmsparams_s::pMessage

8.869.2.7 BYTE* slqssendsmsparams_s::pSmsOnIms

8.870 slqsSessionStateInfo Struct Reference

Data Fields

- qaQmiInterfaceInfo * pQmiInterfaceInfo
- ULONG reconfiguration_required
- ULONG state
- ULONG sessionEndReason

8.870.1 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

8.870.2 Field Documentation

8.870.2.1 qaQmiInterfaceInfo* slqsSessionStateInfo::pQmiInterfaceInfo

8.870.2.2 ULONG slqsSessionStateInfo::reconfiguration_required

8.870.2.3 ULONG slqsSessionStateInfo::sessionEndReason

8.870.2.4 ULONG slqsSessionStateInfo::state

8.871 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

8.871.1 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

8.871.2 Field Documentation

8.871.2.1 struct ecioListElement slqsSignalStrengthInfo::ecioList[18]

8.871.2.2 USHORT slqsSignalStrengthInfo::ecioListLen

8.871.2.3 struct errorRateListElement slqsSignalStrengthInfo::errorRateList[18]

8.871.2.4 USHORT slqsSignalStrengthInfo::errorRateListLen

8.871.2.5 INT32 slqsSignalStrengthInfo::Io

8.871.2.6 SHORT slqsSignalStrengthInfo::ltersrp

8.871.2.7 SHORT slqsSignalStrengthInfo::ltesnr

8.871.2.8 struct rsrqInformation slqsSignalStrengthInfo::rsrqInfo

8.871.2.9 struct rxSignalStrengthListElement slqsSignalStrengthInfo::rxSignalStrengthList[18]

8.871.2.10 USHORT slqsSignalStrengthInfo::rxSignalStrengthListLen

8.871.2.11 USHORT slqsSignalStrengthInfo::signalStrengthReqMask

8.871.2.12 BYTE slqsSignalStrengthInfo::sinr

8.872 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

8.872.1 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

8.872.2 Field Documentation

8.872.2.1 BYTE SLQSSignalStrengthsIndReq::ecioDelta

8.872.2.2 SHORT SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.872.2.3 BYTE SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.872.2.4 BYTE SLQSSignalStrengthsIndReq::ioDelta

8.872.2.5 BYTE SLQSSignalStrengthsIndReq::lteRsrpDelta

8.872.2.6 WORD SLQSSignalStrengthsIndReq::lteSnrDelta

8.872.2.7 BYTE SLQSSignalStrengthsIndReq::rsrqDelta

8.872.2.8 BYTE SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.872.2.9 BYTE SLQSSignalStrengthsIndReq::sinrDelta

8.872.2.10 BYTE SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.872.2.11 BYTE SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.873 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

8.873.1 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

8.873.2 Field Documentation

8.873.2.1 struct ecioListElement SLQSSignalStrengthsInformation::ecioInfo

8.873.2.2 struct errorRateListElement SLQSSignalStrengthsInformation::errorRateInfo

8.873.2.3 ULONG SLQSSignalStrengthsInformation::io

8.873.2.4 struct lteRsrpinformation SLQSSignalStrengthsInformation::lteRsrpinfo

8.873.2.5 struct lteSnrinformation SLQSSignalStrengthsInformation::lteSnrinfo

8.873.2.6 struct rsrqInformation SLQSSignalStrengthsInformation::rsrqInfo

8.873.2.7 struct rxSignalStrengthListElement SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.873.2.8 BYTE SLQSSignalStrengthsInformation::sinr

8.874 slqsWdsEventInfo Struct Reference

Data Fields

- qaQmiInterfaceInfo * pQmiInterfaceInfo
- ULONG * pDormancyStatus
- ULONG * pDataBearer
- ULONG * pPacketsCountTX
- ULONG * pPacketsCountRX
- ULONGLONG * pTotalBytesTX
- ULONGLONG * pTotalBytesRX

8.874.1 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)

8.874.2 Field Documentation

8.874.2.1 ULONG* slqsWdsEventInfo::pDataBearer

8.874.2.2 ULONG* slqsWdsEventInfo::pDormancyStatus

8.874.2.3 ULONG* slqsWdsEventInfo::pPacketsCountRX

8.874.2.4 ULONG* slqsWdsEventInfo::pPacketsCountTX

8.874.2.5 qaQmiInterfaceInfo* slqsWdsEventInfo::pQmiInterfaceInfo

8.874.2.6 ULONGLONG* slqsWdsEventInfo::pTotalBytesRX

8.874.2.7 ULONGLONG* slqsWdsEventInfo::pTotalBytesTX

8.875 sms_BroadcastConfig Struct Reference

Data Fields

- uint16_t fromServiceId
- uint16_t toServiceId
- uint8_t selected

8.875.1 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

8.875.2 Field Documentation

8.875.2.1 uint16_t sms_BroadcastConfig::fromServiceId

8.875.2.2 uint8_t sms_BroadcastConfig::selected

8.875.2.3 uint16_t sms_BroadcastConfig::toServiceId

8.876 sms_CDMABroadcastConfig Struct Reference

Data Fields

- uint16_t serviceCategory
- uint16_t language
- uint8_t selected

8.876.1 Detailed Description

This structure contains CDMABroadcastConfig parameters

Parameters

serviceCategory	• Service category
language	• Language
selected	• Specified service_category and language – 0x00 – Not selected – 0x01 – Selected

8.876.2 Field Documentation

8.876.2.1 uint16_t sms_CDMABroadcastConfig::language

8.876.2.2 uint8_t sms_CDMABroadcastConfig::selected

8.876.2.3 uint16_t sms_CDMABroadcastConfig::serviceCategory

8.877 sms_getIndicationReg Struct Reference

Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

8.877.1 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.

8.877.2 Field Documentation

8.877.2.1 uint8_t* sms_getIndicationReg::pRegCallStatInfoEvt

8.877.2.2 uint8_t* sms_getIndicationReg::pRegTransLayerInfoEvt

8.877.2.3 uint8_t* sms_getIndicationReg::pRegTransNWRegInfoEvt

8.878 sms_getMsgWaitingInfo Struct Reference

Data Fields

- uint8_t numInstances
- sms_messageWaitingInfoContent msgWaitInfo [0xFF]

8.878.1 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.

8.878.2 Field Documentation

8.878.2.1 sms_messageWaitingInfoContent sms_getMsgWaitingInfo::msgWaitInfo[0xFF]

8.878.2.2 uint8_t sms_getMsgWaitingInfo::numInstances

8.879 sms_getTransLayerInfo Struct Reference

Data Fields

- uint8_t * pRegInd
- sms_transLayerInfo * pTransLayerInfo

8.879.1 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.

8.879.2 Field Documentation

8.879.2.1 uint8_t* sms_getTransLayerInfo::pRegInd

8.879.2.2 sms_transLayerInfo* sms_getTransLayerInfo::pTransLayerInfo

8.880 sms_getTransNWRegInfo Struct Reference

Data Fields

- uint8_t * pRegStatus

8.880.1 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.

8.880.2 Field Documentation

8.880.2.1 uint8_t* sms_getTransNWRegInfo::pRegStatus

8.881 sms_maxStorageSizeReq Struct Reference

Data Fields

- uint8_t storageType
- uint8_t * pMessageMode

8.881.1 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

8.881.2 Field Documentation

8.881.2.1 uint8_t* sms_maxStorageSizeReq::pMessageMode

8.881.2.2 uint8_t sms_maxStorageSizeReq::storageType

8.882 sms_maxStorageSizeResp Struct Reference

Data Fields

- uint32_t maxStorageSize
- uint32_t freeSlots

8.882.1 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.

8.882.2 Field Documentation

8.882.2.1 uint32_t sms_maxStorageSizeResp::freeSlots

8.882.2.2 uint32_t sms_maxStorageSizeResp::maxStorageSize

8.883 sms_messageWaitingInfoContent Struct Reference

Data Fields

- uint8_t msgType
- uint8_t activeInd
- uint8_t msgCount

8.883.1 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

8.883.2 Field Documentation

8.883.2.1 uint8_t sms_messageWaitingInfoContent::activeInd

8.883.2.2 uint8_t sms_messageWaitingInfoContent::msgCount

8.883.2.3 uint8_t sms_messageWaitingInfoContent::msgType

8.884 sms_msgProtocolResp Struct Reference

Data Fields

- uint8_t msgProtocol

8.884.1 Detailed Description

This structure contains get message protocol response parameters

Parameters

msgProtocol	- • Message Protocol • Values: – 0x00 - MESSAGE_PROTOCOL_CDMA – 0x01 - MESSAGE_PROTOCOL_WCDMA

8.884.2 Field Documentation

8.884.2.1 uint8_t sms_msgProtocolResp::msgProtocol

8.885 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference

Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_CDMABroadcastConfig CDMABroadcastConfig [0x05]

8.885.1 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

8.885.2 Field Documentation

8.885.2.1 uint8_t sms_qaQmi3GPP2BroadcastCfgInfo::activated_ind

8.885.2.2 sms_CDMABroadcastConfig sms_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]

8.885.2.3 uint16_t sms_qaQmi3GPP2BroadcastCfgInfo::num_instances

8.886 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference

Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_BroadcastConfig broadcastConfig [0x05]

8.886.1 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

8.886.2 Field Documentation

8.886.2.1 uint8_t sms_qaQmi3GPPBroadcastCfgInfo::activated_ind

8.886.2.2 sms_BroadcastConfig sms_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

8.886.2.3 uint16_t sms_qaQmi3GPPBroadcastCfgInfo::num_instances

8.887 sms_routeEntry Struct Reference

Data Fields

- uint8_t messageType
- uint8_t messageClass

- uint8_t routeStorage
- uint8_t receiptAction

8.887.1 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)

8.887.2 Field Documentation

8.887.2.1 uint8_t sms_routeEntry::messageClass

8.887.2.2 uint8_t sms_routeEntry::messageType

8.887.2.3 uint8_t sms_routeEntry::receiptAction

8.887.2.4 uint8_t sms_routeEntry::routeStorage

8.888 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

8.888.1 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.

8.888.2 Field Documentation

8.888.2.1 uint32_t sms_sendAsyncsmsParams::messageFormat

8.888.2.2 uint32_t sms_sendAsyncsmsParams::messageSize

8.888.2.3 uint8_t* sms_sendAsyncsmsParams::pFollowOnDC

8.888.2.4 uint8_t* sms_sendAsyncsmsParams::pForceOnDC

8.888.2.5 uint8_t* sms_sendAsyncsmsParams::pLinktimer

8.888.2.6 uint8_t* sms_sendAsyncsmsParams::pMessage

8.888.2.7 uint8_t* sms_sendAsyncsmsParams::pRetryMessage

8.888.2.8 uint32_t* sms_sendAsyncsmsParams::pRetryMessageId

8.888.2.9 uint8_t* sms_sendAsyncsmsParams::pServiceOption

8.888.2.10 uint8_t* sms_sendAsyncsmsParams::pSmsOnIms

8.888.2.11 uint32_t* sms_sendAsyncsmsParams::pUserData

8.889 sms_setIndicationReg Struct Reference

Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

8.889.1 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

8.889.2 Field Documentation

8.889.2.1 uint8_t* sms_setIndicationReg::pRegCallStatInfoEvt

8.889.2.2 uint8_t* sms_setIndicationReg::pRegTransLayerInfoEvt

8.889.2.3 uint8_t* sms_setIndicationReg::pRegTransNWRegInfoEvt

8.890 sms_setRoutesReq Struct Reference

Data Fields

- uint16_t numOfRoutes

- sms_routeEntry routeList [0x0A]
- uint8_t * pTransferStatusReport

8.890.1 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)

8.890.2 Field Documentation

8.890.2.1 uint16_t sms_setRoutesReq::numOfRoutes

8.890.2.2 uint8_t* sms_setRoutesReq::pTransferStatusReport

8.890.2.3 sms_routeEntry sms_setRoutesReq::routeList[0x0A]

8.891 sms_transLayerInfo Struct Reference

Data Fields

- uint8_t TransType
- uint8_t TransCap

8.891.1 Detailed Description

This structure contains Transport Layer Information

Parameters

TransType	• Transport Type – 0x00 - IMS
TransCap	• Transport Capability • Values: – 0x00 - CDMA – 0x01 - GW

8.891.2 Field Documentation

8.891.2.1 uint8_t sms_transLayerInfo::TransCap

8.891.2.2 uint8_t sms_transLayerInfo::TransType

8.892 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

8.892.1 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.

8.892.2 Field Documentation

8.892.2.1 BYTE SMSAsyncRawSend_s::alphaIDLen

8.892.2.2 WORD SMSAsyncRawSend_s::causeCode

8.892.2.3 BYTE SMSAsyncRawSend_s::errorClass

8.892.2.4 WORD SMSAsyncRawSend_s::messageID

8.892.2.5 BYTE SMSAsyncRawSend_s::msgDelFailureCause

8.892.2.6 BYTE SMSAsyncRawSend_s::msgDelFailureType

8.892.2.7 BYTE* SMSAsyncRawSend_s::pAlphaID

8.892.2.8 WORD SMSAsyncRawSend_s::RPCause

8.892.2.9 WORD SMSAsyncRawSend_s::sendStatus

8.892.2.10 BYTE SMSAsyncRawSend_s::TPCause

8.892.2.11 ULONG SMSAsyncRawSend_s::userData

8.893 SMSCAddress Struct Reference

Data Fields

- BYTE length
- BYTE data [256]

8.893.1 Detailed Description

This structure holds SMSC information

Parameters

length	• Number of sets of following element
data	• SMSC address

8.893.2 Field Documentation

8.893.2.1 BYTE SMSCAddress::data[256]

8.893.2.2 BYTE SMSCAddress::length

8.894 sMSCAddress Struct Reference

Data Fields

- uint8_t length
- uint8_t data [256]

8.894.1 Detailed Description

Parameters

length	• Number of sets of following element
data	• SMSC address

8.894.2 Field Documentation

8.894.2.1 uint8_t sMSCAddress::data[256]

8.894.2.2 uint8_t sMSCAddress::length

8.895 sMSCAddressTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSCAddressInfo SMSCInfo

8.895.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
SMSCInfo	• SMSC Address • See sMSCAddressInfo for more information

8.895.2 Field Documentation

8.895.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

8.895.2.2 uint8_t sMSCAddressTlv::TlvPresent

8.896 sMSEtwsMessage Struct Reference

Data Fields

- uint8_t notificationType
- uint16_t length
- uint8_t data [1254]

8.896.1 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

8.896.2 Field Documentation

8.896.2.1 uint8_t sMSEtwsMessage::data[1254]

8.896.2.2 uint16_t sMSEtwsMessage::length

8.896.2.3 uint8_t sMSEtwsMessage::notificationType

8.897 SMSEtwsMessage Struct Reference

Data Fields

- BYTE notificationType
- WORD length
- BYTE data [1254]

8.897.1 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

8.897.2 Field Documentation

8.897.2.1 BYTE SMSEtwsMessage::data[1254]

8.897.2.2 WORD SMSEtwsMessage::length

8.897.2.3 BYTE SMSEtwsMessage::notificationType

8.898 sMSEtwsMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSEtwsMessageInfo EtwsMessageInfo

8.898.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
EtwsMessage-Info	• ETWS Message • See sMSEtwsMessageInfo for more information

8.898.2 Field Documentation

8.898.2.1 sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo

8.898.2.2 uint8_t sMSEtwsMessageTlv::TlvPresent

8.899 SMSEtwsPlmn Struct Reference

Data Fields

- WORD mobileCountryCode
- WORD mobileNetworkCode

8.899.1 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

8.899.2 Field Documentation

8.899.2.1 WORD SMSEtwsPlmn::mobileCountryCode

8.899.2.2 WORD SMSEtwsPlmn::mobileNetworkCode

8.900 sMSEtwsPlmn Struct Reference

Data Fields

- uint16_t mobileCountryCode
- uint16_t mobileNetworkCode

8.900.1 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

8.900.2 Field Documentation

8.900.2.1 uint16_t sMSEtwsPlmn::mobileCountryCode

8.900.2.2 uint16_t sMSEtwsPlmn::mobileNetworkCode

8.901 SMSEventInfo_s Struct Reference

Data Fields

- BYTE smsEventType

- SMSMTMessageInfo * pMTMessageInfo
- SMSTransferRouteMTMessageInfo * pTransferRouteMTMessageInfo
- SMSMessageModeInfo * pMessageModeInfo
- SMSEtwsMessageInfo * pEtwsMessageInfo
- SMSEtwsPlmnInfo * pEtwsPlmnInfo
- SMSCAddressInfo * pSMSCAddressInfo
- SMSOnIMSInfo * pSMSOnIMSInfo

8.901.1 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

8.901.2 Field Documentation

8.901.2.1 SMSEtwsMessageInfo* SMSEventInfo_s::pEtwsMessageInfo

8.901.2.2 SMSEtwsPlmnInfo* SMSEventInfo_s::pEtwsPlmnInfo

8.901.2.3 SMSMessageModeInfo* SMSEventInfo_s::pMessageModeInfo

8.901.2.4 SMSMTMessageInfo* SMSEventInfo_s::pMTMessageInfo

8.901.2.5 SMSCAddressInfo* SMSEventInfo_s::pSMSCAddressInfo

8.901.2.6 SMSOnIMSInfo* SMSEventInfo_s::pSMSOnIMSInfo

8.901.2.7 SMSTransferRouteMTMessageInfo* SMSEventInfo_s::pTransferRouteMTMessageInfo

8.901.2.8 BYTE SMSEventInfo_s::smsEventType

8.902 smsMaxStorageSizeReq Struct Reference

Data Fields

- BYTE storageType
- BYTE * pMessageMode

8.902.1 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

8.902.2 Field Documentation

8.902.2.1 BYTE* smsMaxStorageSizeReq::pMessageMode

8.902.2.2 BYTE smsMaxStorageSizeReq::storageType

8.903 smsMaxStorageSizeResp Struct Reference

Data Fields

- ULONG maxStorageSize
- ULONG freeSlots

8.903.1 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.

8.903.2 Field Documentation

8.903.2.1 ULONG smsMaxStorageSizeResp::freeSlots

8.903.2.2 ULONG smsMaxStorageSizeResp::maxStorageSize

8.904 SMSMemoryInfo Struct Reference

Data Fields

- BYTE storageType
- BYTE messageMode

8.904.1 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

8.904.2 Field Documentation

8.904.2.1 BYTE SMSMemoryInfo::messageMode

8.904.2.2 BYTE SMSMemoryInfo::storageType

8.905 SMSMessageMode Struct Reference

Data Fields

- BYTE messageMode

8.905.1 Detailed Description

This structure holds information related to message mode

Parameters

messageMode	• Message mode 0x00 - CDMA 0x01 - GW

8.905.2 Field Documentation

8.905.2.1 BYTE SMSMessageMode::messageMode

8.906 sMSMessageMode Struct Reference

Data Fields

- uint8_t messageMode

8.906.1 Detailed Description

Parameters

messageMode	Message Mode

8.906.2 Field Documentation

8.906.2.1 uint8_t sMSMessageMode::messageMode

8.907 smsMsgprotocolResp Struct Reference

Data Fields

- BYTE msgProtocol

8.907.1 Detailed Description

This structure contains get message protocol response parameters

Parameters

msgProtocol	- • Message Protocol • Values: – 0x00 - MESSAGE_PROTOCOL_CDMA – 0x01 - MESSAGE_PROTOCOL_WCDMA

8.907.2 Field Documentation

8.907.2.1 BYTE smsMsgprotocolResp::msgProtocol

8.908 SMSMTMessage Struct Reference

Data Fields

- ULONG storageType
- ULONG messageIndex

8.908.1 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

8.908.2 Field Documentation

8.908.2.1 ULONG SMSMTMessage::messageIndex

8.908.2.2 ULONG SMSMTMessage::storageType

8.909 sMSMTMessage Struct Reference

Data Fields

- uint32_t storageType
- uint32_t messageIndex

8.909.1 Detailed Description

Parameters

storageType	memory storage 0x00-UIM 0x01-NV
messageIndex	MT Message index

8.909.2 Field Documentation

8.909.2.1 uint32_t sMSMTMessage::messageIndex

8.909.2.2 uint32_t sMSMTMessage::storageType

8.910 sMSOnIMS Struct Reference

Data Fields

- uint8_t smsOnIMS

8.910.1 Detailed Description

Parameters

smsOnIMS	SMS on IMS

8.910.2 Field Documentation

8.910.2.1 uint8_t sMSOnIMS::smsOnIMS

8.911 SMSOnIMS Struct Reference

Data Fields

- BYTE smsOnIMS

8.911.1 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.

8.911.2 Field Documentation

8.911.2.1 BYTE SMSOnIMS::smsOnIMS

8.912 sMSOnIMSTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSOnIMSInfo IMSInfo

8.912.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
IMSInfo	• SMS on IMS • See sMSOnIMSInfo for more information

8.912.2 Field Documentation

8.912.2.1 sMSOnIMSInfo sMSOnIMSTlv::IMSInfo

8.912.2.2 uint8_t sMSOnIMSTlv::TlvPresent

8.913 smsRouteEntry Struct Reference

Data Fields

- BYTE messageType
- BYTE messageClass
- BYTE routeStorage
- BYTE receiptAction

8.913.1 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)

8.913.2 Field Documentation

8.913.2.1 BYTE smsRouteEntry::messageClass

8.913.2.2 BYTE smsRouteEntry::messageType

8.913.2.3 BYTE smsRouteEntry::receiptAction

8.913.2.4 BYTE smsRouteEntry::routeStorage

8.914 smsSetRoutesReq Struct Reference

Data Fields

- WORD numOfRoutes
- smsRouteEntry routeList [0x0A]
- BYTE * pTransferStatusReport

8.914.1 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)

8.914.2 Field Documentation

8.914.2.1 WORD smsSetRoutesReq::numOfRoutes

8.914.2.2 BYTE* smsSetRoutesReq::pTransferStatusReport

8.914.2.3 smsRouteEntry smsSetRoutesReq::routeList[0x0A]

8.915 SMSTransferRouteMTMessage Struct Reference

Data Fields

- BYTE ackIndicator
- ULONG transactionID
- BYTE format
- WORD length
- BYTE data [256]

8.915.1 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

8.915.2 Field Documentation

8.915.2.1 BYTE SMSTransferRouteMTMessage::ackIndicator

8.915.2.2 BYTE SMSTransferRouteMTMessage::data[256]

8.915.2.3 BYTE SMSTransferRouteMTMessage::format

8.915.2.4 WORD SMSTransferRouteMTMessage::length

8.915.2.5 ULONG SMSTransferRouteMTMessage::transactionID

8.916 sMSTransferRouteMTMessage Struct Reference

Data Fields

- uint8_t ackIndicator
- uint32_t transactionID
- uint8_t format
- uint16_t length

- uint8_t data [256]

8.916.1 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

8.916.2 Field Documentation

8.916.2.1 uint8_t sMSTransferRouteMTMessage::ackIndicator

8.916.2.2 uint8_t sMSTransferRouteMTMessage::data[256]

8.916.2.3 uint8_t sMSTransferRouteMTMessage::format

8.916.2.4 uint16_t sMSTransferRouteMTMessage::length

8.916.2.5 uint32_t sMSTransferRouteMTMessage::transactionID

8.917 sQosFlowStat Struct Reference

Data Fields

- ULONG bearerId
- ULONG tx_pkt
- ULONG tx_pkt_drp
- ULONGLONG tx_bytes
- ULONGLONG tx_bytes_drp

8.917.1 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

8.917.2 Field Documentation

8.917.2.1 ULONG sQosFlowStat::bearerId

8.917.2.2 ULONGLONG sQosFlowStat::tx_bytes

8.917.2.3 ULONGLONG sQosFlowStat::tx_bytes_drp

8.917.2.4 ULONG sQosFlowStat::tx_pkt

8.917.2.5 ULONG sQosFlowStat::tx_pkt_drp

8.918 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)]

8.918.1 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

8.918.2 Field Documentation

8.918.2.1 ULONG sQosStat::apnId

8.918.2.2 ULONG sQosStat::numQosFlow

8.918.2.3 sQosFlowStat sQosStat::qosFlow[(10)]

8.918.2.4 ULONGLONG sQosStat::total_rx_bytes

8.918.2.5 ULONG sQosStat::total_rx_pkt

8.918.2.6 ULONGLONG sQosStat::total_tx_bytes

8.918.2.7 ULONGLONG sQosStat::total_tx_bytes_drp

8.918.2.8 ULONG sQosStat::total_tx_pkt

8.918.2.9 ULONG sQosStat::total_tx_pkt_drp

8.919 SrvStatusInfo Struct Reference

Data Fields

- BYTE srvStatus
- BYTE isPrefDataPath

8.919.1 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

8.919.2 Field Documentation

8.919.2.1 BYTE SrvStatusInfo::isPrefDataPath

8.919.2.2 BYTE SrvStatusInfo::srvStatus

8.920 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

8.920.1 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

8.920.2 Field Documentation

8.920.2.1 BOOL ssdatasession_params::action

8.920.2.2 ULONG ssdatasession_params::failureReason

8.920.2.3 ULONG ssdatasession_params::failureReasonv4

8.920.2.4 ULONG ssdatasession_params::failureReasonv6

8.920.2.5 BYTE ssdatasession_params::instanceId

8.920.2.6 BYTE ssdatasession_params::ipfamily

8.920.2.7 ULONG* ssdatasession_params::pAuthentication

8.920.2.8 CHAR* ssdatasession_params::pPassword

8.920.2.9 ULONG* ssdatasession_params::pProfileId3GPP

8.920.2.10 ULONG* ssdatasession_params::pProfileId3GPP2

8.920.2.11 ULONG* ssdatasession_params::pTechnology

8.920.2.12 CHAR* ssdatasession_params::pUsername

8.920.2.13 ULONG ssdatasession_params::rcv4

8.920.2.14 ULONG ssdatasession_params::rcv6

8.920.2.15 ULONG ssdatasession_params::sessionId

8.920.2.16 ULONG ssdatasession_params::v4sessionId

8.920.2.17 ULONG ssdatasession_params::v6sessionId

8.920.2.18 ULONG ssdatasession_params::verbFailReason

8.920.2.19 ULONG ssdatasession_params::verbFailReasonType

8.921 SupportedMsgList Struct Reference

Data Fields

- WORD supportedMsgLen
- BYTE supportedMsgs [256]

8.921.1 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.

8.921.2 Field Documentation

8.921.2.1 WORD SupportedMsgList::supportedMsgLen

8.921.2.2 BYTE SupportedMsgList::supportedMsgs[256]

8.922 SUPSInfo Struct Reference

Data Fields

- BYTE svcType
- BYTE isModByCC

8.922.1 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

8.922.2 Field Documentation

8.922.2.1 BYTE SUPSInfo::isModByCC

8.922.2.2 BYTE SUPSInfo::svcType

8.923 SV Struct Reference

Data Fields

- WORD id
- ULONG system
- BYTE mask

8.923.1 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

8.923.2 Field Documentation

8.923.2.1 WORD SV::id

8.923.2.2 BYTE SV::mask

8.923.2.3 ULONG SV::system

8.924 SVInfo Struct Reference

Data Fields

- BYTE len
- SV * pSV

8.924.1 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

8.924.2 Field Documentation

8.924.2.1 BYTE SVInfo::len

8.924.2.2 SV* SVInfo::pSV

8.925 svUsedforFix_s Struct Reference

Data Fields

- BYTE gnssSvUsedList_len
- WORD gnssSvUsedList [255]

8.925.1 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

8.925.2 Field Documentation

8.925.2.1 WORD svUsedforFix_s::gnssSvUsedList[255]

8.925.2.2 BYTE svUsedforFix_s::gnssSvUsedList_len

8.926 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]

8.926.1 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

8.926.2 Field Documentation

8.926.2.1 BYTE SWI_STRUCT_CarrierImage::m_FwBuildId[100]

8.926.2.2 BYTE SWI_STRUCT_CarrierImage::m_FwImageId[16]

8.926.2.3 ULONG SWI_STRUCT_CarrierImage::m_nCarrierId

8.926.2.4 ULONG SWI_STRUCT_CarrierImage::m_nFolderId

8.926.2.5 ULONG SWI_STRUCT_CarrierImage::m_nStorage

8.926.2.6 BYTE SWI_STRUCT_CarrierImage::m_PriBuildId[100]

8.926.2.7 BYTE SWI_STRUCT_CarrierImage::m_PriImageId[16]

8.927 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

8.927.1 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

8.927.2 Field Documentation

8.927.2.1 ULONG SwiLocGetAutoStartResp::fix_rate

8.927.2.2 BOOL SwiLocGetAutoStartResp::fix_rate_reported

8.927.2.3 BYTE SwiLocGetAutoStartResp::fix_type

8.927.2.4 BOOL SwiLocGetAutoStartResp::fix_type_reported

8.927.2.5 BYTE SwiLocGetAutoStartResp::function

8.927.2.6 BOOL SwiLocGetAutoStartResp::function_reported

8.927.2.7 ULONG SwiLocGetAutoStartResp::max_dist

8.927.2.8 BOOL SwiLocGetAutoStartResp::max_dist_reported

8.927.2.9 BYTE SwiLocGetAutoStartResp::max_time

8.927.2.10 BOOL SwiLocGetAutoStartResp::max_time_reported

8.928 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

8.928.1 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

8.928.2 Field Documentation

8.928.2.1 ULONG SwiLocSetAutoStartReq::fix_rate

8.928.2.2 BYTE SwiLocSetAutoStartReq::fix_type

8.928.2.3 BYTE SwiLocSetAutoStartReq::function

8.928.2.4 ULONG SwiLocSetAutoStartReq::max_dist

8.928.2.5 BYTE SwiLocSetAutoStartReq::max_time

8.928.2.6 BOOL SwiLocSetAutoStartReq::set_fix_rate

8.928.2.7 BOOL SwiLocSetAutoStartReq::set_fix_type

8.928.2.8 BOOL SwiLocSetAutoStartReq::set_function

8.928.2.9 BOOL SwiLocSetAutoStartReq::set_max_dist

8.928.2.10 BOOL SwiLocSetAutoStartReq::set_max_time

8.929 swiModemStatusResp Struct Reference

Data Fields

- CommInfo commonInfo
- LTEInfo * pLTEInfo

8.929.1 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

8.929.2 Field Documentation

8.929.2.1 CommInfo swiModemStatusResp::commonInfo

8.929.2.2 LTEInfo* swiModemStatusResp::pLTEInfo

8.930 SwiOTAMsg_s Struct Reference

Data Fields

- ULONG type
- WORD data_len
- BYTE data [2048]
- LteNasReleaseInfo * pLteNasRelInfo
- ULONGLONG * pTime

8.930.1 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

8.930.2 Field Documentation

8.930.2.1 BYTE SwiOTAMsg_s::data[2048]

8.930.2.2 WORD SwiOTAMsg_s::data_len

8.930.2.3 LteNasReleaseInfo* SwiOTAMsg_s::pLteNasRelInfo

8.930.2.4 ULONGLONG* SwiOTAMsg_s::pTime

8.930.2.5 ULONG SwiOTAMsg_s::type

8.931 swiPDPRuntimeSettingsReq Struct Reference

Data Fields

- BYTE contextId
- BYTE contextType

8.931.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

Parameters

contextId	• Context Identifier

contextType	• Identifies technology type – 0 - 3GPP – 1 - 3GPP2

8.931.2 Field Documentation

8.931.2.1 BYTE swiPDPRuntimeSettingsReq::contextId

8.931.2.2 BYTE swiPDPRuntimeSettingsReq::contextType

8.932 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

8.932.1 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.

8.932.2 Field Documentation

8.932.2.1 CHAR* swiPDPRuntimeSettingsResp::pAPNName

8.932.2.2 BYTE* swiPDPRuntimeSettingsResp::pBearerId

8.932.2.3 BYTE* swiPDPRuntimeSettingsResp::pContextId

8.932.2.4 ULONG* swiPDPRuntimeSettingsResp::pIPv4Address

8.932.2.5 ULONG* swiPDPRuntimeSettingsResp::pIPv4GWAddress

8.932.2.6 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6Address

8.932.2.7 struct IPV6AddressInfo* swiPDPRuntimeSettingsResp::pIPv6GWAddress

8.932.2.8 ULONG* swiPDPRuntimeSettingsResp::pPrDNSIPv4Address

8.932.2.9 WORD* swiPDPRuntimeSettingsResp::pPrDNSIPv6Address

8.932.2.10 ULONG* swiPDPRuntimeSettingsResp::pPrPCSCFIPv4Address

8.932.2.11 WORD* swiPDPRuntimeSettingsResp::pPrPCSCFIPv6Address

8.932.2.12 ULONG* swiPDPRuntimeSettingsResp::pSeDNSIPv4Address

8.932.2.13 WORD* swiPDPRuntimeSettingsResp::pSeDNSIPv6Address

8.932.2.14 ULONG* swiPDPRuntimeSettingsResp::pSePCSCFIPv4Address

8.932.2.15 WORD* swiPDPRuntimeSettingsResp::pSePCSCFIPv6Address

8.933 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

8.933.1 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

8.933.2 Field Documentation

8.933.2.1 BYTE swiQosFilter::index

8.933.2.2 ULONG* swiQosFilter::pEspSpi

8.933.2.3 WORD* swiQosFilter::pId

8.933.2.4 IPv4Addr* swiQosFilter::pIPv4DstAddr

8.933.2.5 IPv4Addr* swiQosFilter::pIPv4SrcAddr

8.933.2.6 IPv6Addr* swiQosFilter::pIPv6DstAddr

8.933.2.7 ULONG* swiQosFilter::pIPv6Label

8.933.2.8 IPv6Addr* swiQosFilter::pIPv6SrcAddr

8.933.2.9 IPv6TrafCls* swiQosFilter::pIPv6TrafCls

8.933.2.10 BYTE* swiQosFilter::pNxtHdrProto

8.933.2.11 WORD* swiQosFilter::pPrecedence

8.933.2.12 Port* swiQosFilter::pTCPDstPort

8.933.2.13 Port* swiQosFilter::pTCPSrcPort

8.933.2.14 Tos* swiQosFilter::pTos

8.933.2.15 Port* swiQosFilter::pTranDstPort

8.933.2.16 Port* swiQosFilter::pTranSrcPort

8.933.2.17 Port* swiQosFilter::pUDPDstPort

8.933.2.18 Port* swiQosFilter::pUDPSrcPort

8.933.2.19 BYTE swiQosFilter::version

8.934 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

8.934.1 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
pMinPolicedPktSz	• 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
pMaxAllowedPktSz	• 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
p3GPPResResidualBER	• 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

8.934.2 Field Documentation

8.934.2.1 BYTE swiQosFlow::index

8.934.2.2 BYTE* swiQosFlow::p3GPP2Pri

8.934.2.3 BYTE* swiQosFlow::p3GPPImCn

8.934.2.4 WORD* swiQosFlow::p3GPPResResidualBER

8.934.2.5 BYTE* swiQosFlow::p3GPPSigInd

8.934.2.6 BYTE* swiQosFlow::p3GPPTraHdlPri

8.934.2.7 dataRate* swiQosFlow::pDataRate

8.934.2.8 ULONG* swiQosFlow::pJitter

8.934.2.9 ULONG* swiQosFlow::pLatency

8.934.2.10 BYTE* swiQosFlow::pLteQci

8.934.2.11 ULONG* swiQosFlow::pMaxAllowedPktSz

8.934.2.12 ULONG* swiQosFlow::pMinPolicedPktSz

8.934.2.13 pktErrRate* swiQosFlow::pPktErrRate

8.934.2.14 WORD* swiQosFlow::pProfileId3GPP2

8.934.2.15 tokenBucket* swiQosFlow::pTokenBucket

8.934.2.16 BYTE* swiQosFlow::pTrafficClass

8.935 swiQosGranted Struct Reference

Data Fields

- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow

8.935.1 Detailed Description

This structure contains the QoS granted flow

Parameters

pTxFlow	See swiQosFlow for more information
pRxFlow	See swiQosFlow for more information

8.935.2 Field Documentation

8.935.2.1 swiQosFlow* swiQosGranted::pRxFlow

8.935.2.2 swiQosFlow* swiQosGranted::pTxFlow

8.936 swiQosIds Struct Reference

Data Fields

- BYTE sz
- ULONG * pIds

8.936.1 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

8.936.2 Field Documentation

8.936.2.1 ULONG* swiQosIds::pIds

8.936.2.2 BYTE swiQosIds::sz

8.937 swiQosModifyReq Struct Reference

Data Fields

- ULONG id
- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow
- swiQosFilter * pTxFilter
- swiQosFilter * pRxFilter

8.937.1 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

8.937.2 Field Documentation

8.937.2.1 ULONG swiQosModifyReq::id

8.937.2.2 swiQosFilter* swiQosModifyReq::pRxFilter

8.937.2.3 swiQosFlow* swiQosModifyReq::pRxFlow

8.937.2.4 swiQosFilter* swiQosModifyReq::pTxFilter

8.937.2.5 swiQosFlow* swiQosModifyReq::pTxFlow

8.938 swiQosReq Struct Reference

Data Fields

- BYTE index
- swiQosFlow * pTxFlow
- swiQosFlow * pRxFlow
- swiQosFilter * pTxFilter
- swiQosFilter * pRxFilter

8.938.1 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

8.938.2 Field Documentation

8.938.2.1 BYTE swiQosReq::index

8.938.2.2 swiQosFilter* swiQosReq::pRxFilter

8.938.2.3 swiQosFlow* swiQosReq::pRxFlow

8.938.2.4 swiQosFilter* swiQosReq::pTxFilter

8.938.2.5 swiQosFlow* swiQosReq::pTxFlow

8.939 swiRMTrasnferStaticsReq Struct Reference

Data Fields

- BYTE bResetStatistics
- ULONG ulMask

8.939.1 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

8.939.2 Field Documentation

8.939.2.1 BYTE swiRMTrasnferStaticsReq::bResetStatistics

8.939.2.2 ULONG swiRMTrasnferStaticsReq::ulMask

8.940 sysInfoCommon Struct Reference

Data Fields

- BYTE srvDomainValid
- BYTE srvDomain
- BYTE srvCapabilityValid
- BYTE srvCapability
- BYTE roamStatusValid
- BYTE roamStatus
- BYTE isSysForbiddenValid
- BYTE isSysForbidden

8.940.1 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

8.940.2 Field Documentation

8.940.2.1 BYTE sysInfoCommon::isSysForbidden

8.940.2.2 BYTE sysInfoCommon::isSysForbiddenValid

8.940.2.3 BYTE sysInfoCommon::roamStatus

8.940.2.4 BYTE sysInfoCommon::roamStatusValid

8.940.2.5 BYTE sysInfoCommon::srvCapability

8.940.2.6 BYTE sysInfoCommon::srvCapabilityValid

8.940.2.7 BYTE sysInfoCommon::srvDomain

8.940.2.8 BYTE sysInfoCommon::srvDomainValid

8.941 t_gpsTime Struct Reference

Data Fields

- USHORT gpsWeek
- ULONG gpsTimeOfWeekMs

8.941.1 Field Documentation

8.941.1.1 ULONG t_gpsTime::gpsTimeOfWeekMs

8.941.1.2 USHORT t_gpsTime::gpsWeek

8.942 t_sensor Struct Reference

Data Fields

- ULONG usageMask
- ULONG aidingIndicatorMask

8.942.1 Field Documentation

8.942.1.1 ULONG t_sensor::aidingIndicatorMask

8.942.1.2 ULONG t_sensor::usageMask

8.943 t_Sv Struct Reference

Data Fields

- BYTE len
- USHORT entries [255]

8.943.1 Field Documentation

8.943.1.1 USHORT t_Sv::entries[255]

8.943.1.2 BYTE t_Sv::len

8.944 TDSCDMAECIOThresh Struct Reference

Data Fields

- BYTE TDSCDMAECIOThreshListLen
- ULONG * pTDSCDMAECIOThreshList

8.944.1 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.

8.944.2 Field Documentation

8.944.2.1 ULONG* TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.944.2.2 BYTE TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.945 TDSCDMARSCPThresh Struct Reference

Data Fields

- BYTE TDSCDMARSCPThreshListLen
- WORD * pTDSCDMARSCPThreshList

8.945.1 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).

8.945.2 Field Documentation

8.945.2.1 WORD* TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.945.2.2 BYTE TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.946 TDSCDMARSSIThresh Struct Reference

Data Fields

- BYTE TDSCDMARSSIThreshListLen
- ULONG * pTDSCDMARSSIThreshList

8.946.1 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.

8.946.2 Field Documentation

8.946.2.1 ULONG* TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.946.2.2 BYTE TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.947 tdscdmaSigInfoExt Struct Reference

Data Fields

- float rssi
- float rscp
- float ecio
- float sinr

8.947.1 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

8.947.2 Field Documentation

8.947.2.1 float tdscdmaSigInfoExt::ecio

8.947.2.2 float tdscdmaSigInfoExt::rscp

8.947.2.3 float tdscdmaSigInfoExt::rssi

8.947.2.4 float tdscdmaSigInfoExt::sinr

8.948 TDSCDMASigInfoExt Struct Reference

Data Fields

- FLOAT rssi
- FLOAT rscp
- FLOAT ecio
- FLOAT sinr

8.948.1 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

8.948.2 Field Documentation

8.948.2.1 FLOAT TDSCDMASigInfoExt::ecio

8.948.2.2 FLOAT TDSCDMASigInfoExt::rscp

8.948.2.3 FLOAT TDSCDMASigInfoExt::rssi

8.948.2.4 FLOAT TDSCDMASigInfoExt::sinr

8.949 TDSCDMASINRCONFThresh Struct Reference

Data Fields

- BYTE TDSCDMASINRCONFThreshListLen
- FLOAT * pTDSCDMASINRCONFThreshList

8.949.1 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

8.949.2 Field Documentation

8.949.2.1 FLOAT* TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

8.949.2.2 BYTE TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

8.950 TDSCDMASINRThresh Struct Reference

Data Fields

- BYTE TDSCDMASINRThreshListLen
- ULONG * pTDSCDMASINRThreshList

8.950.1 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

8.950.2 Field Documentation

8.950.2.1 ULONG* TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.950.2.2 BYTE TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.951 tempData_t Struct Reference

Data Fields

- uint32_t timeSource
- uint32_t timeOfFirstSample
- uint8_t temperatureDataLen
- uint16_t timeOffset [64]
- uint32_t temperature [64]

8.951.1 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
timeOfFirstSample	• 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
temperatureDataLen	• 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

8.951.2 Field Documentation

8.951.2.1 uint32_t tempData_t::temperature[64]

8.951.2.2 uint8_t tempData_t::temperatureDataLen

8.951.2.3 uint32_t tempData_t::timeOfFirstSample

8.951.2.4 uint16_t tempData_t::timeOffset[64]

8.951.2.5 uint32_t tempData_t::timeSource

8.952 tempratureData Struct Reference

Data Fields

- ULONG timeSource
- ULONG timeOfFirstSample
- BYTE temperatureDataLen
- WORD timeOffset [64]
- ULONG temperature [64]

8.952.1 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

8.952.2 Field Documentation

8.952.2.1 ULONG tempratureData::temperature[64]

8.952.2.2 BYTE tempratureData::temperatureDataLen

8.952.2.3 ULONG tempratureData::timeOfFirstSample

8.952.2.4 WORD tempratureData::timeOffset[64]

8.952.2.5 ULONG tempratureData::timeSource

8.953 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

8.953.1 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

8.953.2 Field Documentation

8.953.2.1 WORD TFTIDParams::destPortRangeEnd

8.953.2.2 WORD TFTIDParams::destPortRangeStart

8.953.2.3 BYTE TFTIDParams::eValid

8.953.2.4 BYTE TFTIDParams::filterId

8.953.2.5 ULONG TFTIDParams::flowLabel

8.953.2.6 ULONG TFTIDParams::IPSECSPI

8.953.2.7 BYTE TFTIDParams::ipVersion

8.953.2.8 BYTE TFTIDParams::nextHeader

8.953.2.9 WORD* TFTIDParams::pSourceIP

8.953.2.10 BYTE TFTIDParams::sourceIPMask

8.953.2.11 WORD TFTIDParams::srcPortRangeEnd

8.953.2.12 WORD TFTIDParams::srcPortRangeStart

8.953.2.13 WORD TFTIDParams::tosMask

8.954 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

8.954.1 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.

8.954.2 Field Documentation

8.954.2.1 BYTE timeInfo::day

8.954.2.2 BYTE timeInfo::dayLtSavingAdj

8.954.2.3 BYTE timeInfo::dayOfWeek

8.954.2.4 BYTE timeInfo::hour

8.954.2.5 BYTE timeInfo::minute

8.954.2.6 BYTE timeInfo::month

8.954.2.7 BYTE timeInfo::radioInterface

8.954.2.8 BYTE timeInfo::second

8.954.2.9 INT8 timeInfo::timeZone

8.954.2.10 BYTE timeInfo::TlvPresent

8.954.2.11 WORD timeInfo::year

8.955 TmdDeRegNotMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.955.1 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

8.955.2 Field Documentation

8.955.2.1 CHAR TmdDeRegNotMitigationLvlReq::mitigationDevID[255]

8.955.2.2 BYTE TmdDeRegNotMitigationLvlReq::mitigationDevIDLen

8.956 TmdGetMitigationDevListResp Struct Reference

Data Fields

- BYTE * pMitigationDevListLen
- mitigationDevList * pMitigationDevList

8.956.1 Detailed Description

This structure contains mitigation devices list from the remote endpoint

Parameters

pMitigationDevListLen	• Mitigation Device List Length (Optional) • Number of sets of the following elements • pMitigationDevList
pMitigationDevList	• Mitigation Device List (Optional) • See mitigationDevList for more information.

8.956.2 Field Documentation

8.956.2.1 mitigationDevList* TmdGetMitigationDevListResp::pMitigationDevList

8.956.2.2 BYTE* TmdGetMitigationDevListResp::pMitigationDevListLen

8.957 TmdGetMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.957.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

mitigationDevIDLen	• Number of sets of the following elements – mitigation_dev_id
mitigationDevID	• Mitigation device ID

8.957.2 Field Documentation

8.957.2.1 CHAR TmdGetMitigationLvlReq::mitigationDevID[255]

8.957.2.2 BYTE TmdGetMitigationLvlReq::mitigationDevIDLen

8.958 TmdGetMitigationLvlResp Struct Reference

Data Fields

- BYTE * pCurrentmitigationLvl
- BYTE * pReqMitigationLvl

8.958.1 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.

8.958.2 Field Documentation

8.958.2.1 BYTE* TmdGetMitigationLvlResp::pCurrentmitigationLvl

8.958.2.2 BYTE* TmdGetMitigationLvlResp::pReqMitigationLvl

8.959 TmdMitigationLvlIndReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen
- CHAR mitigationDevID [255]

8.959.1 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

8.959.2 Field Documentation

8.959.2.1 CHAR TmdMitigationLvlIndReq::mitigationDevID[255]

8.959.2.2 BYTE TmdMitigationLvlIndReq::mitigationDevIDLen

8.960 TmdRegNotMitigationLvlReq Struct Reference

Data Fields

- BYTE mitigationDevIDLen

- CHAR mitigationDevID [255]

8.960.1 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

8.960.2 Field Documentation

8.960.2.1 CHAR TmdRegNotMitigationLvlReq::mitigationDevID[255]

8.960.2.2 BYTE TmdRegNotMitigationLvlReq::mitigationDevIDLen

8.961 tokenBucket Struct Reference

Data Fields

- ULONG peakRate
- ULONG tokenRate
- ULONG bucketSz

8.961.1 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

8.961.2 Field Documentation

8.961.2.1 ULONG tokenBucket::bucketSz

8.961.2.2 ULONG tokenBucket::peakRate

8.961.2.3 ULONG tokenBucket::tokenRate

8.962 Tos Struct Reference

Data Fields

- BYTE val
- BYTE mask

8.962.1 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.

8.962.2 Field Documentation

8.962.2.1 BYTE Tos::mask

8.962.2.2 BYTE Tos::val

8.963 transferRouteMessageTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- sMSTransferRouteMTMessageInfo TransferRouteMTMessageInfo

8.963.1 Detailed Description

Parameters

TlvPresent	• Boolean indicating the presence of the TLV in the QMI response
TransferRouteMTMessageInfo	• Transfer Route MT Message • See sMSTransferRouteMTMessageInfo for more information

8.963.2 Field Documentation

8.963.2.1 uint8_t transferRouteMessageTlv::TlvPresent

8.963.2.2 sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo

8.964 TransferStatInd Struct Reference

Data Fields

- BYTE StatsPeriod
- ULONG StatsMask

8.964.1 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

8.964.2 Field Documentation

8.964.2.1 ULONG TransferStatInd::StatsMask

8.964.2.2 BYTE TransferStatInd::StatsPeriod

8.965 transferStatInd Struct Reference

Data Fields

- uint8_t StatsPeriod
- uint32_t StatsMask

8.965.1 Detailed Description

This structure contains the transfer statistic indication information

Parameters

StatsPeriod	Field Period between transfer statistic reports.
StatsMask	requested statistic bit mask.

8.965.2 Field Documentation

8.965.2.1 uint32_t transferStatInd::StatsMask

8.965.2.2 uint8_t transferStatInd::StatsPeriod

8.966 TransferStatsDataType Struct Reference

Data Fields

- BYTE interval

8.966.1 Field Documentation

8.966.1.1 BYTE TransferStatsDataType::interval

8.967 TrStatInd Struct Reference

Data Fields

- BYTE statsPeriod
- ULONG statsMask

8.967.1 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 SLQSWdsEventReportCallBack. • All unlisted bits are reserved for future use and must be set to zero.

8.967.2 Field Documentation

8.967.2.1 ULONG TrStatInd::statsMask

8.967.2.2 BYTE TrStatInd::statsPeriod

8.968 trueIMSI Struct Reference

Data Fields

- BYTE mccT [3]
- WORD imsiT1112
- BYTE imsiTS1 [7]
- BYTE imsiTS2 [3]
- BYTE imsiTaddrNum

8.968.1 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

8.968.2 Field Documentation

8.968.2.1 WORD trueIMSI::imsiT1112

8.968.2.2 BYTE trueIMSI::imsiTaddrNum

8.968.2.3 BYTE trueIMSI::imsiTS1[7]

8.968.2.4 BYTE trueIMSI::imsiTS2[3]

8.968.2.5 BYTE trueIMSI::mccT[3]

8.969 TXAGCList Struct Reference

Data Fields

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

8.969.1 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

8.969.2 Field Documentation

8.969.2.1 WORD* TXAGCList::pTXAIG

8.969.2.2 WORD* TXAGCList::pTXComprSlope

8.969.2.3 WORD* TXAGCList::pTXComprThres

8.969.2.4 WORD* TXAGCList::pTXExpSlope

8.969.2.5 WORD* TXAGCList::pTXExpThres

8.969.2.6 WORD* TXAGCList::pTXStaticGain

8.970 txInfo Struct Reference

Data Fields

- BYTE isInTraffic
- INT32 txPower

8.970.1 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.

8.970.2 Field Documentation

8.970.2.1 BYTE txInfo::isInTraffic

8.970.2.2 INT32 txInfo::txPower

8.971 TXPCMIIRFltr Struct Reference

Data Fields

- WORD * pFlag
- WORD * pStageCnt
- BYTE * pStage0Val
- BYTE * pStage1Val
- BYTE * pStage2Val
- BYTE * pStage3Val
- BYTE * pStage4Val

8.971.1 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

8.971.2 Field Documentation

8.971.2.1 WORD* TXPCMIIRFItr::pFlag

8.971.2.2 BYTE* TXPCMIIRFItr::pStage0Val

8.971.2.3 BYTE* TXPCMIIRFItr::pStage1Val

8.971.2.4 BYTE* TXPCMIIRFItr::pStage2Val

8.971.2.5 BYTE* TXPCMIIRFItr::pStage3Val

8.971.2.6 BYTE* TXPCMIIRFItr::pStage4Val

8.971.2.7 WORD* TXPCMIIRFItr::pStageCnt

8.972 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

8.972.1 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.

8.972.2 Field Documentation

8.972.2.1 uint8_t uim_appStatus::aidLength

8.972.2.2 uint8_t uim_appStatus::aidVal[255]

8.972.2.3 uint8_t uim_appStatus::appState

8.972.2.4 uint8_t uim_appStatus::appType

8.972.2.5 uint8_t uim_appStatus::persoFeature

8.972.2.6 uint8_t uim_appStatus::persoRetries

8.972.2.7 uint8_t uim_appStatus::persoState

8.972.2.8 uint8_t uim_appStatus::persoUnblockRetries

8.972.2.9 uint8_t uim_appStatus::pin1Retries

8.972.2.10 uint8_t uim_appStatus::pin1State

8.972.2.11 uint8_t uim_appStatus::pin2Retries

8.972.2.12 uint8_t uim_appStatus::pin2State

8.972.2.13 uint8_t uim_appStatus::puk1Retries

8.972.2.14 uint8_t uim_appStatus::puk2Retries

8.972.2.15 uint8_t uim_appStatus::univPin

8.973 uim_authenticateResult Struct Reference

Data Fields

- uint16_t contentLen

- uint8_t content [1024]

8.973.1 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.

8.973.2 Field Documentation

8.973.2.1 uint8_t uim_authenticateResult::content[1024]

8.973.2.2 uint16_t uim_authenticateResult::contentLen

8.974 uim_authenticationData Struct Reference

Data Fields

- uint8_t context
- uint16_t dataLen
- uint8_t data [1024]

8.974.1 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.

8.974.2 Field Documentation

8.974.2.1 uint8_t uim_authenticationData::context

8.974.2.2 uint8_t uim_authenticationData::data[1024]

8.974.2.3 uint16_t uim_authenticationData::dataLen

8.975 uim_cardResult Struct Reference

Data Fields

- uint8_t sw1
- uint8_t sw2

8.975.1 Detailed Description

This structure contains the information about the card result.

Parameters

sw1	• SW1 received from the card.
sw2	• SW2 received from the card.

8.975.2 Field Documentation

8.975.2.1 uint8_t uim_cardResult::sw1

8.975.2.2 uint8_t uim_cardResult::sw2

8.976 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]

8.976.1 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.

8.976.2 Field Documentation

8.976.2.1 uint16_t uim_cardStatus::index1xPri

8.976.2.2 uint16_t uim_cardStatus::index1xSec

8.976.2.3 uint16_t uim_cardStatus::indexGwPri

8.976.2.4 uint16_t uim_cardStatus::indexGwSec

8.976.2.5 uint8_t uim_cardStatus::numSlot

8.976.2.6 uim_slotInfo uim_cardStatus::SlotInfo[5]

8.977 uim_changeUIMPIN Struct Reference

Data Fields

- uint8_t pinID
- uint8_t oldPINLen
- uint8_t oldPINVal [255]
- uint8_t pinLen
- uint8_t pinVal [255]

8.977.1 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.

8.977.2 Field Documentation

8.977.2.1 uint8_t uim_changeUIMPIN::oldPINLen

8.977.2.2 uint8_t uim_changeUIMPIN::oldPINVal[255]

8.977.2.3 uint8_t uim_changeUIMPIN::pinID

8.977.2.4 uint8_t uim_changeUIMPIN::pinLen

8.977.2.5 uint8_t uim_changeUIMPIN::pinVal[255]

8.978 uim_depersonalizationInformation Struct Reference

Data Fields

- uint8_t feature
- uint8_t operation
- uint8_t ckLen
- uint8_t ckVal [255]

8.978.1 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.

8.978.2 Field Documentation

8.978.2.1 uint8_t uim_depersonalizationInformation::ckLen

8.978.2.2 uint8_t uim_depersonalizationInformation::ckVal[255]

8.978.2.3 uint8_t uim_depersonalizationInformation::feature

8.978.2.4 uint8_t uim_depersonalizationInformation::operation

8.979 uim_encryptedPIN1 Struct Reference

Data Fields

- uint8_t pin1Len
- uint8_t pin1Val [255]

8.979.1 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.

8.979.2 Field Documentation

8.979.2.1 uint8_t uim_encryptedPIN1::pin1Len

8.979.2.2 uint8_t uim_encryptedPIN1::pin1Val[255]

8.980 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]

8.980.1 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.

8.980.2 Field Documentation

8.980.2.1 uint16_t uim_fileAttributes::fileID

8.980.2.2 uint16_t uim_fileAttributes::fileSize

8.980.2.3 uint8_t uim_fileAttributes::fileType

8.980.2.4 uint16_t uim_fileAttributes::rawLen

8.980.2.5 uint8_t uim_fileAttributes::rawValue[255]

8.980.2.6 uint16_t uim_fileAttributes::recordCount

8.980.2.7 uint16_t uim_fileAttributes::recordSize

8.980.2.8 uint8_t uim_fileAttributes::secActivate

8.980.2.9 uint16_t uim_fileAttributes::secActivateMask

8.980.2.10 uint8_t uim_fileAttributes::secDeactivate

8.980.2.11 uint16_t uim_fileAttributes::secDeactivateMask

8.980.2.12 uint8_t uim_fileAttributes::secIncrease

8.980.2.13 uint16_t uim_fileAttributes::secIncreaseMask

8.980.2.14 uint8_t uim_fileAttributes::secRead

8.980.2.15 uint16_t uim_fileAttributes::secReadMask

8.980.2.16 uint8_t uim_fileAttributes::secWrite

8.980.2.17 uint16_t uim_fileAttributes::secWriteMask

8.981 uim_fileInfo Struct Reference

Data Fields

- uint16_t fileID
- uint8_t pathLen
- uint16_t path [255]

8.981.1 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).

8.981.2 Field Documentation

8.981.2.1 uint16_t uim_fileInfo::fileID

8.981.2.2 uint16_t uim_fileInfo::path[255]

8.981.2.3 uint8_t uim_fileInfo::pathLen

8.982 uim_hotSwapStatus Struct Reference

Data Fields

- uint8_t hotSwapLength
- uint8_t hotSwap [255]

8.982.1 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

8.982.2 Field Documentation

8.982.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

8.982.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

8.983 uim_personalizationStatus Struct Reference

Data Fields

- uint8_t numFeatures
- uint8_t feature [12]
- uint8_t verifyLeft [12]
- uint8_t unblockLeft [12]

8.983.1 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.

8.983.2 Field Documentation

8.983.2.1 uint8_t uim_personalizationStatus::feature[12]

8.983.2.2 uint8_t uim_personalizationStatus::numFeatures

8.983.2.3 uint8_t uim_personalizationStatus::unblockLeft[12]

8.983.2.4 uint8_t uim_personalizationStatus::verifyLeft[12]

8.984 uim_readResult Struct Reference

Data Fields

- uint16_t contentLen
- uint8_t content [255]

8.984.1 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.

8.984.2 Field Documentation

8.984.2.1 uint8_t uim_readResult::content[255]

8.984.2.2 uint16_t uim_readResult::contentLen

8.985 uim_readTransparentInfo Struct Reference

Data Fields

- uint16_t offset
- uint16_t length

8.985.1 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.

8.985.2 Field Documentation

8.985.2.1 uint16_t uim_readTransparentInfo::length

8.985.2.2 uint16_t uim_readTransparentInfo::offset

8.986 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]

8.986.1 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

8.986.2 Field Documentation

8.986.2.1 uint8_t uim_refreshevent::aid[255]

8.986.2.2 uint8_t uim_refreshevent::aidLength

8.986.2.3 uim_fileInfo uim_refreshevent::arrfileInfo[255]

8.986.2.4 uint8_t uim_refreshevent::mode

8.986.2.5 uint16_t uim_refreshevent::numOfFiles

8.986.2.6 uint8_t uim_refreshevent::sessionType

8.986.2.7 uint8_t uim_refreshevent::stage

8.987 uim_registerRefresh Struct Reference

Data Fields

- uint8_t registerFlag
- uint8_t voteForInit
- uint16_t numFiles
- uim_fileInfo arrfileInfo [255]

8.987.1 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

8.987.2 Field Documentation

8.987.2.1 uim_fileInfo uim_registerRefresh::arrfileInfo[255]

8.987.2.2 uint16_t uim_registerRefresh::numFiles

8.987.2.3 uint8_t uim_registerRefresh::registerFlag

8.987.2.4 uint8_t uim_registerRefresh::voteForInit

8.988 uim_remainingRetries Struct Reference

Data Fields

- uint8_t verifyLeft
- uint8_t unblockLeft

8.988.1 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.

8.988.2 Field Documentation

8.988.2.1 uint8_t uim_remainingRetries::unblockLeft

8.988.2.2 uint8_t uim_remainingRetries::verifyLeft

8.989 uim_sessionInformation Struct Reference

Data Fields

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

8.989.1 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.

8.989.2 Field Documentation

8.989.2.1 uint8_t uim_sessionInformation::aid[255]

8.989.2.2 uint8_t uim_sessionInformation::aidLength

8.989.2.3 uint8_t uim_sessionInformation::sessionType

8.990 uim_setPINProtection Struct Reference

Data Fields

- uint8_t pinID
- uint8_t pinOperation

- uint8_t pinLength
- uint8_t pinValue [255]

8.990.1 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.

8.990.2 Field Documentation

8.990.2.1 uint8_t uim_setPINProtection::pinID

8.990.2.2 uint8_t uim_setPINProtection::pinLength

8.990.2.3 uint8_t uim_setPINProtection::pinOperation

8.990.2.4 uint8_t uim_setPINProtection::pinValue[255]

8.991 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]

8.991.1 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.

8.991.2 Field Documentation

8.991.2.1 uim_appStatus uim_slotInfo::AppStatus[10]

8.991.2.2 uint8_t uim_slotInfo::cardState

8.991.2.3 uint8_t uim_slotInfo::errorState

8.991.2.4 uint8_t uim_slotInfo::numApp

8.991.2.5 uint8_t uim_slotInfo::upinRetries

8.991.2.6 uint8_t uim_slotInfo::upinState

8.991.2.7 uint8_t uim_slotInfo::upukRetries

8.992 uim_UIMSessionInformation Struct Reference

Data Fields

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

8.992.1 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.

8.992.2 Field Documentation

8.992.2.1 uint8_t uim_UIMSessionInformation::aid[255]

8.992.2.2 uint8_t uim_UIMSessionInformation::aidLength

8.992.2.3 uint8_t uim_UIMSessionInformation::sessionType

8.993 uim_unblockUIMPIN Struct Reference

Data Fields

- uint8_t pinID
- uint8_t pukLen
- uint8_t pukVal [255]
- uint8_t newPINLen
- uint8_t newPINVal [255]

8.993.1 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.

8.993.2 Field Documentation

8.993.2.1 uint8_t uim_unblockUIMPIN::newPINLen

8.993.2.2 uint8_t uim_unblockUIMPIN::newPINVal[255]

8.993.2.3 uint8_t uim_unblockUIMPIN::pinID

8.993.2.4 uint8_t uim_unblockUIMPIN::pukLen

8.993.2.5 uint8_t uim_unblockUIMPIN::pukVal[255]

8.994 uim_verifyUIMPIN Struct Reference

Data Fields

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

8.994.1 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.

8.994.2 Field Documentation

8.994.2.1 uint8_t uim_verifyUIMPIN::pinID

8.994.2.2 uint8_t uim_verifyUIMPIN::pinLen

8.994.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

8.995 UIMAuthenticateReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- authenticationData authData
- ULONG * pIndicationToken

8.995.1 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.

8.995.2 Field Documentation

8.995.2.1 authenticationData UIMAuthenticateReq::authData

8.995.2.2 ULONG* UIMAuthenticateReq::pIndicationToken

8.995.2.3 UIMSessionInformation UIMAuthenticateReq::sessionInfo

8.996 UIMAuthenticateResp Struct Reference

Data Fields

- cardResult * pCardResult
- authenticateResult * pAuthenticateResult
- ULONG * pIndicationToken

8.996.1 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.

8.996.2 Field Documentation

8.996.2.1 authenticateResult* UIMAuthenticateResp::pAuthenticateResult

8.996.2.2 cardResult* UIMAuthenticateResp::pCardResult

8.996.2.3 ULONG* UIMAuthenticateResp::pIndicationToken

8.997 UIMChangePinReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- changeUIMPIN changePIN
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.997.1 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.

8.997.2 Field Documentation

8.997.2.1 changeUIMPIN UIMChangePinReq::changePIN

8.997.2.2 ULONG* UIMChangePinReq::pIndicationToken

8.997.2.3 BYTE* UIMChangePinReq::pKeyReferenceID

8.997.2.4 UIMSessionInformation UIMChangePinReq::sessionInfo

8.998 UIMDepersonalizationReq Struct Reference

Data Fields

- depersonalizationInformation depersonilisationInfo

8.998.1 Detailed Description

This structure contains information of the request parameters associated with a Depersonalization API.

Parameters

depersonilisation-Info	• See depersonalizationInformation for more information.

8.998.2 Field Documentation

8.998.2.1 depersonalizationInformation UIMDepersonalizationReq::depersonilisationInfo

8.999 UIMDepersonalizationResp Struct Reference

Data Fields

- remainingRetries * pRemainingRetries

8.999.1 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.

8.999.2 Field Documentation

8.999.2.1 remainingRetries* UIMDepersonalizationResp::pRemainingRetries

8.1000 UIMEventRegisterReqResp Struct Reference

Data Fields

- ULONG eventMask

8.1000.1 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

8.1000.2 Field Documentation

8.1000.2.1 ULONG UIMEventRegisterReqResp::eventMask

8.1001 UIMGetCardStatusResp Struct Reference

Data Fields

- cardStatus * pCardStatus
- hotSwapStatus * pHotSwapStatus

8.1001.1 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.

8.1001.2 Field Documentation

8.1001.2.1 cardStatus* UIMGetCardStatusResp::pCardStatus

8.1001.2.2 hotSwapStatus* UIMGetCardStatusResp::pHotSwapStatus

8.1002 UIMGetConfigurationReq Struct Reference

Data Fields

- ULONG * pConfigurationMask

8.1002.1 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.

8.1002.2 Field Documentation

8.1002.2.1 ULONG* UIMGetConfigurationReq::pConfigurationMask

8.1003 UIMGetConfigurationResp Struct Reference

Data Fields

- BYTE * pAutoSelection
- personalizationStatus * pPersonalizationStatus
- BYTE * pHaltSubscription

8.1003.1 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

8.1003.2 Field Documentation

8.1003.2.1 BYTE* UIMGetConfigurationResp::pAutoSelection

8.1003.2.2 BYTE* UIMGetConfigurationResp::pHaltSubscription

8.1003.2.3 personalizationStatus* UIMGetConfigurationResp::pPersonalizationStatus

8.1004 UIMGetFileAttributesReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- ULONG * pIndicationToken

8.1004.1 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.

8.1004.2 Field Documentation

8.1004.2.1 fileInfo UIMGetFileAttributesReq::fileIndex

8.1004.2.2 ULONG* UIMGetFileAttributesReq::pIndicationToken

8.1004.2.3 UIMSessionInformation UIMGetFileAttributesReq::sessionInfo

8.1005 UIMGetFileAttributesResp Struct Reference

Data Fields

- cardResult * pCardResult
- fileAttributes * pFileAttributes
- ULONG * pIndicationToken

8.1005.1 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.

8.1005.2 Field Documentation

8.1005.2.1 cardResult∗ UIMGetFileAttributesResp::pCardResult

8.1005.2.2 fileAttributes∗ UIMGetFileAttributesResp::pFileAttributes

8.1005.2.3 ULONG∗ UIMGetFileAttributesResp::pIndicationToken

8.1006 UIMGetSlotsStatusResp Struct Reference

Data Fields

- BYTE ∗ pNumberOfPhySlot
- UIMSlotsStatus ∗ pUimSlotsStatus

8.1006.1 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..

8.1006.2 Field Documentation

8.1006.2.1 BYTE∗ UIMGetSlotsStatusResp::pNumberOfPhySlot

8.1006.2.2 UIMSlotsStatus* UIMGetSlotsStatusResp::pUimSlotsStatus

8.1007 UIMPinResp Struct Reference

Data Fields

- remainingRetries * pRemainingRetries
- encryptedPIN1 * pEncryptedPIN1
- ULONG * pIndicationToken

8.1007.1 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.

8.1007.2 Field Documentation

8.1007.2.1 encryptedPIN1* UIMPinResp::pEncryptedPIN1

8.1007.2.2 ULONG* UIMPinResp::pIndicationToken

8.1007.2.3 remainingRetries* UIMPinResp::pRemainingRetries

8.1008 UIMPowerDownReq Struct Reference

Data Fields

- BYTE slot

8.1008.1 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

8.1008.2 Field Documentation

8.1008.2.1 BYTE UIMPowerDownReq::slot

8.1009 UIMPowerUpReq Struct Reference

Data Fields

- BYTE slot
- BYTE * pIgnoreHotSwapSwitch

8.1009.1 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

8.1009.2 Field Documentation

8.1009.2.1 BYTE* UIMPowerUpReq::pIgnoreHotSwapSwitch

8.1009.2.2 BYTE UIMPowerUpReq::slot

8.1010 UIMReadTransparentReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- readTransparentInfo readTransparent
- ULONG * pIndicationToken
- BYTE * pEncryptData

8.1010.1 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.

8.1010.2 Field Documentation

8.1010.2.1 fileInfo UIMReadTransparentReq::fileIndex

8.1010.2.2 BYTE* UIMReadTransparentReq::pEncryptData

8.1010.2.3 ULONG* UIMReadTransparentReq::pIndicationToken

8.1010.2.4 readTransparentInfo UIMReadTransparentReq::readTransparent

8.1010.2.5 UIMSessionInformation UIMReadTransparentReq::sessionInfo

8.1011 UIMReadTransparentResp Struct Reference

Data Fields

- cardResult * pCardResult
- readResult * pReadResult
- ULONG * pIndicationToken
- BYTE * pEncryptedData

8.1011.1 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.

8.1011.2 Field Documentation

8.1011.2.1 cardResult* UIMReadTransparentResp::pCardResult

8.1011.2.2 BYTE* UIMReadTransparentResp::pEncryptedData

8.1011.2.3 ULONG* UIMReadTransparentResp::pIndicationToken

8.1011.2.4 readResult* UIMReadTransparentResp::pReadResult

8.1012 UIMRefreshCompleteReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- BYTE refreshComplete

8.1012.1 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

8.1012.2 Field Documentation

8.1012.2.1 BYTE UIMRefreshCompleteReq::refreshComplete

8.1012.2.2 UIMSessionInformation UIMRefreshCompleteReq::sessionInfo

8.1013 UIMRefreshEvent Struct Reference

Data Fields

- BYTE stage
- BYTE mode
- BYTE sessionType
- BYTE aidLength
- BYTE aid [255]
- WORD numOfFiles
- fileInfo arrfileInfo [255]

8.1013.1 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

8.1013.2 Field Documentation

8.1013.2.1 BYTE UIMRefreshEvent::aid[255]

8.1013.2.2 BYTE UIMRefreshEvent::aidLength

8.1013.2.3 fileInfo UIMRefreshEvent::arrfileInfo[255]

8.1013.2.4 BYTE UIMRefreshEvent::mode

8.1013.2.5 WORD UIMRefreshEvent::numOfFiles

8.1013.2.6 BYTE UIMRefreshEvent::sessionType

8.1013.2.7 BYTE UIMRefreshEvent::stage

8.1014 UIMRefreshGetLastEventReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo

8.1014.1 Detailed Description

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

Parameters

sessionInfo(-Mandatory)	• See UIMSessionInformation for more information.

8.1014.2 Field Documentation

8.1014.2.1 UIMSessionInformation UIMRefreshGetLastEventReq::sessionInfo

8.1015 UIMRefreshGetLastEventResp Struct Reference

Data Fields

- UIMRefreshEvent * pRefreshEvent

8.1015.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMRefreshGetLastEvent.

Parameters

refreshEvent(-Optional)	• See UIMRefreshEvent for more information.

8.1015.2 Field Documentation

8.1015.2.1 UIMRefreshEvent* UIMRefreshGetLastEventResp::pRefreshEvent

8.1016 UIMRefreshOKReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- BYTE OKtoRefresh

8.1016.1 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

8.1016.2 Field Documentation

8.1016.2.1 BYTE UIMRefreshOKReq::OKtoRefresh

8.1016.2.2 UIMSessionInformation UIMRefreshOKReq::sessionInfo

8.1017 UIMRefreshRegisterReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- registerRefresh regRefresh

8.1017.1 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

8.1017.2 Field Documentation

8.1017.2.1 registerRefresh UIMRefreshRegisterReq::regRefresh

8.1017.2.2 UIMSessionInformation UIMRefreshRegisterReq::sessionInfo

8.1018 UIMSessionInformation Struct Reference

Data Fields

- BYTE sessionType
- BYTE aidLength
- BYTE aid [255]

8.1018.1 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.

8.1018.2 Field Documentation

8.1018.2.1 BYTE UIMSessionInformation::aid[255]

8.1018.2.2 BYTE UIMSessionInformation::aidLength

8.1018.2.3 BYTE UIMSessionInformation::sessionType

8.1019 UIMSetPinProtectionReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- setPINProtection pinProtection
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.1019.1 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.

8.1019.2 Field Documentation

8.1019.2.1 ULONG* UIMSetPinProtectionReq::pIndicationToken

8.1019.2.2 setPINProtection UIMSetPinProtectionReq::pinProtection

8.1019.2.3 BYTE* UIMSetPinProtectionReq::pKeyReferenceID

8.1019.2.4 UIMSessionInformation UIMSetPinProtectionReq::sessionInfo

8.1020 UIMSlotsStatus Struct Reference

Data Fields

- UIMSlotStatus uimSlotStatus [255]

8.1020.1 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.

8.1020.2 Field Documentation

8.1020.2.1 UIMSlotStatus UIMSlotsStatus::uimSlotStatus[255]

8.1021 UIMSlotStatus Struct Reference

Data Fields

- ULONG uPhyCardStatus
- ULONG uPhySlotStatus
- BYTE bLogicalSlot
- BYTE bICCIDLength
- BYTE bICCID [255]

8.1021.1 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.

8.1021.2 Field Documentation

8.1021.2.1 BYTE UIMSlotStatus::bICCID[255]

8.1021.2.2 BYTE UIMSlotStatus::bICCIDLength

8.1021.2.3 BYTE UIMSlotStatus::bLogicalSlot

8.1021.2.4 ULONG UIMSlotStatus::uPhyCardStatus

8.1021.2.5 ULONG UIMSlotStatus::uPhySlotStatus

8.1022 UIMSlotStatusChangeInfo Struct Reference

Data Fields

- UIMSlotsStatus slotsstatusChange
- BYTE bNumberOfPhySlots

8.1022.1 Detailed Description

Structure consist of cardstatus params

Parameters

slotstatus-Change	• See UIMSlotStatus for more information
bNumberOfPhy-Slots	• Number of Physical Slot(s)

8.1022.2 Field Documentation

8.1022.2.1 BYTE UIMSlotStatusChangeInfo::bNumberOfPhySlots

8.1022.2.2 UIMSlotsStatus UIMSlotStatusChangeInfo::slotsstatusChange

8.1023 UIMStatusChangeInfo Struct Reference

Data Fields

- cardStatus statusChange

8.1023.1 Detailed Description

Structure consist of cardstatus params

Parameters

statusChange	• See cardStatus for more information

8.1023.2 Field Documentation

8.1023.2.1 cardStatus UIMStatusChangeInfo::statusChange

8.1024 UIMSwitchSlotReq Struct Reference

Data Fields

- BYTE bLogicalSlot
- ULONG ulPhysicalSlot

8.1024.1 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

8.1024.2 Field Documentation

8.1024.2.1 BYTE UIMSwitchSlotReq::bLogicalSlot

8.1024.2.2 ULONG UIMSwitchSlotReq::ulPhysicalSlot

8.1025 UIMUnblockPinReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- unblockUIMPIN unblockPIN
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.1025.1 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.

8.1025.2 Field Documentation

8.1025.2.1 ULONG* UIMUnblockPinReq::pIndicationToken

8.1025.2.2 BYTE* UIMUnblockPinReq::pKeyReferenceID

8.1025.2.3 UIMSessionInformation UIMUnblockPinReq::sessionInfo

8.1025.2.4 unblockUIMPIN UIMUnblockPinReq::unblockPIN

8.1026 UIMVerifyPinReq Struct Reference

Data Fields

- UIMSessionInformation sessionInfo
- verifyUIMPIN verifyPIN
- encryptedPIN1 * pEncryptedPIN1
- BYTE * pKeyReferenceID
- ULONG * pIndicationToken

8.1026.1 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.

8.1026.2 Field Documentation

8.1026.2.1 encryptedPIN1∗ UIMVerifyPinReq::pEncryptedPIN1

8.1026.2.2 ULONG∗ UIMVerifyPinReq::pIndicationToken

8.1026.2.3 BYTE∗ UIMVerifyPinReq::pKeyReferenceID

8.1026.2.4 UIMSessionInformation UIMVerifyPinReq::sessionInfo

8.1026.2.5 verifyUIMPIN UIMVerifyPinReq::verifyPIN

8.1027 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]

8.1027.1 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.

8.1027.2 Field Documentation

8.1027.2.1 WORD UMTSInfo::cellID

8.1027.2.2 SHORT UMTSInfo::ecio

8.1027.2.3 BYTE UMTSInfo::geranInst

8.1027.2.4 geranInstInfo UMTSInfo::GeranInstInfo[255]

8.1027.2.5 WORD UMTSInfo::lac

8.1027.2.6 BYTE UMTSInfo::plmn[3]

8.1027.2.7 WORD UMTSInfo::psc

8.1027.2.8 SHORT UMTSInfo::rscp

8.1027.2.9 WORD UMTSInfo::uarfcn

8.1027.2.10 BYTE UMTSInfo::umtsInst

8.1027.2.11 UMTSinstInfo UMTSInfo::UMTSInstInfo[255]

8.1028 UMTSinstInfo Struct Reference

Data Fields

- WORD umtsUarfcn
- WORD umtsPsc
- SHORT umtsRscp
- SHORT umtsEcio

8.1028.1 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).

8.1028.2 Field Documentation

8.1028.2.1 SHORT UMTSinstInfo::umtsEcio

8.1028.2.2 WORD UMTSinstInfo::umtsPsc

8.1028.2.3 SHORT UMTSinstInfo::umtsRscp

8.1028.2.4 WORD UMTSinstInfo::umtsUarfcn

8.1029 umtsLTENbrCell Struct Reference

Data Fields

- WORD earfcn
- WORD pci
- ULONG rsrp
- ULONG rsrq
- SHORT srxlev
- BYTE cellIsTDD

8.1029.1 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.

8.1029.2 Field Documentation

8.1029.2.1 BYTE umtsLTENbrCell::cellIsTDD

8.1029.2.2 WORD umtsLTENbrCell::earfcn

8.1029.2.3 WORD umtsLTENbrCell::pci

8.1029.2.4 ULONG umtsLTENbrCell::rsrp

8.1029.2.5 ULONG umtsLTENbrCell::rsrq

8.1029.2.6 SHORT umtsLTENbrCell::srxlev

8.1030 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

8.1030.1 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.

8.1030.2 Field Documentation

8.1030.2.1 BYTE UMTSMinQoS::deliveryErrSDU

8.1030.2.2 ULONG UMTSMinQoS::grntDownlinkBitrate

8.1030.2.3 ULONG UMTSMinQoS::grntUplinkBitrate

8.1030.2.4 ULONG UMTSMinQoS::maxDownlinkBitrate

8.1030.2.5 ULONG UMTSMinQoS::maxSDUSize

8.1030.2.6 ULONG UMTSMinQoS::maxUplinkBitrate

8.1030.2.7 BYTE UMTSMinQoS::qosDeliveryOrder

8.1030.2.8 BYTE UMTSMinQoS::resBerRatio

8.1030.2.9 BYTE UMTSMinQoS::sduErrorRatio

8.1030.2.10 BYTE UMTSMinQoS::trafficClass

8.1030.2.11 ULONG UMTSMinQoS::trafficPriority

8.1030.2.12 ULONG UMTSMinQoS::transferDelay

8.1031 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

8.1031.1 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.

8.1031.2 Field Documentation

8.1031.2.1 BYTE UMTSQoS::deliveryErrSDU

8.1031.2.2 ULONG UMTSQoS::grntDownlinkBitrate

8.1031.2.3 ULONG UMTSQoS::grntUplinkBitrate

8.1031.2.4 ULONG UMTSQoS::maxDownlinkBitrate

8.1031.2.5 ULONG UMTSQoS::maxSDUSize

8.1031.2.6 ULONG UMTSQoS::maxUplinkBitrate

8.1031.2.7 BYTE UMTSQoS::qosDeliveryOrder

8.1031.2.8 BYTE UMTSQoS::resBerRatio

8.1031.2.9 BYTE UMTSQoS::sduErrorRatio

8.1031.2.10 BYTE UMTSQoS::trafficClass

8.1031.2.11 ULONG UMTSQoS::trafficPriority

8.1031.2.12 ULONG UMTSQoS::transferDelay

8.1032 UMTSReqQoSSigInd Struct Reference

Data Fields

- struct UMTSQoS UMTSReqQoS
- BYTE SigInd

8.1032.1 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

8.1032.2 Field Documentation

8.1032.2.1 BYTE UMTSReqQoSSigInd::SigInd

8.1032.2.2 struct UMTSQoS UMTSReqQoSSigInd::UMTSReqQoS

8.1033 unblockUIMPIN Struct Reference

Data Fields

- BYTE pinID
- BYTE pukLen
- BYTE pukVal [255]
- BYTE newPINLen
- BYTE newPINVal [255]

8.1033.1 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.

8.1033.2 Field Documentation

8.1033.2.1 BYTE unblockUIMPIN::newPINLen

8.1033.2.2 BYTE unblockUIMPIN::newPINVal[255]

8.1033.2.3 BYTE unblockUIMPIN::pinID

8.1033.2.4 BYTE unblockUIMPIN::pukLen

8.1033.2.5 BYTE unblockUIMPIN::pukVal[255]

8.1034 UniversalTime Struct Reference

Data Fields

- WORD year
- BYTE month
- BYTE day
- BYTE hour
- BYTE minute
- BYTE second
- BYTE dayOfWeek

8.1034.1 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.

8.1034.2 Field Documentation

8.1034.2.1 BYTE UniversalTime::day

8.1034.2.2 BYTE UniversalTime::dayOfWeek

8.1034.2.3 BYTE UniversalTime::hour

8.1034.2.4 BYTE UniversalTime::minute

8.1034.2.5 BYTE UniversalTime::month

8.1034.2.6 BYTE UniversalTime::second

8.1034.2.7 WORD UniversalTime::year

8.1035 unpack_dms_GetActivationState_t Struct Reference

Data Fields

- uint8_t state

8.1035.1 Detailed Description

This structure contains unpack_dms_GetActivationState parameters.

Parameters

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

8.1035.2 Field Documentation

8.1035.2.1 uint8_t unpack_dms_GetActivationState_t::state

8.1036 unpack_dms_GetBandCapability_t Struct Reference

Data Fields

- uint32_t BandCapability
- uint16_t Tlvresult

8.1036.1 Detailed Description

Structure to store the band capability of the device.

Parameters

BandCapability	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
Tlvresult	• unpack Tlv Result.

8.1036.2 Field Documentation

8.1036.2.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

8.1036.2.2 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

8.1037 unpack_dms_GetCrashAction_t Struct Reference

Data Fields

- uint8_t DevCrashState
- uint16_t Tlvresult

8.1037.1 Detailed Description

This structure is used to store the Crash State from the device.

Parameters

DevCrashState	• 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
Tlvresult	• unpack Tlv result

8.1037.2 Field Documentation

8.1037.2.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

8.1037.2.2 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

8.1038 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

8.1038.1 Detailed Description

This structure contains current settings of custom features

Parameters

GpsEnable	• describes if GPS is enabled or disabled • values: – 0x00 - GPS is disabled – 0x01 - GPS is enabled
DisableIMSI	• optional 1 byte parameter • describes if IMSI display is enabled or disabled • values: – 0x00 - Allow display of IMSI – 0x01 - Do not display IMSI
IPFamSupport	• optional 2 byte BitMask • bitmask representing the IP families supported • values: – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - IPv4v6
RMAutoConnect	• optional 1 byte parameter • QMI Mode RM Net Auto Connect Support • values: – 0x00 - Not Supported – 0x01 - Supported
GPSSel	• 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
SMSSupport	• optional 1 byte parameter • SMS support • values: – 0x00 - Not supported – 0x01 - supported • Used to determine whether or not to hide SMS from user

IsVoiceEnabled	• 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
DHCPRelay-Enabled	• optional 1 byte parameter • DHCP Relay support • values: – 0x00 - Disable DHCP relay – 0x01 - Enable DHCP relay
GPSLPM	• optional 1 byte parameter • GPSLPM support • values: – 0x00 - Enable GPS in Low Power Mode – 0x01 - Disable GPS in Low Power Mode

8.1038.2 Field Documentation

8.1038.2.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

8.1038.2.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

8.1038.2.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

8.1038.2.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

8.1038.2.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

8.1038.2.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

8.1038.2.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

8.1038.2.8 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

8.1038.2.9 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

8.1038.2.10 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

8.1039 unpack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

- DMSgetCustomFeatureV2 GetCustomFeatureV2
- uint16_t Tlvresult

8.1039.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

GetCustomFeatureV2	• See DMSgetCustomFeatureV2 for more information
Tlvresult	• Unpack Result

8.1039.2 Field Documentation

8.1039.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

8.1039.2.2 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

8.1040 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

8.1040.1 Detailed Description

Structure to store the device capabilities

Parameters

MaxTXChannelRate	• Maximum transmission rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies
MaxRXChannelRate	• Maximum reception rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies
DataServiceCapability	• 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

SimCapability	Device SIM capability • 0 - SIM not supported • 1 - SIM supported
RadioIfacesSize	• 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
RadioIfaces	• Radio interface array. This is a structure of array containing the elements below. uint8_t RadioIfaces – See qaGobiApiTableRadioInterfaces.h for Radio Interfaces
Tlvresult	• Unpack Result

8.1040.2 Field Documentation

8.1040.2.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

8.1040.2.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

8.1040.2.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

8.1040.2.4 uint8_t unpack_dms_GetDeviceCap_t::RadioIfaces[64]

8.1040.2.5 uint32_t unpack_dms_GetDeviceCap_t::RadioIfacesSize

8.1040.2.6 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

8.1040.2.7 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

8.1041 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]

8.1041.1 Detailed Description

Gets the device capabilities structure

Parameters

maxTxChannel-Rate	• Maximum transmission rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies
maxRxChannel-Rate	• Maximum reception rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies
dataServiceCa-Capability	• 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
simCapability	• Device SIM capability – 0 - SIM not supported – 1 - SIM supported
radioIfacesSize	• 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
RadioIfaces[OU-T]	• Radio interface array. This is a structure of array containing the elements below. uint8_t RadioIfaces – See qaGobiApiTableRadioInterfaces.h for Radio Interfaces

8.1041.2 Field Documentation

8.1041.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability

8.1041.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate

8.1041.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate

8.1041.2.4 uint8_t unpack_dms_GetDeviceCapabilities_t::RadioIfaces[255]

8.1041.2.5 uint32_t unpack_dms_GetDeviceCapabilities_t::radioIfacesSize

8.1041.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

8.1042 unpack_dms_GetDeviceHardwareRev_t Struct Reference

Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult

8.1042.1 Detailed Description

Structure to store the hardware revision of the device

Parameters

stringSize	• The maximum number of characters (including NULL terminator) that the string array can contain
String	• NULL terminated Hardware Revision string • Maximum Length is 255 Bytes
Tlvresult	• Unpack Result

8.1042.2 Field Documentation

8.1042.2.1 char unpack_dms_GetDeviceHardwareRev_t::String[255]

8.1042.2.2 uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize

8.1042.2.3 uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult

8.1043 unpack_dms_GetDeviceMfr_t Struct Reference

Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult

8.1043.1 Detailed Description

This structure is used to store unpack_dms_GetDeviceMfr parameters

Parameters

stringSize	• The maximum number of characters (including NULL terminator) that the string array can contain
String	• NULL terminated Device Manufacture string • Maximum Length is 255 Bytes
Tlvresult	• unpack Tlv result

8.1043.2 Field Documentation

8.1043.2.1 char unpack_dms_GetDeviceMfr_t::String[255]

8.1043.2.2 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

8.1043.2.3 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

8.1044 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

8.1044.1 Detailed Description

This structure used to store unpack_dms_GetDeviceSerialNumbers parameters

Parameters

esnSize	• The maximum number of characters (including NULL terminator) that the ESN string array can contain
ESNString	• NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed • Maximum Length is 255 Bytes
imeiSize	• The maximum number of characters (including NULL terminator) that the IMEI string array can contain
IMEIString	• NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed • Maximum Length is 255 Bytes
meidSize	• The maximum number of characters (including NULL terminator) that the MEID string array can contain
MEIDString	• NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed • Maximum Length is 255 Bytes
imeiSvnSize	• The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain.

ImeiSvnString	• NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed. • Maximum Length is 255 Bytes
Tlvresult	• unpack Tlv Result.

8.1044.2 Field Documentation

8.1044.2.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

8.1044.2.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

8.1044.2.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

8.1044.2.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

8.1044.2.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

8.1044.2.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

8.1044.2.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

8.1044.2.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

8.1044.2.9 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

8.1045 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

8.1045.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareInfo parameters

Parameters

modelid_str	- Mode ID String. • NULL-terminated Mode ID String.
bootversion_str	- Boot Version. • NULL-terminated Boot Version String.
appversion_str	- Application Version String. • NULL-terminated Application Version String.
sku_str	- SKU String. • NULL-terminated SKU String.
packageid_str	- Package ID String. • NULL-terminated Package ID String. • deprecated on EM/MC74xx(9x30) devices
carrier_str	- Carrier String. • NULL-terminated Carrier String.
priversion_str	- PRI Version String. • NULL-terminated PRI Version String.
cur_carr_name	- Current Carrier Name String. • NULL-terminated Current Carrier Name String.
cur_carr_rev	- Current Carrier Revision String. • NULL-terminated Current Carrier Revision String.
Tlvresult	- unpack Tlv Result.

8.1045.2 Field Documentation

8.1045.2.1 char unpack_dms_GetFirmwareInfo_t::appversion_str[85]

8.1045.2.2 char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]

8.1045.2.3 char unpack_dms_GetFirmwareInfo_t::carrier_str[20]

8.1045.2.4 char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]

8.1045.2.5 char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]

8.1045.2.6 char unpack_dms_GetFirmwareInfo_t::modelid_str[20]

8.1045.2.7 char unpack_dms_GetFirmwareInfo_t::packageid_str[85]

8.1045.2.8 char unpack_dms_GetFirmwareInfo_t::priversion_str[16]

8.1045.2.9 char unpack_dms_GetFirmwareInfo_t::sku_str[15]

8.1045.2.10 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

8.1046 unpack_dms_GetFirmwareRevision_t Struct Reference

Data Fields

- uint8_t amssSize
- char AMSSString [255]
- char PRIString [255]
- uint16_t Tlvresult

8.1046.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareRevision parameters

Parameters

amssSize	• Maximum number of characters (including NULL terminator) that the AMSS string array can contain.
AMSSString	• NULL-terminated AMSS revision string. • Maximum Length is 255 Bytes
PRIString[OUT]	• NULL-terminated PRI revision string. • Maximum Length is 255 Bytes
Tlvresult	• unpack Tlv Result.

8.1046.2 Field Documentation

8.1046.2.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

8.1046.2.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

8.1046.2.3 char unpack_dms_GetFirmwareRevision_t::PRIString[255]

8.1046.2.4 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

8.1047 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

8.1047.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareRevisions parameters

Parameters

amssSize	• Maximum number of characters (including NULL terminator) that the AMSS string array can contain
AMSSString	• NULL-terminated AMSS revision string • Maximum Length is 255 Bytes
bootSize	• Maximum number of characters (including NULL terminator) that the boot string array can contain
BootString	• NULL-terminated boot code revision string • Maximum Length is 255 Bytes
priSize	• Maximum number of characters (including NULL terminator) that the PRI string array can contain
PRIString	• NULL-terminated PRI revision string • Maximum Length is 255 Bytes
Tlvresult	• unpack Tlv Result

8.1047.2 Field Documentation

8.1047.2.1 uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize

8.1047.2.2 char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]

8.1047.2.3 uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize

8.1047.2.4 char unpack_dms_GetFirmwareRevisions_t::BootString[255]

8.1047.2.5 uint8_t unpack_dms_GetFirmwareRevisions_t::priSize

8.1047.2.6 char unpack_dms_GetFirmwareRevisions_t::PRIString[255]

8.1047.2.7 uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult

8.1048 unpack_dms_GetFSN_t Struct Reference

Data Fields

- char String [255]

- uint16_t Tlvresult

8.1048.1 Detailed Description

This structure used to store Factory Sequence Number parameter

Parameters

String	• Facorty Sequence Number • Maximum Length is 255 Bytes
Tlvresult	• Unpack Result

8.1048.2 Field Documentation

8.1048.2.1 char unpack_dms_GetFSN_t::String[255]

8.1048.2.2 uint16_t unpack_dms_GetFSN_t::Tlvresult

8.1049 unpack_dms_GetHardwareRevision_t Struct Reference

Data Fields

- char hwVer [255]

8.1049.1 Detailed Description

This structure used to store unpack_dms_GetHardwareRevision parameters

Parameters

hwVer	- Hardware version • NULL-terminated string • Maximum Length is 255 Bytes

8.1049.2 Field Documentation

8.1049.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

8.1050 unpack_dms_GetIMSI_t Struct Reference

Data Fields

- char imsi [255]
- uint16_t Tlvresult

8.1050.1 Detailed Description

This structure used to store unpack_dms_GetIMSI parameters

Parameters

imsi	- IMSI no. • NULL-terminated String.
Tlvresult	- unpack Tlv Result.

8.1050.2 Field Documentation

8.1050.2.1 char unpack_dms_GetIMSI_t::imsi[255]

8.1050.2.2 uint16_t unpack_dms_GetIMSI_t::Tlvresult

8.1051 unpack_dms_GetManufacturer_t Struct Reference

Data Fields

- char manufacturer [255]
- uint16_t Tlvresult

8.1051.1 Detailed Description

This structure is used to store device manufacturer information.

Parameters

manufacturer[OUT]	• NULL terminated string
Tlvresult	• Unpack Result

8.1051.2 Field Documentation

8.1051.2.1 char unpack_dms_GetManufacturer_t::manufacturer[255]

8.1051.2.2 uint16_t unpack_dms_GetManufacturer_t::Tlvresult

8.1052 unpack_dms_GetModelID_t Struct Reference

Data Fields

- char modelid [255]
- uint16_t Tlvresult

8.1052.1 Detailed Description

This structure used to store unpack_dms_GetModelID parameters

Parameters

modelid	- Device model id. • NULL-terminated String.
Tlvresult	- unpack Tlv Result.

8.1052.2 Field Documentation

8.1052.2.1 char unpack_dms_GetModelID_t::modelid[255]

8.1052.2.2 uint16_t unpack_dms_GetModelID_t::Tlvresult

8.1053 unpack_dms_GetNetworkTime_t Struct Reference

Data Fields

- uint16_t source
- uint64_t timestamp
- uint16_t Tlvresult

8.1053.1 Detailed Description

This structure used to store unpack_dms_GetNetworkTime parameters

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.

8.1053.2 Field Documentation

8.1053.2.1 uint16_t unpack_dms_GetNetworkTime_t::source

8.1053.2.2 uint64_t unpack_dms_GetNetworkTime_t::timestamp

8.1053.2.3 uint16_t unpack_dms_GetNetworkTime_t::Tlvresult

8.1054 unpack_dms_GetOfflineReason_t Struct Reference

Data Fields

- uint32_t * pReasonMask
- uint32_t * pbPlatform
- uint16_t Tlvresult

8.1054.1 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

8.1054.2 Field Documentation

8.1054.2.1 uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform

8.1054.2.2 uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask

8.1054.2.3 uint16_t unpack_dms_GetOfflineReason_t::Tlvresult

8.1055 unpack_dms_GetPower_t Struct Reference

Data Fields

- uint32_t OperationMode
- uint32_t OfflineReason
- uint32_t HardwareControlledMode
- uint16_t Tlvresult

8.1055.1 Detailed Description

This structure used to store unpack_dms_GetPower parameters

Parameters

OperationMode	- operating mode. • 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.

Note

Valid transitions for Power Modes

- Online to Low Power, Persistent low power, Factory test, Offline or Shut Down
- Low power to online, Persistent low power, Offline, or Shut Down
- Persistent low power to Online, Low power, Offline or Shut down
- Factory test to online
- Offline to Reset

Parameters

OfflineReason	- offline reason.
Hardware-ControlledMode	- hardware restricted mode.
Tlvresult	- unpack Tlv Result.

8.1055.2 Field Documentation

8.1055.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

8.1055.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

8.1055.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

8.1055.2.4 uint16_t unpack_dms_GetPower_t::Tlvresult

8.1056 unpack_dms_GetPRLVersion_t Struct Reference

Data Fields

- uint8_t u8PRLPreference
- uint16_t u16PRLVersion
- uint16_t Tlvresult

8.1056.1 Detailed Description

This structure used to store unpack_dms_GetPRLVersion parameters

Parameters

u8PRL-Preference	• PRL Preference – 0 - Unset – 1 - Set
u16PRLVersion	• PRL version of device.
Tlvresult	• unpack Tlv Result

8.1056.2 Field Documentation

8.1056.2.1 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

8.1056.2.2 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

8.1056.2.3 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

8.1057 unpack_dms_GetSerialNumbers_t Struct Reference

Data Fields

- char esn [255]
- char imei_no [255]
- char meid [255]
- char imeisv_svn [255]

8.1057.1 Detailed Description

This structure used to store unpack_dms_GetSerialNumbers parameters

Parameters

esn	- Electronic Serial Number of the device • NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed
imei_no	- International Mobile Equipment Identity of the device. • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed
meid	- Mobile Equipment Identifier of the device. • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed
imeisv_svn	- NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.imei software version revision

8.1057.2 Field Documentation

8.1057.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

8.1057.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

8.1057.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

8.1057.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

8.1058 unpack_dms_GetUSBComp_t Struct Reference

Data Fields

- uint8_t USBComp [255]
- uint8_t NumSupUSBComps
- uint8_t SupUSBComps
- uint16_t Tlvresult

8.1058.1 Detailed Description

This structure is used to store retrieved USB Composition

Parameters

USBComp	• 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
NumSupUSB-Comps	• Number of supported USB compositions in the parameter to follow • Range - 0-255

SupUSBComps	• 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
Tlvresult	• unpack Tlv Result.

8.1058.2 Field Documentation

8.1058.2.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

8.1058.2.2 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

8.1058.2.3 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

8.1058.2.4 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

8.1059 unpack_dms_GetVoiceNumber_t Struct Reference

Data Fields

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

8.1059.1 Detailed Description

Structure to store the voice number in use by the device

Parameters

voiceNumber-Size	• Maximum number of characters (including NULL terminator) that the voice number array can contain.
VoiceNumber	• Voice number string: MDN or MS ISDN • Maximum Length is 255 Bytes
minSize	• Maximum number of characters (including NULL terminator) that the MIN array can contain.
MIN	• Optional Parameter • MIN string: Empty string returned when MIN is not supported/ programmed. • Maximum Length is 255 Bytes
Tlvresult	• Unpack Result

8.1059.2 Field Documentation

8.1059.2.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

8.1059.2.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

8.1059.2.3 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

8.1059.2.4 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

8.1059.2.5 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

8.1060 unpack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1060.1 Detailed Description

This structure contains reset to factory default unpack

Parameters

Tlvresult	• Unpack Result

8.1060.2 Field Documentation

8.1060.2.1 uint16_t unpack_dms_ResetToFactoryDefaults_t::Tlvresult

8.1061 unpack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1061.1 Detailed Description

This structure is used to store Set Service Activation Status callback parameter unpack.

Parameters

Tlvresult	• Unpack Result

8.1061.2 Field Documentation

8.1061.2.1 uint16_t unpack_dms_SetActivationStatusCallback_t::Tlvresult

8.1062 unpack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t notused

8.1062.1 Detailed Description

This structure is used to store unpack_dms_SetCrashAction parameters

Parameters

notused	• unpack Tlv result

8.1062.2 Field Documentation

8.1062.2.1 uint8_t unpack_dms_SetCrashAction_t::notused

8.1063 unpack_dms_SetCustFeature_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1063.1 Detailed Description

This structure is used to store unpack_dms_SetCustFeature parameters

Parameters

Tlvresult	• unpack Tlv Result.

8.1063.2 Field Documentation

8.1063.2.1 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

8.1064 unpack_dms_SetCustFeaturesV2_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1064.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

Tlvresult	• Unpack Result

8.1064.2 Field Documentation

8.1064.2.1 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

8.1065 unpack_dms_SetEventReport_ind_t Struct Reference

Data Fields

- dms_ActivationStatusTlv ActivationStatusTlv
- dms_OperatingModeTlv OperatingModeTlv
- uint16_t Tlvresult

8.1065.1 Detailed Description

DMS Event Report indication structure

Parameters

ActivationStatus-Tlv	• See dms_ActivationStatusTlv
OperatingMode-Tlv	• See dms_OperatingModeTlv

Tlvresult	• Unpack Result

8.1065.2 Field Documentation

8.1065.2.1 dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv

8.1065.2.2 dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv

8.1065.2.3 uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult

8.1066 unpack_dms_SetEventReport_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1066.1 Detailed Description

This structure is used to store unpack_dms_SetEventReport parameters.

Parameters

Tlvresult	• unpack Tlv result

8.1066.2 Field Documentation

8.1066.2.1 uint16_t unpack_dms_SetEventReport_t::Tlvresult

8.1067 unpack_dms_SetFirmwarePreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1067.1 Detailed Description

This structure is used to store unpack_dms_SetFirmwarePreference parameters

Parameters

Tlvresult	• unpack Tlv result

8.1067.2 Field Documentation

8.1067.2.1 uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult

8.1068 unpack_dms_SetPower_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1068.1 Detailed Description

Structure to store unpack the operating mode of the device.

Parameters

Tlvresult	• unpack Tlv Result.

8.1068.2 Field Documentation

8.1068.2.1 uint16_t unpack_dms_SetPower_t::Tlvresult

8.1069 unpack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1069.1 Detailed Description

This structure is used to store unpack USB composition information

Parameters

Tlvresult	• unpack Tlv Result.

8.1069.2 Field Documentation

8.1069.2.1 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

8.1070 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult

8.1070.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

type[OUT]	• type of reset or power down, possible values listed below: – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down
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[OUT]	• Unpack Result

8.1070.2 Field Documentation

8.1070.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

8.1070.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

8.1070.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

8.1071 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult

8.1071.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

type[OUT]	• type of reset or power down, possible values listed below: – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down
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

8.1071.2 Field Documentation

8.1071.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

8.1071.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

8.1071.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

8.1072 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1072.1 Detailed Description

This structure contains set registration state for different indication unpack

Parameters

Tlvresult	• Unpack Result

8.1072.2 Field Documentation

8.1072.2.1 uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::TlvResult

8.1073 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

8.1073.1 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

8.1073.2 Field Documentation

8.1073.2.1 uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability

8.1073.2.2 int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available

8.1073.2.3 int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available

8.1073.2.4 uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability

8.1073.2.5 uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability

8.1074 unpack_dms_SLQSGetERIFile_t Struct Reference

Data Fields

- uint16_t Tlvresult
- eriDataparams eriFile

8.1074.1 Detailed Description

This structure contains Get ERI file unpack

Parameters

Tlvresult	• Unpack Result
eriFile	• Pointer to structure ERIFileparams • See eriDataparams for more information

8.1074.2 Field Documentation

8.1074.2.1 eriDataparams unpack_dms_SLQSGetERIFile_t::eriFile

8.1074.2.2 uint16_t unpack_dms_SLQSGetERIFile_t::Tlvresult

8.1075 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

Data Fields

- uint16_t TlvresultTlvresult

8.1075.1 Detailed Description

This structure contains Clear Dying GASP unpack

Parameters

Tlvresult	• Unpack Result

8.1075.2 Field Documentation

8.1075.2.1 uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult

8.1076 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference

Data Fields

- uint16_t TlvresultTlvresult
- crashInfoParams crashInfoParam

8.1076.1 Detailed Description

This structure contains SWI get crash information unpack information

Parameters

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

8.1076.2 Field Documentation

8.1076.2.1 crashInfoParams unpack_dms_SLQSSwiGetCrashInfo_t::crashInfoParam

8.1076.2.2 uint16_t unpack_dms_SLQSSwiGetCrashInfo_t::Tlvresult

8.1077 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

Data Fields

- packgetDyingGaspCfg * pGetDyingGaspCfg
- uint16_t TlvresultTlvresult

8.1077.1 Detailed Description

This structure contains Get Dying GASP Config unpack

Parameters

pGetDyingGasp-Cfg	• See packgetDyingGaspCfg
Tlvresult	• Unpack Result

8.1077.2 Field Documentation

8.1077.2.1 packgetDyingGaspCfg∗ unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg

8.1077.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult

8.1078 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference

Data Fields

- packgetDyingGaspStatistics ∗ pGetDyingGaspStatistics
- uint16_t Tlvresult

8.1078.1 Detailed Description

This structure contains Get Dying GASP Statistics unpack.

Parameters

pGetDyingGasp-Statistics	• See packgetDyingGaspStatistics
Tlvresult	• Unpack Result

8.1078.2 Field Documentation

8.1078.2.1 packgetDyingGaspStatistics∗ unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics

8.1078.2.2 uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult

8.1079 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]

8.1079.1 Detailed Description

This structure is used to store unpack_dms_SLQSSwiGetFirmwareCurr parameters

Parameters

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

8.1079.2 Field Documentation

8.1079.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

8.1079.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

8.1079.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

8.1079.2.4 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

8.1079.2.5 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

8.1079.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

8.1080 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

8.1080.1 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

8.1080.2 Field Documentation

8.1080.2.1 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType

8.1080.2.2 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]

8.1080.2.3 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData

8.1080.2.4 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]

8.1080.2.5 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode

8.1080.2.6 uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult

8.1081 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference

Data Fields

- uint16_t Tlvresult
- char manString [255]
- char modelString [255]
- char swVerString [255]
- char plasmaIDString [255]

8.1081.1 Detailed Description

This structure contains SWI get host device info unpack information

Parameters

Tlvresult	• unpack Result
manString[OUT]	• Host Device Manufacturer Name • Null terminated ASCII String
modelString[OUT]	• Host Device Model String • Null terminated ASCII String
swVerString[OUT]	• Host Device Software Version String • Null terminated ASCII string
plasmaIDString[OUT]	• Host Device Plasma ID String • Null terminated alphanumeric ASCII String.

8.1081.2 Field Documentation

8.1081.2.1 char unpack_dms_SLQSSwiGetHostDevInfo_t::manString[255]

8.1081.2.2 char unpack_dms_SLQSSwiGetHostDevInfo_t::modelString[255]

8.1081.2.3 char unpack_dms_SLQSSwiGetHostDevInfo_t::plasmaIDString[255]

8.1081.2.4 char unpack_dms_SLQSSwiGetHostDevInfo_t::swVerString[255]

8.1081.2.5 uint16_t unpack_dms_SLQSSwiGetHostDevInfo_t::Tlvresult

8.1082 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference

Data Fields

- uint16_t TlvResult
- char nameString [255]
- char versionString [255]

8.1082.1 Detailed Description

This structure contains SWI get host os info unpack information

Parameters

TlvResult	• Unpack Result
nameString[OUT]	• Host operating system name
VersionString[OUT]	• Host operating system version

8.1082.2 Field Documentation

8.1082.2.1 char unpack_dms_SLQSSwiGetOSInfo_t::nameString[255]

8.1082.2.2 uint16_t unpack_dms_SLQSSwiGetOSInfo_t::TlvResult

8.1082.2.3 char unpack_dms_SLQSSwiGetOSInfo_t::versionString[255]

8.1083 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference

Data Fields

- uint16_t TlvResult
- char meidString [8]

8.1083.1 Detailed Description

This structure contains SWI get device serial number extension unpack information

Parameters

TlvResult	• pack Result
meidString[OUT]	• optional parameter, mobile equipment identifier

8.1083.2 Field Documentation

8.1083.2.1 char unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]

8.1083.2.2 uint16_t unpack_dms_SLQSSwiGetSerialNoExt_t::Tlvresult

8.1084 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1084.1 Detailed Description

This structure contains set Dying GASP Config unpack

Parameters

Tlvresult	• Unpack Result

8.1084.2 Field Documentation

8.1084.2.1 uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

8.1085 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1085.1 Detailed Description

This structure contains SWI set host dev info unpack

Parameters

Tlvresult	• Unpack Result

8.1085.2 Field Documentation

8.1085.2.1 uint16_t unpack_dms_SLQSSwiSetHostDevInfo_t::Tlvresult

8.1086 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1086.1 Detailed Description

This structure contains SWI set host OS info unpack

Parameters

TlvresultResult	• Unpack Result

8.1086.2 Field Documentation

8.1086.2.1 uint16_t unpack_dms_SLQSSwiSetOSInfo_t::TlvresultResult

8.1087 unpack_dms_SLQSUIMGetState_t Struct Reference

Data Fields

- uint8_t state
- uint16_t TlvresultResult

8.1087.1 Detailed Description

This structure contains UIM get state unpack information

Parameters

state[OUT]	• UIM state
TlvresultResult	• Unpack Result

8.1087.2 Field Documentation

8.1087.2.1 uint8_t unpack_dms_SLQSUIMGetState_t::state

8.1087.2.2 uint16_t unpack_dms_SLQSUIMGetState_t::TlvresultResult

8.1088 unpack_dms_UIMGetControlKeyStatus_t Struct Reference

Data Fields

- uint8_t facilityState
- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t TlvresultResult

8.1088.1 Detailed Description

This structure contains PIN retries status

Parameters

facilityState[OUT]	• Control key status – 0 - Deactivated – 1 - Activated – 2 - Blocked
verifyRetriesLeft[OUT]	• Optional parameter • Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. – 0xFFFFFFFF - Unknown
unblockRetriesLeft[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

8.1088.2 Field Documentation

8.1088.2.1 uint8_t unpack_dms_UIMGetControlKeyStatus_t::facilityState

8.1088.2.2 uint16_t unpack_dms_UIMGetControlKeyStatus_t::Tlvresult

8.1088.2.3 uint8_t unpack_dms_UIMGetControlKeyStatus_t::unblockRetriesLeft

8.1088.2.4 uint8_t unpack_dms_UIMGetControlKeyStatus_t::verifyRetriesLeft

8.1089 unpack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint8_t stringSize
- uint8_t String [255]
- uint16_t Tlvresult

8.1089.1 Detailed Description

This structure is used to store unpack_dms_UIMGetICCID parameters.

Parameters

stringSize	• Size of String.
String	• ICCID String. • Maximum Length is 255 Bytes

Tlvresult	• Unack result.

8.1089.2 Field Documentation

8.1089.2.1 uint8_t unpack_dms_UIMGetICCID_t::String[255]

8.1089.2.2 uint8_t unpack_dms_UIMGetICCID_t::stringSize

8.1089.2.3 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

8.1090 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

8.1090.1 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

8.1090.2 Field Documentation

8.1090.2.1 uint8_t unpack_dms_UIMGetPINStatus_t::p1Status

8.1090.2.2 uint8_t unpack_dms_UIMGetPINStatus_t::p1UnblockRetriesLeft

8.1090.2.3 uint8_t unpack_dms_UIMGetPINStatus_t::p1VerifyRetriesLeft

8.1090.2.4 uint8_t unpack_dms_UIMGetPINStatus_t::p2Status

8.1090.2.5 uint8_t unpack_dms_UIMGetPINStatus_t::p2UnblockRetriesLeft

8.1090.2.6 uint8_t unpack_dms_UIMGetPINStatus_t::p2VerifyRetriesLeft

8.1090.2.7 uint16_t unpack_dms_UIMGetPINStatus_t::Tlvresult

8.1091 unpack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

- uint8_t verifyRetriesLeft
- uint16_t Tlvresult

8.1091.1 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

8.1091.2 Field Documentation

8.1091.2.1 uint16_t unpack_dms_UIMSetControlKeyProtection_t::Tlvresult

8.1091.2.2 uint8_t unpack_dms_UIMSetControlKeyProtection_t::verifyRetriesLeft

8.1092 unpack_dms_UIMSetPINProtection_t Struct Reference

Data Fields

- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

8.1092.1 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[OUT]	• Unpack Result

8.1092.2 Field Documentation

8.1092.2.1 uint16_t unpack_dms_UIMSetPINProtection_t::Tlvresult

8.1092.2.2 uint8_t unpack_dms_UIMSetPINProtection_t::unblockRetriesLeft

8.1092.2.3 uint8_t unpack_dms_UIMSetPINProtection_t::verifyRetriesLeft

8.1093 unpack_dms_UIMUnblockControlKey_t Struct Reference

Data Fields

- uint8_t unblockRetriesLeft
- uint16_t Tlvresult

8.1093.1 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[OUT]	• Unpack Result

8.1093.2 Field Documentation

8.1093.2.1 uint16_t unpack_dms_UIMUnblockControlKey_t::Tlvresult

8.1093.2.2 uint8_t unpack_dms_UIMUnblockControlKey_t::unblockRetriesLeft

8.1094 unpack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint32_t ImageListSize
- FMSPrefImageList * pImageList
- uint16_t Tlvresult

8.1094.1 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

8.1094.2 Field Documentation

8.1094.2.1 uint32_t unpack_fms_GetImagesPreference_t::ImageListSize

8.1094.2.2 FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList

8.1094.2.3 uint16_t unpack_fms_GetImagesPreference_t::Tlvresult

8.1095 unpack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint32_t imagelistSize
- FMSImageList imageList
- uint16_t Tlvresult

8.1095.1 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

8.1095.2 Field Documentation

8.1095.2.1 FMSImageList unpack_fms_GetStoredImages_t::imageList

8.1095.2.2 uint32_t unpack_fms_GetStoredImages_t::imagelistSize

8.1095.2.3 uint16_t unpack_fms_GetStoredImages_t::Tlvresult

8.1096 unpack_fms_SetImagesPreference_t Struct Reference

Data Fields

- uint32_t ImageTypesSize
- uint8_t ImageTypes [255]
- uint16_t Tlvresult

8.1096.1 Detailed Description

This structure contains the Set Images Preference unpack

Parameters

ImageTypesSize	• Image Type Size
ImageTypes	• Image Type
TlvResult	• Unpack result

8.1096.2 Field Documentation

8.1096.2.1 uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]

8.1096.2.2 uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize

8.1096.2.3 uint16_t unpack_fms_SetImagesPreference_t::TlvResult

8.1097 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

8.1097.1 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

8.1097.2 Field Documentation

8.1097.2.1 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid

8.1097.2.2 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel

8.1097.2.3 loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime

8.1097.2.4 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading

8.1097.2.5 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc

8.1097.2.6 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf

8.1097.2.7 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf

8.1097.2.8 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability

8.1097.2.9 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular

8.1097.2.10 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth

8.1097.2.11 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor

8.1097.2.12 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor

8.1097.2.13 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude

8.1097.2.14 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude

8.1097.2.15 uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation

8.1097.2.16 loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution

8.1097.2.17 loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage

8.1097.2.18 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal

8.1097.2.19 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc

8.1097.2.20 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical

8.1097.2.21 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc

8.1097.2.22 loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix

8.1097.2.23 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask

8.1097.2.24 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc

8.1097.2.25 uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc

8.1097.2.26 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc

8.1097.2.27 uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence

8.1097.2.28 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability

8.1097.2.29 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc

8.1097.2.30 uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid

8.1097.2.31 uint32_t unpack_loc_BestAvailPos_Ind_t::status

8.1097.2.32 uint16_t unpack_loc_BestAvailPos_Ind_t::TlvResult

8.1098 unpack_loc_Delete_Assist_Data_t Struct Reference

Data Fields

- uint16_t TlvResult

8.1098.1 Detailed Description

This structure contains LOC delete assist data unpack

Parameters

Tlvresult	• Unpack result.

8.1098.2 Field Documentation

8.1098.2.1 uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult

8.1099 unpack_loc_DeleteAssistData_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1099.1 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

8.1099.2 Field Documentation

8.1099.2.1 uint32_t unpack_loc_DeleteAssistData_Ind_t::status

8.1099.2.2 uint16_t unpack_loc_DeleteAssistData_Ind_t::Tlvresult

8.1100 unpack_loc_EngineState_Ind_t Struct Reference

Data Fields

- uint32_t engineState
- uint16_t Tlvresult

8.1100.1 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

8.1100.2 Field Documentation

8.1100.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

8.1100.2.2 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

8.1101 unpack_loc_EventNMEA_Ind_t Struct Reference

Data Fields

- char NMEAData [201]
- uint16_t Tlvresult

8.1101.1 Detailed Description

Contain the parameters passed for SetLocEventMaskNMEACallback by the device.

Parameters

NMEAData	• NMEA string • Type - NULL terminated string • Maximum string length (including NULL terminator) - 201

TlvresultUnpack result	• unpack result

8.1101.2 Field Documentation

8.1101.2.1 char unpack_loc_EventNMEA_Ind_t::NMEAData[201]

8.1101.2.2 uint16_t unpack_loc_EventNMEA_Ind_t::Tlvresult

8.1102 unpack_loc_EventRegister_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1102.1 Detailed Description

This structure contains Event Register unpack

Parameters

TlvresultUnpack result.	• Unpack result.

8.1102.2 Field Documentation

8.1102.2.1 uint16_t unpack_loc_EventRegister_t::Tlvresult

8.1103 unpack_loc_GetServer_Ind_t Struct Reference

Data Fields

- uint32_t serverStatus
- uint32_t serverType
- loc_IPv4Info * pIPv4AddrInfo
- loc_IPv6Info * pIPv6AddrInfo
- loc_urlAddr * pURL
- uint16_t Tlvresult

8.1103.1 Detailed Description

Contain the parameters passed for SetLocGetServerCallback by the device.

Parameters

serverStatus	• Status of the Get Server 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
serverType	• Type of server. • Valid values – eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE – eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC – eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP – eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) - Server type is custom PDE
pIPv4AddrInfo	• See loc_IPv4Info for more information.
pIPv6AddrInfo	• See loc_IPv6Info for more information.
pURL	• See loc_urlAddr for more information.

8.1103.2 Field Documentation

8.1103.2.1 loc_IPv4Info* unpack_loc_GetServer_Ind_t::pIPv4AddrInfo

8.1103.2.2 loc_IPv6Info* unpack_loc_GetServer_Ind_t::pIPv6AddrInfo

8.1103.2.3 loc_urlAddr* unpack_loc_GetServer_Ind_t::pURL

8.1103.2.4 uint32_t unpack_loc_GetServer_Ind_t::serverStatus

8.1103.2.5 uint32_t unpack_loc_GetServer_Ind_t::serverType

8.1103.2.6 uint16_t unpack_loc_GetServer_Ind_t::TlvResult

8.1104 unpack_loc_GnssSvInfo_Ind_t Struct Reference

Data Fields

- uint8_t altitudeAssumed
- loc_satelliteInfo * pSatelliteInfo
- uint16_t TlvResult

8.1104.1 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

8.1104.2 Field Documentation

8.1104.2.1 uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed

8.1104.2.2 loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo

8.1104.2.3 uint16_t unpack_loc_GnssSvInfo_Ind_t::TlvResult

8.1105 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

8.1105.1 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

8.1105.2 Field Documentation

8.1105.2.1 uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed

8.1105.2.2 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid

8.1105.2.3 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel

8.1105.2.4 uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId

8.1105.2.5 loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime

8.1105.2.6 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading

8.1105.2.7 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc

8.1105.2.8 uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence

8.1105.2.9 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability

8.1105.2.10 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular

8.1105.2.11 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth

8.1105.2.12 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor

8.1105.2.13 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor

8.1105.2.14 uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude

8.1105.2.15 uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds

8.1105.2.16 uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude

8.1105.2.17 uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation

8.1105.2.18 loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution

8.1105.2.19 loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage

8.1105.2.20 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal

8.1105.2.21 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc

8.1105.2.22 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical

8.1105.2.23 loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix

8.1105.2.24 uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask

8.1105.2.25 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc

8.1105.2.26 uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc

8.1105.2.27 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc

8.1105.2.28 uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence

8.1105.2.29 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability

8.1105.2.30 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc

8.1105.2.31 uint8_t unpack_loc_PositionRpt_Ind_t::sessionId

8.1105.2.32 uint32_t unpack_loc_PositionRpt_Ind_t::sessionStatus

8.1105.2.33 uint16_t unpack_loc_PositionRpt_Ind_t::Tlvresult

8.1106 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1106.1 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

8.1106.2 Field Documentation

8.1106.2.1 uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status

8.1106.2.2 uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult

8.1107 unpack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1107.1 Detailed Description

This structure contains Set Ext Power State unpack

Parameters

Tlvresult	• Unpack result.

8.1107.2 Field Documentation

8.1107.2.1 uint16_t unpack_loc_SetExtPowerState_t::Tlvresult

8.1108 unpack_loc_SetOperationMode_Ind_t Struct Reference

Data Fields

- uint32_t status
- uint16_t Tlvresult

8.1108.1 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

8.1108.2 Field Documentation

8.1108.2.1 uint32_t unpack_loc_SetOperationMode_Ind_t::status

8.1108.2.2 uint16_t unpack_loc_SetOperationMode_Ind_t::Tlvresult

8.1109 unpack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1109.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

Tlvresult	• Unpack result.

8.1109.2 Field Documentation

8.1109.2.1 uint16_t unpack_loc_SetOperationMode_t::Tlvresult

8.1110 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1110.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

Tlvresult	• Unpack result.

8.1110.2 Field Documentation

8.1110.2.1 uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.1111 unpack_loc_Start_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1111.1 Detailed Description

This structure contains Start LOC unpack

Parameters

Tlvresult	• Unpack result.

8.1111.2 Field Documentation

8.1111.2.1 uint16_t unpack_loc_Start_t::Tlvresult

8.1112 unpack_loc_Stop_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1112.1 Detailed Description

This structure contains Stop LOC unpack

Parameters

Tlvresult	• Unpack result.

8.1112.2 Field Documentation

8.1112.2.1 uint16_t unpack_loc_Stop_t::Tlvresult

8.1113 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

8.1113.1 Detailed Description

Gets the current CDMA network parameters

Parameters

SCI	• Slot cycle index – 0xFF-Unknown
SCM	• Station class mark – 0xFF-Unknown
RegHomeSID	• Register on home SID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown

RegForeignSID	• Register on foreign SID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
RegForeignNID	• Register on foreign NID – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
ForceRev0	• Force CDMA 1x-EV-DO Rev. 0 mode – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
CustomSCP	• Use a custom config for CDMA 1x-EV-DO SCP – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
Protocol	• 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
Broadcast	• Broadcast mask for custom SCP config – 0x00000001 - Generic broadcast enabled – 0xFFFFFFFF - Unknown
Application	• Application mask for custom SCP config – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application – 0xFFFFFFFF - Unknown

Roaming	• Roaming preference – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming – 0xFFFFFFFF - Unknown

8.1113.2 Field Documentation

8.1113.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

8.1113.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

8.1113.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

8.1113.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

8.1113.2.5 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

8.1113.2.6 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

8.1113.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

8.1113.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

8.1113.2.9 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

8.1113.2.10 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

8.1113.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

8.1114 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

8.1114.1 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 (liteqmi_helper_decode7bitAsciiEncString) – 0x04 - Unicode – 0x09 - GSM 7-bit default
nw2DescLen	• Network Description Length (3GPP2 only).

pNw2Name	• Network Name (3GPP2 only).

8.1114.2 Field Documentation

8.1114.2.1 uint8_t unpack_nas_GetHomeNetwork3GPP2_t::nameSize

8.1114.2.2 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMCC

8.1114.2.3 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMNC

8.1114.2.4 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pName

8.1114.2.5 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNID

8.1114.2.6 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescDisp

8.1114.2.7 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescEnc

8.1114.2.8 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescLen

8.1114.2.9 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MCC

8.1114.2.10 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MNC

8.1114.2.11 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2Name

8.1114.2.12 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pSID

8.1115 unpack_nas_GetHomeNetwork_t Struct Reference

Data Fields

- uint16_t mcc
- uint16_t mnc
- char name [255]
- uint16_t sid
- uint16_t nid

8.1115.1 Detailed Description

Parameters

mcc	• Mobile country code (UMTS only).
mnc	• Mobile network code (UMTS only).
name	• Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).

sid	• Home network system ID – 0xFFFF - Unknown. – Only applies to cdma2000
nid	• Home network ID – 0xFFFF - Unknown. – Only applies to cdma2000

8.1115.2 Field Documentation

8.1115.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

8.1115.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

8.1115.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

8.1115.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

8.1115.2.5 uint16_t unpack_nas_GetHomeNetwork_t::sid

8.1116 unpack_nas_GetNetworkPreference_t Struct Reference

Data Fields

- uint32_t ActiveTechPref
- uint32_t Duration
- uint32_t PersistentTechPref
- uint16_t Tlvresult

8.1116.1 Detailed Description

Returns the network registration preference. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use unpack_nas_SLQSGetSys-SelectionPref() for new firmware versions and new modules

Parameters

TechnologyPref[-OUT]	• Bitmask representing the radio technology preference set. • 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

8.1116.2 Field Documentation

8.1116.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

8.1116.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

8.1116.2.3 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

8.1116.2.4 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

8.1117 unpack_nas_GetRFInfo_t Struct Reference

Data Fields

- uint8_t instancesSize
- RFBandInfoElements RFBandInfoElements [255]

8.1117.1 Detailed Description

Structur to strore the RFInfoList

Parameters

instanceSize	• 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.
RFBandInfo-Elements	• RF info instances array – See RFBandInfoElements for more information

8.1117.2 Field Documentation

8.1117.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

8.1117.2.2 RFBandInfoElements unpack_nas_GetRFInfo_t::RFBandInfoElements[255]

8.1118 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]

8.1118.1 Detailed Description

Parameters

Registration-State	• Registration state: – 0 - Not registered – 1 - Registered – 2 - Searching/Not Registered – 3 - Registration Denied – 4 - Unknown
CSDomain	• Circuit switch domain status: – 0 - Unknown/Not Applicable – 1 - Attached – 2 - Detached

PSDomain	• Packet switch domain status – 0 - Unknown/Not Applicable – 1 - Attached – 2 - Detached
RAN	• Type of radio access network on which mobile is registered: – 0 - Unknown – 1 - cdma2000 network – 2 - UMTS network
RadioIfacesSize	• Upon input, maximum number of elements that the radio interface array contain. • Upon successful output, actual number of elements in the radio interface array.
RadioIfaces	• An array of Radio Interface Technology – See qaGobiApiTableRadioInterfaces.h for the Radio Interface Technologies
Roaming	• Roaming indicator
MCC	• Mobile country code
MNC	• Mobile network code
nameSize	• Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks
Name	• Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks
DataCapsLen[IN/OUT]	• Upon input, the maximum number of elements the data capabilities array can contain. • Upon output, the actual number of elements in the data capabilities array.

DataCaps[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

8.1118.2 Field Documentation

8.1118.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

8.1118.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

8.1118.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

8.1118.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

8.1118.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

8.1118.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

8.1118.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

8.1118.2.8 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

8.1118.2.9 uint8_t unpack_nas_GetServingNetwork_t::RadioIfaces[255]

8.1118.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadioIfacesSize

8.1118.2.11 uint32_t unpack_nas_GetServingNetwork_t::RAN

8.1118.2.12 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

8.1118.2.13 uint32_t unpack_nas_GetServingNetwork_t::Roaming

8.1119 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

Data Fields

- uint8_t DataCapsLen
- uint8_t DataCaps [255]

8.1119.1 Detailed Description

Parameters

DataCapsLen	• 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.
DataCaps[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

8.1119.2 Field Documentation

8.1119.2.1 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]

8.1119.2.2 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen

8.1120 unpack_nas_GetSignalStrengths_t Struct Reference

Data Fields

- uint32_t len
- signed char rssi [8]
- uint32_t radio [8]

8.1120.1 Detailed Description

Parameters

len	number of rssi & radio items following
rssi	signal strength array
radio	radio interface array

8.1120.2 Field Documentation

8.1120.2.1 uint32_t unpack_nas_GetSignalStrengths_t::len

8.1120.2.2 uint32_t unpack_nas_GetSignalStrengths_t::radio[8]

8.1120.2.3 signed char unpack_nas_GetSignalStrengths_t::rssi[8]

8.1121 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

8.1121.1 Detailed Description

Contain the network scan information.

Parameters

p3GppNetwork-InstanceSize	• Upon input, maximum number of elements that the network info instance array can contain. • Upon successful output, the actual number of elements in the network info instance array.
p3GppNetwork-InfoInstances	• Network info instance array – See nas_QmiNas3GppNetworkInfo for more information
pRATInstance-Size	• 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.
pRATINstance	• RAT info instance array – See nas_QmiNas3GppNetworkRAT for more information
pPCSInstance-Size	• 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.
pPCSInstance	• PCS Digit info instance array – See nas_QmisNasPcsDigit for more information
pScanResult	• 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

8.1121.2 Field Documentation

8.1121.2.1 nas_QmiNas3GppNetworkInfo* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances

8.1121.2.2 uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize

8.1121.2.3 nas_QmisNasPcsDigit* unpack_nas_PerformNetworkScan_t::pPCSInstance

8.1121.2.4 uint8_t* unpack_nas_PerformNetworkScan_t::pPCSInstanceSize

8.1121.2.5 nas_QmiNas3GppNetworkRAT* unpack_nas_PerformNetworkScan_t::pRATINstance

8.1121.2.6 uint8_t* unpack_nas_PerformNetworkScan_t::pRATInstanceSize

8.1121.2.7 uint32_t* unpack_nas_PerformNetworkScan_t::pScanResult

8.1122 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference

Data Fields

- uint8_t dataCapsSize
- uint8_t dataCaps [255]

8.1122.1 Detailed Description

Parameters

dataCapsSize	Number of Data Capabilities
dataCaps	Data Capabilities

8.1122.2 Field Documentation

8.1122.2.1 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]

8.1122.2.2 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize

8.1123 unpack_nas_SetEventReportInd_t Struct Reference

Data Fields

- nas_SignalStrengthTlv SSTlv
- nas_RFInfoTlv RFTlv
- nas_RejectReasonTlv RRTlv
- nas_SLQSSignalStrengthsTlv SLQSSSTlv

8.1123.1 Detailed Description

Parameters

SSTlv	signal strength tlv
RFTlv	RF tlv
RRTlv	RR tlv
SLQSSSTlv	signal strength complete info tlv

8.1123.2 Field Documentation

8.1123.2.1 nas_RFInfoTlv unpack_nas_SetEventReportInd_t::RFTlv

8.1123.2.2 nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv

8.1123.2.3 nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv

8.1123.2.4 nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv

8.1124 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference

Data Fields

- nas_PhyCaAggScellIndType sPhyCaAggScellIndType
- nas_PhyCaAggScellDlBw sPhyCaAggScellDlBw
- nas_PhyCaAggScellInfo sPhyCaAggScellInfo
- nas_PhyCaAggPcellInfo sPhyCaAggPcellInfo
- nas_PhyCaAggScellIndex sPhyCaAggScellIndex

8.1124.1 Detailed Description

Structure for storing the LTEC PHY CA indication parameters.

Parameters

pPhyCaAggScellIndType	• See nas_PhyCaAggScellIndType for more information.
sPhyCaAggScellDlBw	• See nas_PhyCaAggScellDlBw for more information.
sPhyCaAggScellInfo	• See nas_PhyCaAggScellInfo for more information.
sPhyCaAggPcellInfo	• See nas_PhyCaAggPcellInfo for more information.
sPhyCaAggScellIndex	• See nas_PhyCaAggScellIndex for more information.

8.1124.2 Field Documentation

8.1124.2.1 nas_PhyCaAggPcellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggPcellInfo

8.1124.2.2 nas_PhyCaAggScellDlBw unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellDlBw

8.1124.2.3 nas_PhyCaAggScellIndex unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellIndex

8.1124.2.4 nas_PhyCaAggScellIndType unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellIndType

8.1124.2.5 nas_PhyCaAggScellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellInfo

8.1125 unpack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint16_t TlvResult

8.1125.1 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

8.1125.2 Field Documentation

8.1125.2.1 uint16_t unpack_nas_SetNetworkPreference_t::TlvResult

8.1126 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

Data Fields

- uint8_t roaming

8.1126.1 Detailed Description

Parameters

roaming	• Roaming Indication – 0 - Roaming – 1 - Home – 2 - Roaming partner – >2 - Operator defined values

8.1126.2 Field Documentation

8.1126.2.1 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

8.1127 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

Data Fields

- NASServingSystemInfo SSInfo
- uint16_t Tlvresult

8.1127.1 Detailed Description

Parameters

SSInfo	• Serving system parameters information – See NASServingSystemInfo for more details
Tlvresult	• unpack result

8.1127.2 Field Documentation

8.1127.2.1 NASServingSystemInfo unpack_nas_SetServingSystemCallback_ind_t::SSInfo

8.1127.2.2 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

8.1128 unpack_nas_SLQSGetErrorRate_t Struct Reference

Data Fields

- uint16_t * pCDMAFrameErrRate
- uint16_t * pHDRPackErrRate
- uint8_t * pGSMBER
- uint8_t * pWCDMABER

8.1128.1 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.

8.1128.2 Field Documentation

8.1128.2.1 uint16_t* unpack_nas_SLQSGetErrorRate_t::pCDMAFrameErrRate

8.1128.2.2 uint8_t* unpack_nas_SLQSGetErrorRate_t::pGSMBER

8.1128.2.3 uint16_t* unpack_nas_SLQSGetErrorRate_t::pHDRPackErrRate

8.1128.2.4 uint8_t* unpack_nas_SLQSGetErrorRate_t::pWCDMABER

8.1129 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

Data Fields

- NasGetLTECphyCaInfo LTECphyCaInfo
- uint16_t Tlvresult

8.1129.1 Detailed Description

Parameters

LTECphyCa	• Carrier aggregation event information – See NasGetLTECphyCaInfo for more details
Tlvresult	• unpack result

8.1129.2 Field Documentation

8.1129.2.1 NasGetLTECphyCaInfo unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCaInfo

8.1129.2.2 uint16_t unpack_nas_SlqsGetLTECphyCAInfo_t::Tlvresult

8.1130 unpack_nas_SLQSGetNetworkTime_t Struct Reference

Data Fields

- nas_timeInfo * p3GPP2TimeInfo
- nas_timeInfo * p3GPPTimeInfo

8.1130.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

Parameters

p3GPP2Time-Info	[Optional] • See nas_timeInfo for more information
p3GPPTimeInfo	[Optional] • See nas_timeInfo for more information

8.1130.2 Field Documentation

8.1130.2.1 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPP2TimeInfo

8.1130.2.2 nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPPTimeInfo

8.1131 unpack_nas_SLQSGetOperatorNameData_t Struct Reference

Data Fields

- nas_serviceProviderName * pSrvcProviderName
- nas_operatorPLMNList * pOperatorPLMNList
- nas_PLMNNetworkName * pPLMNNetworkName
- nas_operatorNameString * pOperatorNameString
- nas_PLMNNetworkNameData * pNITZInformation

8.1131.1 Detailed Description

Structure for Operator Name unpack.

Parameters

pSrvcProvider-Name	• 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.
pOperatorNameString	• Refer nas_operatorNameString for details (Optional). • Can provide NULL if this parameter is not required.
pNITZInformation	• Refer nas_PLMNNetworkNameData for details (Optional). • Can provide NULL if this parameter is not required.

8.1131.2 Field Documentation

8.1131.2.1 nas_PLMNNetworkNameData∗ unpack_nas_SLQSGetOperatorNameData_t::pNITZInformation

8.1131.2.2 nas_operatorNameString∗ unpack_nas_SLQSGetOperatorNameData_t::pOperatorNameString

8.1131.2.3 nas_operatorPLMNList∗ unpack_nas_SLQSGetOperatorNameData_t::pOperatorPLMNList

8.1131.2.4 nas_PLMNNetworkName∗ unpack_nas_SLQSGetOperatorNameData_t::pPLMNNetworkName

8.1131.2.5 nas_serviceProviderName∗ unpack_nas_SLQSGetOperatorNameData_t::pSrvcProviderName

8.1132 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]

8.1132.1 Field Documentation

8.1132.1.1 char unpack_nas_SLQSGetPLMNName_t::longName[255]

8.1132.1.2 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameCI

8.1132.1.3 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameEn

8.1132.1.4 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameLen

8.1132.1.5 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameSB

8.1132.1.6 uint8_t unpack_nas_SLQSGetPLMNName_t::shortName[255]

8.1132.1.7 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameCI

8.1132.1.8 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameEn

8.1132.1.9 char unpack_nas_SLQSGetPLMNName_t::shortNameLen

8.1132.1.10 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameSB

8.1132.1.11 char unpack_nas_SLQSGetPLMNName_t::spn[255]

8.1132.1.12 uint8_t unpack_nas_SLQSGetPLMNName_t::spnEncoding

8.1132.1.13 uint8_t unpack_nas_SLQSGetPLMNName_t::spnLength

8.1133 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

8.1133.1 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 nas_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 nas_dataSrvCapabilities for more information
CurrentPLMN	• Optional parameter indicating Current PLMN • See nas_currentPLMN for more information
SystemID	• Optional parameter indicating System ID
NetworkID	• Optional parameter indicating Network ID
BaseStationID	• Optional parameter indicating Base Station Identification Number
BaseStation-Latitude	• Optional parameter indicating Base station latitude in units of 0.25 sec,expressed as a two's complement signed number with positive numbers signifying North latitude
Basestation-Longitude	• Optional parameter indicating Base station longitude in units of 0.25 sec, expressed as a Two's complement signed number with positive numbers signifying East longitude
Roaming-IndicatorList	• Optional parameter indicating Roaming Indicator List • See nas_roamIndList for more information
defaultRoamInd	• Optional parameter indicating Default Roaming Indicator • Values: – 0x00 - Roaming – 0x01 - Home

Gpp2TimeZone	• Optional parameter indicating 3GPP2 Time Zone • See nas_qaQmi3Gpp2TimeZone for more information
CDMA_P_Rev	• Optional parameter indicating CDMA P_Rev in use
GppTimeZone	• Optional parameter indicating Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min. (signed value).
GppNetworkDS-TAdjustment	• Optional parameter indicating 3GPP network daylight saving adjustment • 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 nas_detailSvcInfo for more information
CDMASystem-InfoExt	• Optional parameter indicating CDMA System Info Ext • See nas_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 nas_callBarStatus for more information

8.1133.2 Field Documentation

8.1133.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID

8.1133.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude

8.1133.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude

8.1133.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus

8.1133.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev

8.1133.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt

8.1133.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID

8.1133.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo

8.1133.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN

8.1133.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities

8.1133.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd

8.1133.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo

8.1133.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd

8.1133.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone

8.1133.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment

8.1133.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone

8.1133.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality

8.1133.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac

8.1133.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID

8.1133.2.20 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd

8.1133.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal

8.1133.2.22 nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList

8.1133.2.23 nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem

8.1133.2.24 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID

8.1133.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

8.1134 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

8.1134.1 Detailed Description

This structure contains the Signal Strength Information

Parameters

signalStrength-ReqMask	• 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
rxSignal-StrengthListLen	• Number of elements in Receive Signal Strength List
rxSignal-StrengthList	• See nas_rxSignalStrengthListElement for more information
ecioListLen	• Number of elements in ECIO List
ecioList	• See nas_ecioListElement for more information
Io	• Received Io in dBm; IO is only applicable for 1xEV-DO
sinr	• SINR level – SINR is only applicable for 1xEV-DO; valid levels are 0 to 8 where maximum value for 0 - SINR_LEVEL_0 is -9 dB 1 - SINR_LEVEL_1 is -6 dB 2 - SINR_LEVEL_2 is -4.5 dB 3 - SINR_LEVEL_3 is -3 dB 4 - SINR_LEVEL_4 is -2 dB 5 - SINR_LEVEL_5 is +1 dB 6 - SINR_LEVEL_6 is +3 dB 7 - SINR_LEVEL_7 is +6 dB 8 - SINR_LEVEL_8 is +9 dB

errorRateListLen	• Number of elements in Error Rate List
errorRateList	• See nas_errorRateListElement for more information
rsrqInfo	• See nas_rsrqInformation for more information
ltesnr	• LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE
ltersrp	• LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE

8.1134.2 Field Documentation

8.1134.2.1 nas_ecioListElement unpack_nas_SLQSGetSignalStrength_t::ecioList[18]

8.1134.2.2 uint16_t unpack_nas_SLQSGetSignalStrength_t::ecioListLen

8.1134.2.3 nas_errorRateListElement unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]

8.1134.2.4 uint16_t unpack_nas_SLQSGetSignalStrength_t::errorRateListLen

8.1134.2.5 int32_t unpack_nas_SLQSGetSignalStrength_t::lo

8.1134.2.6 int16_t unpack_nas_SLQSGetSignalStrength_t::ltersrp

8.1134.2.7 int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr

8.1134.2.8 nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo

8.1134.2.9 nas_rxSignalStrengthListElement unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]

8.1134.2.10 uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen

8.1134.2.11 uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask

8.1134.2.12 uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr

8.1135 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

8.1135.1 Detailed Description

Parameters

pCDMASrvStatusInfo	• See nas_SrvStatusInfo for more information.
pHDRSrvStatusInfo	• See nas_SrvStatusInfo for more information.
pGSMSrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pWCDMASrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pLTESrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pCDMASysInfo	• See nas_CDMASysInfo for more information.
pHDRSysInfo	• See nas_HDRSysInfo for more information.
pGSMSysInfo	• See nas_GSMSysInfo for more information.
pWCDMASysInfo	• See nas_WCDMASysInfo for more information.
pLTESysInfo	• See nas_LTESysInfo for more information.
pAddCDMASysInfo	• See nas_AddCDMASysInfo for more information.

pAddHDRSysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
pAddGSMSysInfo	• See nas_AddSysInfo for more information.
pAddWCDMASysInfo	• See nas_AddSysInfo for more information.
pAddLTESysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
pGSMCallBarringSysInfo	• See nas_CallBarringSysInfo for more information.
pWCDMACallBarringSysInfo	• See nas_CallBarringSysInfo for more information.
pLTEVoiceSupportSysInfo	• Indicates voice support status on LTE. – 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

8.1135.2 Field Documentation

8.1135.2.1 nas_AddCDMASysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo

8.1135.2.2 nas_AddSysInfo∗ unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo

8.1135.2.3 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo

8.1135.2.4 uint16_t∗ unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo

8.1135.2.5 **nas_AddSysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo

8.1135.2.6 **nas_SrvStatusInfo**∗ unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo

8.1135.2.7 **nas_CDMASysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo

8.1135.2.8 **nas_CallBarringSysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo

8.1135.2.9 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo

8.1135.2.10 **nas_GSMSrvStatusInfo**∗ unpack_nas_SLQSGetSysInfo_t::pGSMSrvStatusInfo

8.1135.2.11 **nas_GSMSysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pGSMSysInfo

8.1135.2.12 **nas_SrvStatusInfo**∗ unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo

8.1135.2.13 **nas_HDRSysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo

8.1135.2.14 **nas_GSMSrvStatusInfo**∗ unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo

8.1135.2.15 **nas_LTESysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pLTESysInfo

8.1135.2.16 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo

8.1135.2.17 **nas_CallBarringSysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo

8.1135.2.18 uint8_t∗ unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo

8.1135.2.19 **nas_GSMSrvStatusInfo**∗ unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo

8.1135.2.20 **nas_WCDMASysInfo**∗ unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo

8.1136 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

8.1136.1 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

8.1136.2 Field Documentation

8.1136.2.1 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

8.1136.2.2 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

8.1136.2.3 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

8.1136.2.4 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

8.1136.2.5 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref

8.1136.2.6 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

8.1136.2.7 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

8.1136.2.8 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

8.1136.2.9 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

8.1137 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

8.1137.1 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

8.1137.2 Field Documentation

8.1137.2.1 nas_CDMAChannel* unpack_nas_SLQSNasGet3GPP2Subscription_t::pCDMAChannel

8.1137.2.2 nas_dirNum* unpack_nas_SLQSNasGet3GPP2Subscription_t::pDirNum

8.1137.2.3 nas_homeSIDNID* unpack_nas_SLQSNasGet3GPP2Subscription_t::pHomeSIDNID

8.1137.2.4 nas_minBasedIMSI* unpack_nas_SLQSNasGet3GPP2Subscription_t::pMinBasedIMSI

8.1137.2.5 nas_namName* unpack_nas_SLQSNasGet3GPP2Subscription_t::pNAMNameInfo

8.1137.2.6 nas_trueIMSI* unpack_nas_SLQSNasGet3GPP2Subscription_t::pTrueIMSI

8.1138 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

8.1138.1 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-NeighboringGSM	• See nas_LTEInfoNeighboringGSM for more information.
pLTEInfo-NeighboringWCDMA	• See nas_LTEInfoNeighboringWCDMA for more information.
pUMTSCellID	• Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
pWCDMAInfoLTENeighborCell	• See nas_WCDMAInfoLTENeighborCell for more information.

8.1138.2 Field Documentation

8.1138.2.1 nas_CDMAInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo

8.1138.2.2 nas_GERANInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo

8.1138.2.3 nas_LTEInfoInterfreq∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq

8.1138.2.4 nas_LTEInfoIntrafreq∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq

8.1138.2.5 nas_LTEInfoNeighboringGSM∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM

8.1138.2.6 nas_LTEInfoNeighboringWCDMA∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA

8.1138.2.7 uint32_t∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID

8.1138.2.8 nas_UMTSInfo∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo

8.1138.2.9 nas_WCDMAInfoLTENeighborCell∗ unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

8.1139 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference

Data Fields

- uint8_t ∗ pColorCode

8.1139.1 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

8.1139.2 Field Documentation

8.1139.2.1 uint8_t∗ unpack_nas_SLQSNasGetHDRColorCode_t::pColorCode

8.1140 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

Data Fields

- cdmaSSInfo CDMASSInfo
- hdrSSInfo HDRSSInfo
- int8_t GSMSSInfo
- cdmaSSInfo WCDMASSInfo
- lteSSInfo LTESSInfo

8.1140.1 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

8.1140.2 Field Documentation

8.1140.2.1 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo

8.1140.2.2 int8_t unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo

8.1140.2.3 hdrSSInfo unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo

8.1140.2.4 lteSSInfo unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo

8.1140.2.5 cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo

8.1141 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- nas_rxInfo * pRXChain0Info
- nas_rxInfo * pRXChain1Info
- nas_txInfo * pTXInfo

8.1141.1 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.

8.1141.2 Field Documentation

8.1141.2.1 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain0Info

8.1141.2.2 nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain1Info

8.1141.2.3 nas_txInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pTXInfo

8.1142 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference

Data Fields

- nas_UniversalTime universalTime
- uint8_t * pTimeZone
- uint8_t * pDayltSavAdj
- uint8_t * pRadioInterface

8.1142.1 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

8.1142.2 Field Documentation

8.1142.2.1 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

8.1142.2.2 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

8.1142.2.3 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

8.1142.2.4 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

8.1143 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

8.1143.1 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.

8.1143.2 Field Documentation

8.1143.2.1 cdmaSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo

8.1143.2.2 int8_t* unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo

8.1143.2.3 hdrSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo

8.1143.2.4 lteSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo

8.1143.2.5 int8_t* unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp

8.1143.2.6 tdscdmaSigInfoExt* unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt

8.1143.2.7 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo

8.1144 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference

Data Fields

- nas_wcdmaUARFCN ∗ pWcdmaUARFCN
- nas_lteEARFCN ∗ pLteEARFCN
- nas_ltePCI ∗ pLtePCI

8.1144.1 Detailed Description

Structure for Get Channel Lock unpack.

Parameters

pWcdmaUARFCN	[Optional] • See nas_wcdmaUARFCN for more information
pLteEARFCN	[Optional] • See nas_lteEARFCN for more information
pLtePCI	[Optional] • See nas_ltePCI for more information

8.1144.2 Field Documentation

8.1144.2.1 nas_lteEARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLteEARFCN

8.1144.2.2 nas_ltePCI∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pLtePCI

8.1144.2.3 nas_wcdmaUARFCN∗ unpack_nas_SLQSNASSwiGetChannelLock_t::pWcdmaUARFCN

8.1145 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

Data Fields

- nas_CommInfo commonInfo
- nas_LTEInfo ∗ pLTEInfo

8.1145.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

Parameters

commonInfo	(mandatory) • See nas_CommInfo for more information
pLTEInfo	(optional) • See nas_LTEInfo for more information

8.1145.2 Field Documentation

8.1145.2.1 nas_CommInfo unpack_nas_SLQSNasSwiModemStatus_t::commonInfo

8.1145.2.2 nas_LTEInfo* unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo

8.1146 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference

Data Fields

- NASQmiCbkNasSwiOTAMessageInd Info
- uint16_t Tlvresult

8.1146.1 Detailed Description

Parameters

Info	• Structure used to store all QMI Notification Info. – See NASQmiCbkNasSwiOTAMessageInd for more details
Tlvresult	• unpack result

8.1146.2 Field Documentation

8.1146.2.1 NASQmiCbkNasSwiOTAMessageInd unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info

8.1146.2.2 uint16_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult

8.1147 unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference

Data Fields

- char t3396_apn [101]
- uint8_t t3396_plmn_id [3]
- uint32_t t3396_val

8.1147.1 Detailed Description

Structure for Network Timer indication parameters.

Parameters

t3396_apn	• apn
t3396_plmn_id	• plmn id
t3396_val	• timer value

8.1147.2 Field Documentation

8.1147.2.1 char unpack_nas_SLQSNasTimerCallback_ind_t::t3396_apn[101]

8.1147.2.2 uint8_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_plmn_id[3]

8.1147.2.3 uint32_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_val

8.1148 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference

Data Fields

- NASQmiCbkNasSystemSelPrefInd Info
- uint16_t Tlvresult

8.1148.1 Detailed Description

Parameters

Info	• Structure used to store all QMI Notification Info. – See NASQmiCbkNasSystemSelPrefInd for more details
Tlvresult	• unpack result

8.1148.2 Field Documentation

8.1148.2.1 NASQmiCbkNasSystemSelPrefInd unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

8.1148.2.2 uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult

8.1149 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference

Data Fields

- uint16_t * pCurrentPersonality
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtocolSubtypeElement

8.1149.1 Detailed Description

Structure for HDRPersonality unpack.

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 nas_protocolSubtypeElement for more information.

8.1149.2 Field Documentation

8.1149.2.1 uint16_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pCurrentPersonality

8.1149.2.2 uint8_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pPersonalityListLength

8.1149.2.3 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRPersonality_t::pProtocolSubtypeElement

8.1150 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference

Data Fields

- uint16_t * pCurrentPrsnlty
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtoSubTypElmnt
- uint64_t * pAppSubType

8.1150.1 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,

8.1150.2 Field Documentation

8.1150.2.1 uint64_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pAppSubType

8.1150.2.2 uint16_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pCurrentPrsnlty

8.1150.2.3 uint8_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pPersonalityListLength

8.1150.2.4 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pProtoSubTypElmnt

8.1151 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference

Data Fields

- nas_DRCParams * pDRCParams
- uint8_t * pUATI
- nas_PilotSetData * pPilotSetData

8.1151.1 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.

8.1151.2 Field Documentation

8.1151.2.1 nas_DRCParams* unpack_nas_SLQSSwiGetHRPDStats_t::pDRCParams

8.1151.2.2 nas_PilotSetData* unpack_nas_SLQSSwiGetHRPDStats_t::pPilotSetData

8.1151.2.3 uint8_t* unpack_nas_SLQSSwiGetHRPDStats_t::pUATI

8.1152 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

Data Fields

- uint8_t ValidityCW0
- uint8_t CQIValueCW0
- uint8_t ValidityCW1
- uint8_t CQIValueCW1

8.1152.1 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

8.1152.2 Field Documentation

8.1152.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

8.1152.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

8.1152.2.3 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

8.1152.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

8.1153 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference

Data Fields

- nas_SccRxInfo ∗ pSccRxInfo

8.1153.1 Detailed Description

Parameters

pSccRxInfo	Secondary carrier Rx signal level info

8.1153.2 Field Documentation

8.1153.2.1 nas_SccRxInfo∗ unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo

8.1154 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference

Data Fields

- uint8_t ∗ pObjectVer
- nas_NetworkStat1x ∗ pNetworkStat1x
- nas_NetworkStatEVDO ∗ pNetworkStatEVDO
- nas_DeviceConfigDetail ∗ pDeviceConfigDetail
- nas_DataStatusDetail ∗ pDataStatusDetail

8.1154.1 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
pNetworkStatE-VDO	• See nas_NetworkStatEVDO for more information.
pDeviceConfig-Detail	• See nas_DeviceConfigDetail for more information.
pDataStatus-Detail	• See nas_DataStatusDetail for more information.

8.1154.2 Field Documentation

8.1154.2.1 nas_DataStatusDetail∗ unpack_nas_SLQSSwiNetworkDebug_t::pDataStatusDetail

8.1154.2.2 nas_DeviceConfigDetail∗ unpack_nas_SLQSSwiNetworkDebug_t::pDeviceConfigDetail

8.1154.2.3 nas_NetworkStat1x∗ unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStat1x

8.1154.2.4 nas_NetworkStatEVDO∗ unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStatEVDO

8.1154.2.5 uint8_t∗ unpack_nas_SLQSSwiNetworkDebug_t::pObjectVer

8.1155 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

8.1155.1 Detailed Description

Parameters

pCDMASrvStatusInfo	• See nas_SrvStatusInfo for more information.
pHDRSrvStatusInfo	• See nas_SrvStatusInfo for more information.
pGSMSrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pWCDMASrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pLTESrvStatusInfo	• See nas_GSMSrvStatusInfo for more information.
pCDMASysInfo	• See nas_CDMASysInfo for more information.
pHDRSysInfo	• See nas_HDRSysInfo for more information.
pGSMSysInfo	• See nas_GSMSysInfo for more information.
pWCDMASysInfo	• See nas_WCDMASysInfo for more information.
pLTESysInfo	• See nas_LTESysInfo for more information.
pAddCDMASysInfo	• See nas_AddCDMASysInfo for more information.
pAddHDRSysInfo	• System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
pAddGSMSysInfo	• See nas_AddSysInfo for more information.

pAddWCDMA-SysInfo	• See nas_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 nas_CallBarringSysInfo for more information.
pWCDMACall-BarringSysInfo	• See nas_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
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

8.1155.2 Field Documentation

8.1155.2.1 nas_AddCDMASysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo

8.1155.2.2 nas_AddSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSMSysInfo

8.1155.2.3 uint16_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDRSysInfo

8.1155.2.4 uint16_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTESysInfo

8.1155.2.5 nas_AddSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo

8.1155.2.6 nas_SrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo

8.1155.2.7 **nas_CDMASysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo

8.1155.2.8 **nas_CallBarringSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo

8.1155.2.9 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo

8.1155.2.10 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo

8.1155.2.11 **nas_GSMSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo

8.1155.2.12 **nas_SrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSrvStatusInfo

8.1155.2.13 **nas_HDRSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSysInfo

8.1155.2.14 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo

8.1155.2.15 **nas_LTESysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo

8.1155.2.16 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo

8.1155.2.17 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange

8.1155.2.18 **nas_CallBarringSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo

8.1155.2.19 uint8_t* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo

8.1155.2.20 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo

8.1155.2.21 **nas_WCDMASysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo

8.1156 unpack_omaDmConfigTlv_t Struct Reference

Data Fields

- uint8_t state
- uint8_t userInputReq
- uint16_t userInputTimeout
- uint16_t alertmsglength
- uint8_t alertmsg [256]

8.1156.1 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
userInputReq	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject

userInput-Timeout	• Timeout for user input in minutes. A value of 0 means no time-out
alertmsglength	• Length of Alert message string in bytes
alertmsg	• Alert message in UCS2 (Max 256 characters)

8.1156.2 Field Documentation

8.1156.2.1 uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]

8.1156.2.2 uint16_t unpack_omaDmConfigTlv_t::alertmsglength

8.1156.2.3 uint8_t unpack_omaDmConfigTlv_t::state

8.1156.2.4 uint8_t unpack_omaDmConfigTlv_t::userInputReq

8.1156.2.5 uint16_t unpack_omaDmConfigTlv_t::userInputTimeout

8.1157 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

8.1157.1 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
userInputReq	- Bit mask of available user inputs • 0x00 - No user input required.Informational indication • 0x01 - Accept • 0x02 - Reject
userInput-Timeout	• Timeout for user input in minutes. A value of 0 means no time-out
fwdloadsize	• The size (in bytes) of the firmware update package
fwloadComplete	• The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment.
update-CompleteStatus	• This field should be looked at only when the OMADM session is complete. • See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h for update complete status.
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

8.1157.2 Field Documentation

8.1157.2.1 uint8_t unpack_omaDmFotaTlv_t::description[256]

8.1157.2.2 uint16_t unpack_omaDmFotaTlv_t::descriptionlength

8.1157.2.3 uint32_t unpack_omaDmFotaTlv_t::fwdloadsize

8.1157.2.4 uint32_t unpack_omaDmFotaTlv_t::fwloadComplete

8.1157.2.5 uint16_t unpack_omaDmFotaTlv_t::namelength

8.1157.2.6 uint8_t unpack_omaDmFotaTlv_t::package_name[256]

8.1157.2.7 uint8_t unpack_omaDmFotaTlv_t::sessionType

8.1157.2.8 uint8_t unpack_omaDmFotaTlv_t::severity

8.1157.2.9 uint8_t unpack_omaDmFotaTlv_t::state

8.1157.2.10 uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus

8.1157.2.11 uint8_t unpack_omaDmFotaTlv_t::userInputReq

8.1157.2.12 uint16_t unpack_omaDmFotaTlv_t::userInputTimeout

8.1157.2.13 uint8_t unpack_omaDmFotaTlv_t::version[256]

8.1157.2.14 uint16_t unpack_omaDmFotaTlv_t::versionlength

8.1158 unpack_omaDmNotificationsTlv_t Struct Reference

Data Fields

- uint8_t notification
- uint16_t sessionStatus

8.1158.1 Detailed Description

This structure will hold the SwiOmaDmConfig session notification parameters information.

Parameters

notification	• 0x01 - GPS settings change • 0x02 - Device reset • 0x03 - Device factory reset • 0x04 - CI-DC Session start • 0x05 - CI-DC Session end • 0x06 - CI-PRL Session start • 0x07 - CI-PRL Session end • 0x08 - CI-FUMO Session start • 0x09 - CI-FUMO session end • 0x0A - HFA-DC Session start • 0x0B - HFA-DC Session end • 0x0C - HFA-PRL Session start • 0x0D - HFA-PRL Session end • 0x0E - HFA-FUMO Session start • 0x0F - HFA-FUMO session end • 0x10 - NI Session start • 0x11 - NI session end
sessionStatus	• This field will set to the session status for notifications that occur at the end of a session, zero for all other notifications • See qaGobiApiTableSwiOMADMSessionStatus.h for session completion code.

8.1158.2 Field Documentation

8.1158.2.1 uint8_t unpack_omaDmNotificationsTlv_t::notification

8.1158.2.2 uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus

8.1159 unpack_pds_GetPDSDefaults_t Struct Reference

Data Fields

- uint32_t * pOperation
- uint8_t * pTimeout
- uint32_t * pInterval
- uint32_t * pAccuracy

8.1159.1 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)

8.1159.2 Field Documentation

8.1159.2.1 uint32_t* unpack_pds_GetPDSDefaults_t::pAccuracy

8.1159.2.2 uint32_t* unpack_pds_GetPDSDefaults_t::pInterval

8.1159.2.3 uint32_t* unpack_pds_GetPDSDefaults_t::pOperation

8.1159.2.4 uint8_t* unpack_pds_GetPDSDefaults_t::pTimeout

8.1160 unpack_pds_GetPDSState_t Struct Reference

Data Fields

- uint32_t * pEnabledStatus
- uint32_t * pTrackingStatus

8.1160.1 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

8.1160.2 Field Documentation

8.1160.2.1 uint32_t* unpack_pds_GetPDSState_t::pEnabledStatus

8.1160.2.2 uint32_t* unpack_pds_GetPDSState_t::pTrackingStatus

8.1161 unpack_pds_GetPortAutomaticTracking_t Struct Reference

Data Fields

- uint32_t * pbAuto

8.1161.1 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

8.1161.2 Field Documentation

8.1161.2.1 uint32_t* unpack_pds_GetPortAutomaticTracking_t::pbAuto

8.1162 unpack_pds_GetServiceAutomaticTracking_t Struct Reference

Data Fields

- uint32_t * pbAuto

8.1162.1 Detailed Description

Structure contain the parameter for automatic tracking state.

Parameters

pbAuto	• Automatic tracking session started for service – 0x00 - Disabled – 0x01 - Enabled

8.1162.2 Field Documentation

8.1162.2.1 uint32_t* unpack_pds_GetServiceAutomaticTracking_t::pbAuto

8.1163 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference

Data Fields

- uint32_t * pbEnabled
- uint16_t * pInterval

8.1163.1 Detailed Description

Structure contain the parameter of automatic tracking configuration for the NMEA COM port.

Parameters

pbEnabled	• Automatic XTRA download status – 0 - Disabled – 1 - Enabled
pInterval	• Interval (hours) between XTRA downloads

8.1163.2 Field Documentation

8.1163.2.1 uint32_t* unpack_pds_GetXTRAAutomaticDownload_t::pbEnabled

8.1163.2.2 uint16_t* unpack_pds_GetXTRAAutomaticDownload_t::pInterval

8.1164 unpack_pds_GetXTRANetwork_t Struct Reference

Data Fields

- uint32_t * pPreference

8.1164.1 Detailed Description

Structure contain the parameter for XTRA WWAN network preference

Parameters

pPreference	• 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

8.1164.2 Field Documentation

8.1164.2.1 uint32_t* unpack_pds_GetXTRANetwork_t::pPreference

8.1165 unpack_pds_GetXTRAValidity_t Struct Reference

Data Fields

- uint16_t * pGPSWeek
- uint16_t * pGPSWeekOffset
- uint16_t * pDuration

8.1165.1 Detailed Description

Structure contain the parameter for XTRA database validity period

Parameters

pGPSWeek	• Starting GPS week of validity period
pGPSWeek-Offset	• Starting GPS week offset (minutes) of validity period
pDuration	• Length of validity period (hours)

8.1165.2 Field Documentation

8.1165.2.1 uint16_t* unpack_pds_GetXTRAValidity_t::pDuration

8.1165.2.2 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeek

8.1165.2.3 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeekOffset

8.1166 unpack_pds_SetEventReport_Ind_t Struct Reference

Data Fields

- uint8_t has_PositionDataNMEA
- char PositionDataNMEA [200]
- uint8_t has_dLongitude
- double dLongitude
- uint8_t has_dLatitude
- double dLatitude
- uint8_t has_SessionStatus
- uint8_t SessionStatus
- uint8_t has_posSrc
- uint32_t posSrc

8.1166.1 Detailed Description

Structure contain the parameter for the Set event report Indication. Please check has_<Param_Name> field for presence of optional parameters

Parameters

PositionDataNM-EA	• Position Data NMEA. • String containing the position data in NMEA sentence format (maximum 200 bytes).

dLongitude	• Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of the Greenwich Meridian. • Units - Decimal degrees • Range - -180 to +180 • Note - Value is in double float format
dLatitude	• Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator. • Units: Decimal degrees • Range: -90 to +90 • Note - Value is in double float format
session_status	• Position Session Status • Status of current session. • Values – 0x00 - Success – 0x01 - In progress – 0x02 - General failure – 0x03 - Timeout – 0x04 - User ended the session – 0x05 - Bad parameter – 0x06 - Phone is offline – 0x07 - Engine is locked – 0x08 - E911 session in progress • Note - This TLV is always sent while a fix is being generated or if it halts due to an error. The Parsed Position Data TLV (0x13) is included if the session status is Success (0x00) or In Progress (0x01); otherwise this TLV is passed alone.
posSrc	• Position Source • This TLV is always sent with the Parsed Position Data TLV (0x13) when the latitude/longitude is marked as valid. Source of the position (bitmask). • Values – 0x00000001 - GPS – 0x00000002 – Cell ID – 0x00000004 – GLONASS – 0x00000008 – Network – 0x00000010 – External position injection

8.1166.2 Field Documentation

8.1166.2.1 double unpack_pds_SetEventReport_Ind_t::dLatitude

8.1166.2.2 double unpack_pds_SetEventReport_Ind_t::dLongitude

8.1166.2.3 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLatitude

8.1166.2.4 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLongitude

8.1166.2.5 uint8_t unpack_pds_SetEventReport_Ind_t::has_PositionDataNMEA

8.1166.2.6 uint8_t unpack_pds_SetEventReport_Ind_t::has_posSrc

8.1166.2.7 uint8_t unpack_pds_SetEventReport_Ind_t::has_SessionStatus

8.1166.2.8 char unpack_pds_SetEventReport_Ind_t::PositionDataNMEA[200]

8.1166.2.9 uint32_t unpack_pds_SetEventReport_Ind_t::posSrc

8.1166.2.10 uint8_t unpack_pds_SetEventReport_Ind_t::SessionStatus

8.1167 unpack_pds_SetPdsState_Ind_t Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t EnabledStatus
- uint32_t TrackingStatus

8.1167.1 Detailed Description

Structure contain the parameter for the Set PDS State Indication.

Parameters

EnabledStatus	• GPS service state. • Values – 0x00 - Disable – 0x01 - Enable
TrackingStatus	• Tracking session state. • Values – 0x00 - Unknown – 0x01 - Inactive – 0x02 - Active

8.1167.2 Field Documentation

8.1167.2.1 uint32_t unpack_pds_SetPdsState_Ind_t::EnabledStatus

8.1167.2.2 uint8_t unpack_pds_SetPdsState_Ind_t::TlvPresent

8.1167.2.3 uint32_t unpack_pds_SetPdsState_Ind_t::TrackingStatus

8.1168 unpack_pds_SLQSGetAGPSConfig_t Struct Reference

Data Fields

- uint32_t * pServerAddress
- uint32_t * pServerPort
- uint8_t * pServerURL
- uint8_t * pServerURLLength

8.1168.1 Detailed Description

Structure contain the parameter for PDS AGPS (MS-based) configuration unpack.

Parameters

pServerAddress	• IPv4 address of AGPS server. "0" if not set
pServerPort	• Port number of AGPS server. "0" if not set
pServerURL	• URL of the AGPS server. "0" if not set
pServerURL-Length	• URL length of AGPS server. "0" if not set

8.1168.2 Field Documentation

8.1168.2.1 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerAddress

8.1168.2.2 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerPort

8.1168.2.3 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURL

8.1168.2.4 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURLLength

8.1169 unpack_pds_SLQSGetGPSStateInfo_t Struct Reference

Data Fields

- uint8_t EngineState
- uint32_t ValidMask
- uint64_t Latitude
- uint64_t Longitude
- uint32_t HorizontalUncertainty
- uint32_t Altitude
- uint32_t VerticalUncertainty
- uint32_t TimeStmp_tow_ms
- uint16_t TimeStmp_gps_week
- uint32_t Time_uncert_ms
- uint8_t Iono_valid
- uint32_t gps_ephemeris_sv_msk
- uint32_t gps_almanac_sv_msk
- uint32_t gps_health_sv_msk

- uint32_t gps_visible_sv_msk
- uint32_t glo_ephemeris_sv_msk
- uint32_t glo_almanac_sv_msk
- uint32_t glo_health_sv_msk
- uint32_t glo_visible_sv_msk
- uint32_t sbas_ephemeris_sv_msk
- uint32_t sbas_almanac_sv_msk
- uint32_t sbas_health_sv_msk
- uint32_t sbas_visible_sv_msk
- uint16_t xtra_start_gps_week
- uint16_t xtra_start_gps_minutes
- uint16_t xtra_valid_duration_hours

8.1169.1 Detailed Description

Structure contain the parameter for 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

8.1169.2 Field Documentation

8.1169.2.1 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Altitude

8.1169.2.2 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::EngineState

8.1169.2.3 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_almanac_sv_msk

8.1169.2.4 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_ephemeris_sv_msk

8.1169.2.5 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_health_sv_msk

8.1169.2.6 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_visible_sv_msk

8.1169.2.7 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_almanac_sv_msk

8.1169.2.8 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_ephemeris_sv_msk

8.1169.2.9 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_health_sv_msk

8.1169.2.10 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_visible_sv_msk

8.1169.2.11 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::HorizontalUncertainty

8.1169.2.12 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::Iono_valid

8.1169.2.13 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Latitude

8.1169.2.14 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Longitude

8.1169.2.15 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_almanac_sv_msk

8.1169.2.16 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_ephemeris_sv_msk

8.1169.2.17 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_health_sv_msk

8.1169.2.18 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_visible_sv_msk

8.1169.2.19 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Time_uncert_ms

8.1169.2.20 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_gps_week

8.1169.2.21 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_tow_ms

8.1169.2.22 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::ValidMask

8.1169.2.23 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::VerticalUncertainty

8.1169.2.24 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_minutes

8.1169.2.25 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_week

8.1169.2.26 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_valid_duration_hours

8.1170 unpack_qmi_t Struct Reference

Data Fields

- enum msgtype type
- uint16_t msgid
- uint16_t xid

8.1170.1 Detailed Description

qmi response context

Parameters

out	*type*	message type
out	*msgid*	message id
out	*xid*	transaction id

8.1170.2 Field Documentation

8.1170.2.1 uint16_t unpack_qmi_t::msgid

8.1170.2.2 enum msgtype unpack_qmi_t::type

8.1170.2.3 uint16_t unpack_qmi_t::xid

8.1171 unpack_qos_dataRate_t Struct Reference

Data Fields

- uint32_t dataRateMax
- uint32_t guaranteedRate

8.1171.1 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)

8.1171.2 Field Documentation

8.1171.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

8.1171.2.2 uint32_t unpack_qos_dataRate_t::guaranteedRate

8.1172 unpack_qos_IPv4Addr_t Struct Reference

Data Fields

- uint32_t addr
- uint32_t subnetMask

8.1172.1 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

8.1172.2 Field Documentation

8.1172.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

8.1172.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

8.1173 unpack_qos_IPv6Addr_t Struct Reference

Data Fields

- uint8_t addr [16]
- uint8_t prefixLen

8.1173.1 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.

8.1173.2 Field Documentation

8.1173.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

8.1173.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

8.1174 unpack_qos_IPv6TrafCls_t Struct Reference

Data Fields

- uint8_t val
- uint8_t mask

8.1174.1 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

8.1174.2 Field Documentation

8.1174.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

8.1174.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

8.1175 unpack_qos_pktErrRate_t Struct Reference

Data Fields

- uint16_t multiplier
- uint16_t exponent

8.1175.1 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)

8.1175.2 Field Documentation

8.1175.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

8.1175.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

8.1176 unpack_qos_Port_t Struct Reference

Data Fields

- uint16_t port
- uint16_t range

8.1176.1 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

8.1176.2 Field Documentation

8.1176.2.1 uint16_t unpack_qos_Port_t::port

8.1176.2.2 uint16_t unpack_qos_Port_t::range

8.1177 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

8.1177.1 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.

8.1177.2 Field Documentation

8.1177.2.1 uint8_t unpack_qos_QosFlowInfo_t::BearerID

8.1177.2.2 uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available

8.1177.2.3 uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available

8.1177.2.4 uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters

8.1177.2.5 uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters

8.1177.2.6 unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState

8.1177.2.7 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]

8.1177.2.8 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted

8.1177.2.9 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]

8.1177.2.10 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted

8.1178 unpack_qos_QosFlowInfoState_t Struct Reference

Data Fields

- uint32_t id
- uint8_t isNewFlow
- uint8_t state

8.1178.1 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

8.1178.2 Field Documentation

8.1178.2.1 uint32_t unpack_qos_QosFlowInfoState_t::id

8.1178.2.2 uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow

8.1178.2.3 uint8_t unpack_qos_QosFlowInfoState_t::state

8.1179 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference

Data Fields

- uint8_t NWQoSStatus

8.1179.1 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

8.1179.2 Field Documentation

8.1179.2.1 uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus

8.1180 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

8.1180.1 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

8.1180.2 Field Documentation

8.1180.2.1 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl

8.1180.2.2 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext

8.1180.2.3 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext2

8.1180.2.4 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul

8.1180.2.5 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext

8.1180.2.6 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext2

8.1180.2.7 uint32_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.1181 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]

8.1181.1 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

8.1181.2 Field Documentation

8.1181.2.1 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId

8.1181.2.2 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow

8.1181.2.3 unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]

8.1181.2.4 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes

8.1181.2.5 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt

8.1181.2.6 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes

8.1181.2.7 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp

8.1181.2.8 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt

8.1181.2.9 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp

8.1182 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

Data Fields

- uint8_t NumFlows
- unpack_qos_QosFlowInfo_t QosFlowInfo [8]

8.1182.1 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

8.1182.2 Field Documentation

8.1182.2.1 uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows

8.1182.2.2 unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]

8.1183 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference

Data Fields

- uint8_t status

8.1183.1 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

8.1183.2 Field Documentation

8.1183.2.1 uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status

8.1184 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference

Data Fields

- uint16_t event

8.1184.1 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

8.1184.2 Field Documentation

8.1184.2.1 uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event

8.1185 unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference

Data Fields

- uint32_t id
- uint8_t status
- uint8_t event
- uint8_t reason

8.1185.1 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

8.1185.2 Field Documentation

8.1185.2.1 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event

8.1185.2.2 uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id

8.1185.2.3 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason

8.1185.2.4 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status

8.1186 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

8.1186.1 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

8.1186.2 Field Documentation

8.1186.2.1 uint32_t unpack_qos_swiQosFilter_t::EspSpi

8.1186.2.2 uint16_t unpack_qos_swiQosFilter_t::Id

8.1186.2.3 uint8_t unpack_qos_swiQosFilter_t::index

8.1186.2.4 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr

8.1186.2.5 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr

8.1186.2.6 unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos

8.1186.2.7 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr

8.1186.2.8 uint32_t unpack_qos_swiQosFilter_t::IPv6Label

8.1186.2.9 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr

8.1186.2.10 unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls

8.1186.2.11 uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available

8.1186.2.12 uint8_t unpack_qos_swiQosFilter_t::is_Id_Available

8.1186.2.13 uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available

8.1186.2.14 uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available

8.1186.2.15 uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available

8.1186.2.16 uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available

8.1186.2.17 uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available

8.1186.2.18 uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available

8.1186.2.19 uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available

8.1186.2.20 uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available

8.1186.2.21 uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available

8.1186.2.22 uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available

8.1186.2.23 uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available

8.1186.2.24 uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available

8.1186.2.25 uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available

8.1186.2.26 uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available

8.1186.2.27 uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available

8.1186.2.28 uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto

8.1186.2.29 uint16_t unpack_qos_swiQosFilter_t::Precedence

8.1186.2.30 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort

8.1186.2.31 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort

8.1186.2.32 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort

8.1186.2.33 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort

8.1186.2.34 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort

8.1186.2.35 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort

8.1186.2.36 uint8_t unpack_qos_swiQosFilter_t::version

8.1187 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

8.1187.1 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

8.1187.2 Field Documentation

8.1187.2.1 unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate

8.1187.2.2 uint8_t unpack_qos_swiQosFlow_t::index

8.1187.2.3 uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available

8.1187.2.4 uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available

8.1187.2.5 uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available

8.1187.2.6 uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available

8.1187.2.7 uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available

8.1187.2.8 uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available

8.1187.2.9 uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available

8.1187.2.10 uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available

8.1187.2.11 uint8_t unpack_qos_swiQosFlow_t::is_TokenBucket_Available

8.1187.2.12 uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available

8.1187.2.13 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available

8.1187.2.14 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available

8.1187.2.15 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available

8.1187.2.16 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available

8.1187.2.17 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available

8.1187.2.18 uint32_t unpack_qos_swiQosFlow_t::Jitter

8.1187.2.19 uint32_t unpack_qos_swiQosFlow_t::Latency

8.1187.2.20 uint8_t unpack_qos_swiQosFlow_t::LteQci

8.1187.2.21 uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz

8.1187.2.22 uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz

8.1187.2.23 unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate

8.1187.2.24 uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2

8.1187.2.25 unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket

8.1187.2.26 uint8_t unpack_qos_swiQosFlow_t::TrafficClass

8.1187.2.27 uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri

8.1187.2.28 uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn

8.1187.2.29 uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER

8.1187.2.30 uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd

8.1187.2.31 uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri

8.1188 unpack_qos_tokenBucket_t Struct Reference

Data Fields

- uint32_t peakRate
- uint32_t tokenRate
- uint32_t bucketSz

8.1188.1 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

8.1188.2 Field Documentation

8.1188.2.1 uint32_t unpack_qos_tokenBucket_t::bucketSz

8.1188.2.2 uint32_t unpack_qos_tokenBucket_t::peakRate

8.1188.2.3 uint32_t unpack_qos_tokenBucket_t::tokenRate

8.1189 unpack_qos_Tos_t Struct Reference

Data Fields

- uint8_t val
- uint8_t mask

8.1189.1 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.

8.1189.2 Field Documentation

8.1189.2.1 uint8_t unpack_qos_Tos_t::mask

8.1189.2.2 uint8_t unpack_qos_Tos_t::val

8.1190 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

8.1190.1 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

8.1190.2 Field Documentation

8.1190.2.1 uint32_t unpack_QosFlowStat_t::bearerId

8.1190.2.2 uint64_t unpack_QosFlowStat_t::tx_bytes

8.1190.2.3 uint64_t unpack_QosFlowStat_t::tx_bytes_drp

8.1190.2.4 uint32_t unpack_QosFlowStat_t::tx_pkt

8.1190.2.5 uint32_t unpack_QosFlowStat_t::tx_pkt_drp

8.1191 unpack_rms_GetSMSWake_t Struct Reference

Data Fields

- uint32_t enabled
- uint32_t wake_mask

8.1191.1 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)

8.1191.2 Field Documentation

8.1191.2.1 uint32_t unpack_rms_GetSMSWake_t::enabled

8.1191.2.2 uint32_t unpack_rms_GetSMSWake_t::wake_mask

8.1192 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

8.1192.1 Detailed Description

WDS packet RM Transfer Statistics data structure for individual session

Parameters

TxOkConutTlv	• Tx Ok Packet Tlv Value.
RxOkConutTlv	• Rx Ok Packet Tlv Value.
TxOkByteCount-Tlv	• Tx Ok Byte Count Packet Tlv Value.
RxOkByteCount-Tlv	• Rx Ok Byte Count Packet Tlv Value.
TxDropConutTlv	• Tx Drop Count Packet Tlv Value.
RxDropConutTlv	• Rx Drop Count Packet Tlv Value.

8.1192.2 Field Documentation

8.1192.2.1 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv

8.1192.2.2 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::RxOkByteCountTlv

8.1192.2.3 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxOkConutTlv

8.1192.2.4 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxDropConutTlv

8.1192.2.5 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::TxOkByteCountTlv

8.1192.2.6 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxOkConutTlv

8.1193 unpack_sar_SLQSGetRfSarState_t Struct Reference

Data Fields

- uint32_t * pSarRFState

8.1193.1 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

8.1193.2 Field Documentation

8.1193.2.1 uint32_t* unpack_sar_SLQSGetRfSarState_t::pSarRFState

8.1194 unpack_sms_GetSMSCAddress_t Struct Reference

Data Fields

- uint8_t addressSize
- uint8_t * pSMSCAddress
- uint8_t typeSize
- uint8_t * pSMSCType

8.1194.1 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.

8.1194.2 Field Documentation

8.1194.2.1 uint8_t unpack_sms_GetSMSCAddress_t::addressSize

8.1194.2.2 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCAddress

8.1194.2.3 uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCType

8.1194.2.4 uint8_t unpack_sms_GetSMSCAddress_t::typeSize

8.1195 unpack_sms_SaveSMS_t Struct Reference

Data Fields

- uint32_t * pMessageIndex

8.1195.1 Detailed Description

Parameters

pMessageIndex	• The message index assigned by the device

8.1195.2 Field Documentation

8.1195.2.1 uint32_t* unpack_sms_SaveSMS_t::pMessageIndex

8.1196 unpack_sms_SendSMS_t Struct Reference

Data Fields

- uint16_t messageID
- uint32_t messageFailureCode

8.1196.1 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

8.1196.2 Field Documentation

8.1196.2.1 uint32_t unpack_sms_SendSMS_t::messageFailureCode

8.1196.2.2 uint16_t unpack_sms_SendSMS_t::messageID

8.1197 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

8.1197.1 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

8.1197.2 Field Documentation

8.1197.2.1 struct eTWSPLMNInfoTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv

8.1197.2.2 struct sMSEtwsMessageTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv

8.1197.2.3 struct sMSOnIMSTlv unpack_sms_SetNewSMSCallback_ind_t::IMSTlv

8.1197.2.4 struct messageModeTlv unpack_sms_SetNewSMSCallback_ind_t::MMTlv

8.1197.2.5 struct newMTMessageTlv unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv

8.1197.2.6 struct sMSCAddressTlv unpack_sms_SetNewSMSCallback_ind_t::SMSCTlv

8.1197.2.7 struct transferRouteMessageTlv unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv

8.1198 unpack_sms_SetNewSMSCallback_t Struct Reference

8.1199 unpack_sms_SLQSDeleteSMS_t Struct Reference

8.1200 unpack_sms_SLQSGetIndicationRegister_t Struct Reference

Data Fields

- sms_getIndicationReg * pGetIndicationRegInfo

8.1200.1 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

8.1200.2 Field Documentation

8.1200.2.1 sms_getIndicationReg* unpack_sms_SLQSGetIndicationRegister_t::pGetIndicationRegInfo

8.1201 unpack_sms_SLQSGetMessageWaiting_t Struct Reference

Data Fields

- sms_getMsgWaitingInfo * pGetMsgWaitingInfoResp

8.1201.1 Detailed Description

Structure contain Parameter that provide information about the message waiting information.

Parameters

pGetMsg-WaitingInfoResp	• Pointer to structure of getMsgWaitingInfoResp – See sms_getMsgWaitingInfo for more information

8.1201.2 Field Documentation

8.1201.2.1 sms_getMsgWaitingInfo* unpack_sms_SLQSGetMessageWaiting_t::pGetMsgWaitingInfoResp

8.1202 unpack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t messageTag
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t message [2048]

8.1202.1 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

8.1202.2 Field Documentation

8.1202.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[2048]

8.1202.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

8.1202.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

8.1202.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

8.1203 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

Data Fields

- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig

- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

8.1203.1 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).

8.1203.2 Field Documentation

8.1203.2.1 sms_qaQmi3GPPBroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pBroadcastConfig

8.1203.2.2 sms_qaQmi3GPP2BroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pCDMABroadcast-Config

8.1204 unpack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t messageListSize
- qmiSmsMessageList messageList [255]

8.1204.1 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

8.1204.2 Field Documentation

8.1204.2.1 qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]

8.1204.2.2 uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize

8.1205 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference

Data Fields

- sms_getTransLayerInfo * pGetTransLayerInfo

8.1205.1 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

8.1205.2 Field Documentation

8.1205.2.1 sms_getTransLayerInfo* unpack_sms_SLQSGetTransLayerInfo_t::pGetTransLayerInfo

8.1206 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference

Data Fields

- sms_getTransNWRegInfo * pGetTransNWRegInfo

8.1206.1 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

8.1206.2 Field Documentation

8.1206.2.1 sms_getTransNWRegInfo* unpack_sms_SLQSGetTransNWRegInfo_t::pGetTransNWRegInfo

8.1207 unpack_sms_SLQSModifySMSStatus_t Struct Reference

8.1208 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

Data Fields

- sms_maxStorageSizeResp * pMaxStorageSizeResp

8.1208.1 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

8.1208.2 Field Documentation

8.1208.2.1 sms_maxStorageSizeResp∗ unpack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeResp

8.1209 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference

Data Fields

- sms_msgProtocolResp ∗ pMessageProtocol

8.1209.1 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

8.1209.2 Field Documentation

8.1209.2.1 sms_msgProtocolResp∗ unpack_sms_SLQSSmsGetMessageProtocol_t::pMessageProtocol

8.1210 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference

Data Fields

- uint32_t ∗ pSmsStorage

8.1210.1 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

8.1210.2 Field Documentation

8.1210.2.1 uint32_t* unpack_sms_SLQSSwiGetSMSStorage_t::pSmsStorage

8.1211 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference

Data Fields

- uint8_t storageType
- uint8_t messageMode

8.1211.1 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)

8.1211.2 Field Documentation

8.1211.2.1 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode

8.1211.2.2 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType

8.1212 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

8.1212.1 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

8.1212.2 Field Documentation

8.1212.2.1 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate

8.1212.2.2 int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported

8.1212.2.3 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type

8.1212.2.4 int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported

8.1212.2.5 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function

8.1212.2.6 int unpack_swiloc_SwiLocGetAutoStart_t::function_reported

8.1212.2.7 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist

8.1212.2.8 int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported

8.1212.2.9 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time

8.1212.2.10 int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported

8.1213 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference

Data Fields

- uint32_t eventType
- unpack_omaDmFotaTlv_t SessionInfoFota
- unpack_omaDmConfigTlv_t SessionInfoConfig
- unpack_omaDmNotificationsTlv_t SessionInfoNotification

8.1213.1 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

8.1213.2 Field Documentation

8.1213.2.1 uint32_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::eventType

8.1213.2.2 unpack_omaDmConfigTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoConfig

8.1213.2.3 unpack_omaDmFotaTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoFota

8.1213.2.4 unpack_omaDmNotificationsTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfo-Notification

8.1214 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

8.1214.1 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

8.1214.2 Field Documentation

8.1214.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

8.1214.2.2 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength

8.1214.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength

8.1214.2.4 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]

8.1214.2.5 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgName[255]

8.1214.2.6 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgNameLength

8.1214.2.7 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::RetryCount

8.1214.2.8 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionState

8.1214.2.9 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

8.1214.2.10 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Severity

8.1214.2.11 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Source[255]

8.1214.2.12 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SourceLength

8.1214.2.13 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Status

8.1214.2.14 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Time[255]

8.1214.2.15 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::TimeLength

8.1214.2.16 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus

8.1215 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference

Data Fields

- uint32_t OMADMEnabled
- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t Autosdm

- uint8_t FwAutoCheck

8.1215.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

Parameters

out	*OMADMEnabled*	• 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.
out	*FOTAdownload*	• 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.

Parameters

out	*FOTAUpdate*	• 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.
out	*Autosdm*	• 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.

out	*FwAutoCheck*	• 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.

8.1215.2 Field Documentation

8.1215.2.1 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::Autosdm

8.1215.2.2 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAdownload

8.1215.2.3 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAUpdate

8.1215.2.4 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FwAutoCheck

8.1215.2.5 uint32_t unpack_swioma_SLQSOMADMGetSettings_t::OMADMEnabled

8.1216 unpack_swioma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t FwAvailability

8.1216.1 Detailed Description

Structure that contains the responses for OMA start session command

Parameters

out	*FwAvailability*	• 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

8.1216.2 Field Documentation

8.1216.2.1 uint32_t unpack_swioma_SLQSOMADMStartSession_t::FwAvailability

8.1217 unpack_uim_ChangePin_t Struct Reference

Data Fields

- uim_remainingRetries * pRemainingRetries
- uim_encryptedPIN1 * pEncryptedPIN1

- uint32_t * pIndicationToken
- uint16_t Tlvresult

8.1217.1 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.

8.1217.2 Field Documentation

8.1217.2.1 uim_encryptedPIN1* unpack_uim_ChangePin_t::pEncryptedPIN1

8.1217.2.2 uint32_t* unpack_uim_ChangePin_t::pIndicationToken

8.1217.2.3 uim_remainingRetries* unpack_uim_ChangePin_t::pRemainingRetries

8.1217.2.4 uint16_t unpack_uim_ChangePin_t::Tlvresult

8.1218 unpack_uim_GetCardStatus_t Struct Reference

Data Fields

- uim_cardStatus * pCardStatus
- uim_hotSwapStatus * pHotSwapStatus
- uint16_t Tlvresult

8.1218.1 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.

8.1218.2 Field Documentation

8.1218.2.1 uim_cardStatus* unpack_uim_GetCardStatus_t::pCardStatus

8.1218.2.2 uim_hotSwapStatus* unpack_uim_GetCardStatus_t::pHotSwapStatus

8.1218.2.3 uint16_t unpack_uim_GetCardStatus_t::Tlvresult

8.1219 unpack_uim_ReadTransparent_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_readResult * pReadResult
- uint32_t * pIndicationToken
- uint8_t * pEncryptedData
- uint16_t Tlvresult

8.1219.1 Detailed Description

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

Parameters

pCardResult	• See uim_cardResult for more information.
pReadResult	• See uim_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.

8.1219.2 Field Documentation

8.1219.2.1 uim_cardResult* unpack_uim_ReadTransparent_t::pCardResult

8.1219.2.2 uint8_t* unpack_uim_ReadTransparent_t::pEncryptedData

8.1219.2.3 uint32_t* unpack_uim_ReadTransparent_t::pIndicationToken

8.1219.2.4 uim_readResult* unpack_uim_ReadTransparent_t::pReadResult

8.1219.2.5 uint16_t unpack_uim_ReadTransparent_t::Tlvresult

8.1220 unpack_uim_SetPinProtection_t Struct Reference

Data Fields

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

8.1220.1 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.

8.1220.2 Field Documentation

8.1220.2.1 uim_encryptedPIN1* unpack_uim_SetPinProtection_t::pEncryptedPIN1

8.1220.2.2 uint32_t* unpack_uim_SetPinProtection_t::pIndicationToken

8.1220.2.3 uim_remainingRetries* unpack_uim_SetPinProtection_t::pRemainingRetries

8.1220.2.4 uint16_t unpack_uim_SetPinProtection_t::Tlvresult

8.1221 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference

Data Fields

- slots_t slotsstatusChange
- uint8_t bNumberOfPhySlots

8.1221.1 Detailed Description

Structure consist of card status params

Parameters

slotsstatus-Change	• See slot_t for more information
bNumberOfPhy-Slots	• Number of Physical Slot(s)

8.1221.2 Field Documentation

8.1221.2.1 uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots

8.1221.2.2 slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange

8.1222 unpack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_authenticateResult * pAuthenticateResult
- uint32_t * pIndicationToken

8.1222.1 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.

8.1222.2 Field Documentation

8.1222.2.1 uim_authenticateResult* unpack_uim_SLQSUIMAuthenticate_t::pAuthenticateResult

8.1222.2.2 uim_cardResult* unpack_uim_SLQSUIMAuthenticate_t::pCardResult

8.1222.2.3 uint32_t* unpack_uim_SLQSUIMAuthenticate_t::pIndicationToken

8.1223 unpack_uim_SLQSUIMDepersonalization_t Struct Reference

Data Fields

- uim_remainingRetries * pRemainingRetries

8.1223.1 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.

8.1223.2 Field Documentation

8.1223.2.1 uim_remainingRetries* unpack_uim_SLQSUIMDepersonalization_t::pRemainingRetries

8.1224 unpack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t eventMask

8.1224.1 Detailed Description

Parameters

eventMask	- bit 0 - card status • bit 1 - SAP connection • bit 4 - physical slot status

8.1224.2 Field Documentation

8.1224.2.1 uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask

8.1225 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference

Data Fields

- uint8_t * pAutoSelection
- uim_personalizationStatus * pPersonalizationStatus
- uint8_t * pHaltSubscription

8.1225.1 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

8.1225.2 Field Documentation

8.1225.2.1 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pAutoSelection

8.1225.2.2 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pHaltSubscription

8.1225.2.3 uim_personalizationStatus* unpack_uim_SLQSUIMGetConfiguration_t::pPersonalizationStatus

8.1226 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference

Data Fields

- uim_cardResult * pCardResult
- uim_fileAttributes * pFileAttributes
- uint32_t * pIndicationToken

8.1226.1 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.

8.1226.2 Field Documentation

8.1226.2.1 uim_cardResult* unpack_uim_SLQSUIMGetFileAttributes_t::pCardResult

8.1226.2.2 uim_fileAttributes* unpack_uim_SLQSUIMGetFileAttributes_t::pFileAttributes

8.1226.2.3 uint32_t* unpack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

8.1227 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference

Data Fields

- uint8_t * pNumberOfPhySlot
- slots_t * pUimSlotsStatus

8.1227.1 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..

8.1227.2 Field Documentation

8.1227.2.1 uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot

8.1227.2.2 slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus

8.1228 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

Data Fields

- uim_refreshevent * pRefreshEvent

8.1228.1 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.

8.1228.2 Field Documentation

8.1228.2.1 uim_refreshevent* unpack_uim_SLQSUIMRefreshGetLastEvent_t::pRefreshEvent

8.1229 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference

Data Fields

- uim_cardStatus * pCardStatus

8.1229.1 Detailed Description

This structure contains information about Status change callback.

Parameters

pCardStatus	Card Status • See uim_cardStatus for more information.

8.1229.2 Field Documentation

8.1229.2.1 uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus

8.1230 unpack_uim_UnblockPin_t Struct Reference

Data Fields

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

8.1230.1 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

8.1230.2 Field Documentation

8.1230.2.1 uim_encryptedPIN1∗ unpack_uim_UnblockPin_t::pEncryptedPIN1

8.1230.2.2 uint32_t∗ unpack_uim_UnblockPin_t::pIndicationToken

8.1230.2.3 uim_remainingRetries∗ unpack_uim_UnblockPin_t::pRemainingRetries

8.1230.2.4 uint16_t unpack_uim_UnblockPin_t::Tlvresult

8.1231 unpack_uim_VerifyPin_t Struct Reference

Data Fields

- uim_remainingRetries ∗ pRemainingRetries
- uim_encryptedPIN1 ∗ pEncryptedPIN1
- uint32_t ∗ pIndicationToken
- uint16_t Tlvresult

8.1231.1 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.

8.1231.2 Field Documentation

8.1231.2.1 uim_encryptedPIN1∗ unpack_uim_VerifyPin_t::pEncryptedPIN1

8.1231.2.2 uint32_t∗ unpack_uim_VerifyPin_t::pIndicationToken

8.1231.2.3 uim_remainingRetries∗ unpack_uim_VerifyPin_t::pRemainingRetries

8.1231.2.4 uint16_t unpack_uim_VerifyPin_t::Tlvresult

8.1232 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

8.1232.1 Field Documentation

8.1232.1.1 voice_alphaIDInfo* unpack_voice_SLQSOriginateUSSD_t::pAlphaIDInfo

8.1232.1.2 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCallId

8.1232.1.3 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCcResultType

8.1232.1.4 voice_ccSUPSType* unpack_voice_SLQSOriginateUSSD_t::pCCSuppsType

8.1232.1.5 uint16_t* unpack_voice_SLQSOriginateUSSD_t::pfailureCause

8.1232.1.6 struct voice_USSInfo* unpack_voice_SLQSOriginateUSSD_t::pUSSDInfo

8.1233 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.1233.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

pCallId	• Unique call identifier for the call that must be answered.

8.1233.2 Field Documentation

8.1233.2.1 uint8_t* unpack_voice_SLQSVoiceAnswerCall_t::pCallId

8.1234 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1234.1 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

8.1234.2 Field Documentation

8.1234.2.1 uint8_t* unpack_voice_SLQSVoiceBurstDTMF_t::pCallID

8.1235 unpack_voice_SLQSVoiceDialCall_t Struct Reference

Data Fields

- uint8_t * pCallID
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- voice_ccSUPSType * pCCSUPSType

8.1235.1 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

8.1235.2 Field Documentation

8.1235.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceDialCall_t::pAlphaIDInfo

8.1235.2.2 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCallID

8.1235.2.3 uint8_t* unpack_voice_SLQSVoiceDialCall_t::pCCResultType

8.1235.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceDialCall_t::pCCSUPSType

8.1236 unpack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- uint8_t * pCallId

8.1236.1 Detailed Description

Parameters

pCallId	• Unique call identifier for the call that must be ended

8.1236.2 Field Documentation

8.1236.2.1 uint8_t* unpack_voice_SLQSVoiceEndCall_t::pCallId

8.1237 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

8.1237.1 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.

8.1237.2 Field Documentation

8.1237.2.1 voice_arrAlertingPattern∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingPattern

8.1237.2.2 voice_arrAlertingType∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingType

8.1237.2.3 voice_arrAlphaID∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlphaID

8.1237.2.4 voice_arrCalledPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCalledPartyNum

8.1237.2.5 voice_arrCallEndReason∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallEndReason

8.1237.2.6 voice_arrCallInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallInfo

8.1237.2.7 voice_arrConnectPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrConnectPartyNum

8.1237.2.8 voice_arrDiagInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrDiagInfo

8.1237.2.9 voice_arrRedirPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRedirPartyNum

8.1237.2.10 voice_arrRemotePartyName∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyName

8.1237.2.11 voice_arrRemotePartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyNum

8.1237.2.12 voice_arrSvcOption* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrSvcOption

8.1237.2.13 voice_arrUUSInfo* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrUUSInfo

8.1237.2.14 uint8_t* unpack_voice_SLQSVoiceGetAllCallInfo_t::pOTASPStatus

8.1237.2.15 uint8_t* unpack_voice_SLQSVoiceGetAllCallInfo_t::pVoicePrivacy

8.1238 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

8.1238.1 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.

8.1238.2 Field Documentation

8.1238.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallBarring_t::pAlphaIDInfo

8.1238.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCallID

8.1238.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCCResType

8.1238.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallBarring_t::pCCSUPSType

8.1238.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallBarring_t::pFailCause

8.1238.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

8.1239 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

8.1239.1 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.

8.1239.2 Field Documentation

8.1239.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pAlphaIDInfo

8.1239.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCallID

8.1239.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCResType

8.1239.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCSUPSType

8.1239.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pFailCause

8.1239.2.6 voice_getCallFWExtInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWExtInfo

8.1239.2.7 voice_getCallFWInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWInfo

8.1240 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

8.1240.1 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 voice_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

8.1240.2 Field Documentation

8.1240.2.1 uint32_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertingPattern

8.1240.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertType

8.1240.2.3 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pAlphaIDInfo

8.1240.2.4 voice_callInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pCallInfo

8.1240.2.5 voice_connectNumInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pConnectNumInfo

8.1240.2.6 voice_diagInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pDiagInfo

8.1240.2.7 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pOTASPStatus

8.1240.2.8 voice_remotePartyName* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyName

8.1240.2.9 voice_remotePartyNum* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyNum

8.1240.2.10 uint16_t* unpack_voice_SLQSVoiceGetCallInfo_t::pSrvOpt

8.1240.2.11 voice_UUSInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pUUSInfo

8.1240.2.12 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pVoicePrivacy

8.1241 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

8.1241.1 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.

8.1241.2 Field Documentation

8.1241.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallWaiting_t::pAlphaIDInfo

8.1241.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCallID

8.1241.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCResType

8.1241.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCSUPSType

8.1241.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pFailCause

8.1241.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

8.1242 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

8.1242.1 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.

8.1242.2 Field Documentation

8.1242.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIP_t::pAlphaIDInfo

8.1242.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCallID

8.1242.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCCResType

8.1242.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIP_t::pCCSUPSType

8.1242.2.5 voice_CLIPResp* unpack_voice_SLQSVoiceGetCLIP_t::pCLIPResp

8.1242.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIP_t::pFailCause

8.1243 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

8.1243.1 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.

8.1243.2 Field Documentation

8.1243.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIR_t::pAlphaIDInfo

8.1243.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCallID

8.1243.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCCResType

8.1243.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIR_t::pCCSUPSType

8.1243.2.5 voice_CLIRResp* unpack_voice_SLQSVoiceGetCLIR_t::pCLIRResp

8.1243.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIR_t::pFailCause

8.1244 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

8.1244.1 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.

8.1244.2 Field Documentation

8.1244.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCNAP_t::pAlphaIDInfo

8.1244.2.2 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCallID

8.1244.2.3 uint8_t* unpack_voice_SLQSVoiceGetCNAP_t::pCCResType

8.1244.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCNAP_t::pCCSUPSType

8.1244.2.5 voice_CNAPResp* unpack_voice_SLQSVoiceGetCNAP_t::pCNAPResp

8.1244.2.6 uint16_t* unpack_voice_SLQSVoiceGetCNAP_t::pFailCause

8.1245 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

8.1245.1 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.

8.1245.2 Field Documentation

8.1245.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLP_t::pAlphaIDInfo

8.1245.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCallID

8.1245.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCCResType

8.1245.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLP_t::pCCSUPSType

8.1245.2.5 voice_COLPResp* unpack_voice_SLQSVoiceGetCOLP_t::pCOLPResp

8.1245.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLP_t::pFailCause

8.1246 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

8.1246.1 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.

8.1246.2 Field Documentation

8.1246.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLR_t::pAlphaIDInfo

8.1246.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCallID

8.1246.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCCResType

8.1246.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLR_t::pCCSUPSType

8.1246.2.5 voice_COLRResp* unpack_voice_SLQSVoiceGetCOLR_t::pCOLRResp

8.1246.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLR_t::pFailCause

8.1247 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

8.1247.1 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 voice_airTimer for more information
pRoamTimer-Cnt(optional)	• Roam Timer Count • Value returned is read from NV_ROAM_CNT_I. • See voice_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.

8.1247.2 Field Documentation

8.1247.2.1 voice_airTimer* unpack_voice_SLQSVoiceGetConfig_t::pAirTimerCnt

8.1247.2.2 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pAutoAnswerStat

8.1247.2.3 voice_curAMRConfig* unpack_voice_SLQSVoiceGetConfig_t::pCurAMRConfig

8.1247.2.4 voice_prefVoiceSO* unpack_voice_SLQSVoiceGetConfig_t::pCurPrefVoiceSO

8.1247.2.5 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurrTTYMode

8.1247.2.6 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoiceDomainPref

8.1247.2.7 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoicePrivacyPref

8.1247.2.8 voice_roamTimer* unpack_voice_SLQSVoiceGetConfig_t::pRoamTimerCnt

8.1248 unpack_voice_SLQSVoiceManageCalls_t Struct Reference

Data Fields

- uint16_t * pFailCause

8.1248.1 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

8.1248.2 Field Documentation

8.1248.2.1 uint16_t* unpack_voice_SLQSVoiceManageCalls_t::pFailCause

8.1249 unpack_voice_SLQSVoiceSendFlash_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1249.1 Detailed Description

This structure contains the flash information associated with a call.

Parameters

pCallID	• Unique call identifier associated with the current call.

8.1249.2 Field Documentation

8.1249.2.1 uint8_t* unpack_voice_SLQSVoiceSendFlash_t::pCallID

8.1250 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

8.1250.1 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.

8.1250.2 Field Documentation

8.1250.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pAlphaIDInfo

8.1250.2.2 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCallID

8.1250.2.3 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCResType

8.1250.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCSUPSType

8.1250.2.5 uint16_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pFailCause

8.1251 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

8.1251.1 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.

8.1251.2 Field Documentation

8.1251.2.1 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAirTimerStatus

8.1251.2.2 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAutoAnsStatus

8.1251.2.3 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSOStatus

8.1251.2.4 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pRoamTimerStatus

8.1251.2.5 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pTTYConfigStatus

8.1251.2.6 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pVoiceDomainPrefStatus

8.1252 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference

Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

8.1252.1 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

8.1252.2 Field Documentation

8.1252.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetSUPSService_t::pAlphaIDInfo

8.1252.2.2 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCallID

8.1252.2.3 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCCResultType

8.1252.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetSUPSService_t::pCCSUPSType

8.1252.2.5 uint16_t* unpack_voice_SLQSVoiceSetSUPSService_t::pFailCause

8.1253 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference

Data Fields

- uint8_t * pCallID

8.1253.1 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

8.1253.2 Field Documentation

8.1253.2.1 uint8_t* unpack_voice_SLQSVoiceStartContDTMF_t::pCallID

8.1254 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference

Data Fields

- uint8_t callID

8.1254.1 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

8.1254.2 Field Documentation

8.1254.2.1 uint8_t unpack_voice_SLQSVoiceStopContDTMF_t::callID

8.1255 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference

Data Fields

- wds_DHCPProfileIdTlv ProfileIdTlv
- wds_DHCPLeaseStateTlv DHCPv4LeaseStateTlv
- wds_IPv4AdTlv IPv4AddrTlv
- wds_DHCPLeaseOptTlv DHCPv4LeaseOptTlv

8.1255.1 Detailed Description

This structure contains DHCP IPv4 client lease information

Parameters

ProfileIdTlv	profile identifier information, see wds_DHCPProfileIdTlv for more details
DHCPv4Lease-StateTlv	lease state, see wds_DHCPLeaseStateTlv for more details
IPv4AddrTlv	IPv4 address, see wds_IPv4AdTlv for more details
DHCPv4Lease-OptTlv	DHCP lease option, see wds_DHCPLeaseOptTlv for more details

8.1255.2 Field Documentation

8.1255.2.1 wds_DHCPLeaseOptTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseOptTlv

8.1255.2.2 wds_DHCPLeaseStateTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseStateTlv

8.1255.2.3 wds_IPv4AdTlv unpack_wds_DHCPv4ClientLease_ind_t::IPv4AddrTlv

8.1255.2.4 wds_DHCPProfileIdTlv unpack_wds_DHCPv4ClientLease_ind_t::ProfileIdTlv

8.1256 unpack_wds_GetAutoconnect_t Struct Reference

Data Fields

- uint32_t * psetting

8.1256.1 Detailed Description

auto connect data session setting parameter.

Parameters

pSetting	• NDIS auto connect setting – 0 - Disabled – 1 - Enabled

8.1256.2 Field Documentation

8.1256.2.1 uint32_t* unpack_wds_GetAutoconnect_t::psetting

8.1257 unpack_wds_GetByteTotals_t Struct Reference

Data Fields

- uint64_t * pTXTotalBytes
- uint64_t * pRXTotalBytes

8.1257.1 Detailed Description

Parameters

pTXTotalBytes	• Bytes transmitted without error
pRXTotalBytes	• Bytes received without error

8.1257.2 Field Documentation

8.1257.2.1 uint64_t* unpack_wds_GetByteTotals_t::pRXTotalBytes

8.1257.2.2 uint64_t* unpack_wds_GetByteTotals_t::pTXTotalBytes

8.1258 unpack_wds_GetConnectionRate_t Struct Reference

Data Fields

- uint32_t currentChannelTXRate
- uint32_t currentChannelRXRate
- uint32_t maxChannelTXRate
- uint32_t maxChannelRXRate

8.1258.1 Detailed Description

Parameters

pCurrentChannelTXRate	• Current channel Tx rate (in bps)
pCurrentChannelRXRate	• Current channel Rx rate (in bps)
pMaxChannelTXRate	• Maximum Tx rate (bps) that may be assigned to device by serving system.
pMaxChannelRXRate	• Maximum Rx rate (bps) that may be assigned to device by serving system.

8.1258.2 Field Documentation

8.1258.2.1 uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate

8.1258.2.2 uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate

8.1258.2.3 uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate

8.1258.2.4 uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate

8.1259 unpack_wds_GetDataBearerTechnology_t Struct Reference

Data Fields

- uint32_t * pDataBearer

8.1259.1 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

8.1259.2 Field Documentation

8.1259.2.1 uint32_t* unpack_wds_GetDataBearerTechnology_t::pDataBearer

8.1260 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]

8.1260.1 Detailed Description

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

Parameters

pdptype	• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile – 0 - PDP-IP (IPv4) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
ipaddr	• Preferred IPv4 address to be assigned to device
pridns	• Primary DNS IPv4 address preference
secdns	• Secondary DNS IPv4 address preference
ipaddrv6	• Preferred IPv6 address to be assigned to device
pridnsv6	• Primary DNS Pv6 address preference
secdnsv6	• Secondary DNS IPv6 address preference
auth	• 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.
name	• Profile name

apnsize	• Maximum number of characters (including NULL terminator) that APN name array can contain
apnname	• 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.
username	• Username used during network authentication

8.1260.2 Field Documentation

8.1260.2.1 int8_t unpack_wds_GetDefaultProfile_t::apnname[255]

8.1260.2.2 uint8_t unpack_wds_GetDefaultProfile_t::apnsize

8.1260.2.3 uint32_t unpack_wds_GetDefaultProfile_t::auth

8.1260.2.4 uint32_t unpack_wds_GetDefaultProfile_t::ipaddr

8.1260.2.5 uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6

8.1260.2.6 int8_t unpack_wds_GetDefaultProfile_t::name[255]

8.1260.2.7 uint8_t unpack_wds_GetDefaultProfile_t::namesize

8.1260.2.8 uint32_t unpack_wds_GetDefaultProfile_t::pdptype

8.1260.2.9 uint32_t unpack_wds_GetDefaultProfile_t::pridns

8.1260.2.10 uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6

8.1260.2.11 uint32_t unpack_wds_GetDefaultProfile_t::secdns

8.1260.2.12 uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6

8.1260.2.13 int8_t unpack_wds_GetDefaultProfile_t::username[255]

8.1260.2.14 uint8_t unpack_wds_GetDefaultProfile_t::usersize

8.1261 unpack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t index

8.1261.1 Detailed Description

Parameters

index	profile index

8.1261.2 Field Documentation

8.1261.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index

8.1262 unpack_wds_GetDormancyState_t Struct Reference

Data Fields

- uint32_t dormancyState

8.1262.1 Detailed Description

Parameters

dormancyState	• Dormancy state of current packet data session – 1 - Traffic channel dormant – 2 - Traffic channel active

8.1262.2 Field Documentation

8.1262.2.1 uint32_t unpack_wds_GetDormancyState_t::dormancyState

8.1263 unpack_wds_GetLastMobileIPError_t Struct Reference

Data Fields

- uint32_t error

8.1263.1 Detailed Description

structure to store last mobile IP error.

Parameters

error	• Status of last MIP call (or attempt) – Zero - Success – NonZero - Error code See qaGobiApiTableCallEndReasons.h for Mobile IP Error codes

8.1263.2 Field Documentation

8.1263.2.1 uint32_t unpack_wds_GetLastMobileIPError_t::error

8.1264 unpack_wds_GetMobileIP_t Struct Reference

Data Fields

- uint32_t mipMode

8.1264.1 Detailed Description

Parameters

mipMode	• Mobile IP setting – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 2 - Mobile IP only

8.1264.2 Field Documentation

8.1264.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

8.1265 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

8.1265.1 Detailed Description

Parameters

enabled	• Profile enabled: – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
address	• Home IPv4 address: – 0xFFFFFFFF - Unknown

primaryHA	• Primary home agent IPv4 address – 0xFFFFFFFF - Unknown
secondaryHA	• Secondary home agent IPv4 address – 0xFFFFFFFF - Unknown
revTunneling	• Reverse tunneling enabled – 0 - Disabled – 1 - Enabled – 0xFF - Unknown
naiSize	• The maximum number of characters (including NULL terminator) that the NAI array can contain.
NAI	• Network access identifier string
HASPI	• Home agent security parameter index
AAASPI	• AAA server security parameter index – 0xFFFFFFFF - Unknown
HAState	• Home agent key state – 0 - Unset – 1 - Set, default value – 2 - Set, modified from default – 3 - 0xFFFFFFFF - Unknown
AAAState	• AAA key state – 0 - Unset – 1 - Set, default value – 2 - Set, modified from default – 3 - 0xFFFFFFFF - Unknown

8.1265.2 Field Documentation

8.1265.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

8.1265.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAAState

8.1265.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address

8.1265.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled

8.1265.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI

8.1265.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HAState

8.1265.2.7 int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]

8.1265.2.8 uint8_t unpack_wds_GetMobileIPProfile_t::naiSize

8.1265.2.9 uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA

8.1265.2.10 uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling

8.1265.2.11 uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA

8.1266 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

8.1266.1 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	• Number of bytes transmitted without error.
pRXOkBytes-Count	• Number of bytes received without error.
pTXOKBytes-LastCall	• Number of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
pRXOKBytes-LastCall	• Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
pTXDropped-Count	• Number of outgoing packets dropped.
pRXDropped-Count	• Number of incoming packets dropped.

8.1266.2 Field Documentation

8.1266.2.1 uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount

8.1266.2.2 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount

8.1266.2.3 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOKBytesLastCall

8.1266.2.4 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors

8.1266.2.5 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows

8.1266.2.6 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses

8.1266.2.7 uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount

8.1266.2.8 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount

8.1266.2.9 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOKBytesLastCall

8.1266.2.10 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors

8.1266.2.11 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows

8.1266.2.12 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses

8.1267 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

8.1267.1 Detailed Description

Returns the packet data transfer statistics since the start of the current packet data.

Parameters

tXPacket-Successes	• No. of packets transmitted without error
rXPacket-Successes	• No. of packets received without error
tXPacketErrors	• No. of outgoing packets with framing errors
rXPacketErrors	• No. of incoming packets with framing errors
txPacket-Overflows	• Number of packets dropped because Tx buffer overflowed
rxPacket-Overflows	• Number of packets dropped because Rx buffer overflowed
txOkBytesCount	• No of bytes transmitted without error.
rxOkBytesCount	• No of bytes received without error.
txOKBytesLast-Call	• No of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
rxOKBytesLast-Call	• Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support

txDroppedCount	• Number of outgoing packets dropped.
rxDroppedCount	• Number of incoming packets dropped.

8.1267.2 Field Documentation

8.1267.2.1 uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount

8.1267.2.2 uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount

8.1267.2.3 uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall

8.1267.2.4 uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors

8.1267.2.5 uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows

8.1267.2.6 uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses

8.1267.2.7 uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount

8.1267.2.8 uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount

8.1267.2.9 uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall

8.1267.2.10 uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors

8.1267.2.11 uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows

8.1267.2.12 uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses

8.1268 unpack_wds_GetSessionDuration_t Struct Reference

Data Fields

- uint64_t callDuration

8.1268.1 Detailed Description

Parameters

callDuration	• Duration of the current packet session in milliseconds

8.1268.2 Field Documentation

8.1268.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

8.1269 unpack_wds_GetSessionState_t Struct Reference

Data Fields

- uint32_t connectionStatus

8.1269.1 Detailed Description

Parameters

connection-Status	state of the current packet data session

8.1269.2 Field Documentation

8.1269.2.1 uint32_t unpack_wds_GetSessionState_t::connectionStatus

8.1270 unpack_wds_RMSetTransferStatistics_t Struct Reference

8.1270.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.1271 unpack_wds_SetMobileIPProfile_t Struct Reference

8.1271.1 Detailed Description

Parameters

NULL	• this is a dummy structure

8.1272 unpack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

- PackCreateProfileOut * pCreateProfileOut
- uint8_t * pProfileID
- uint16_t Tlvresult

8.1272.1 Detailed Description

Parameters

pCreateProfile-Out	• SLQS Create profile Information

pProfileID	• SLQS profile identifier information
TlvresuIt	• TLV present flag, TURE if presented, otherwise FALSE

8.1272.2 Field Documentation

8.1272.2.1 PackCreateProfileOut* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut

8.1272.2.2 uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID

8.1272.2.3 uint16_t unpack_wds_SLQSCreateProfile_t::TlvresuIt

8.1273 unpack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint16_t extendedErrorCode

8.1273.1 Detailed Description

Parameters

pExtendedError-Code[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_INTERN-AL is returned by device. • See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description.

8.1273.2 Field Documentation

8.1273.2.1 uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode

8.1274 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

8.1274.1 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

8.1274.2 Field Documentation

8.1274.2.1 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease

8.1274.2.2 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled

8.1274.2.3 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile

8.1274.2.4 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]

8.1274.2.5 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen

8.1274.2.6 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]

8.1275 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

Data Fields

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

8.1275.1 Detailed Description

Data System Status

Parameters

prefNetwork	• Preferred Network • Values: – 0 - 3GPP – 1 - 3GPP2
networkInfoLen	• 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
currNetworkInfo	• Network information • See currNetworkInfo for more details

8.1275.2 Field Documentation

8.1275.2.1 currNetworkInfo unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]

8.1275.2.2 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen

8.1275.2.3 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork

8.1276 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

8.1276.1 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.

8.1276.2 Field Documentation

8.1276.2.1 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_rx_rate

8.1276.2.2 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_tx_rate

8.1276.2.3 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_rx_rate

8.1276.2.4 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_tx_rate

8.1277 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference

Data Fields

- uint8_t dataBearerMask
- qmiWDSDataBearerTechnology curDataBearerTechnology
- qmiWDSDataBearerTechnology lastCallDataBearerTechnology

8.1277.1 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.
curDataBearerTechnology[OUT]	• current data bearer technology value. – NULL if the parameter is not required
lastCallDataBearerTechnology[OUT]	• last call data bearer technology value. – NULL if the parameter is not required

8.1277.2 Field Documentation

8.1277.2.1 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::curDataBearerTechnology

8.1277.2.2 uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask

8.1277.2.3 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology

8.1278 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

8.1278.1 Detailed Description

This structure contains the DUN Call Info response parameters

Parameters

connectionStatus	• See connectionStatus for more information
callEndReason	• Last modem call end reason • See qaGobiApiTableCallEndReasons.h for Call End Reason • Only valid if the last call made was DUN, else zero is returned

txOKBytesCount	• Number of bytes transmitted without error • Returned only if a data call is up
rxOKBytesCount	• Number of bytes received without error • Returned only if a data call is up
dormancyStatus	• Current traffic channel status • Returned if a data call is up – 0x01 - Traffic channel dormant – 0x02 - Traffic channel active
dataBearerTech	• 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
channelRate	• See dunchannelRate for more information
lastCallTXOK-BytesCnt	• Number of bytes transmitted without error during the last data call (0 if no call was made). • Return only if not in a call and the previous call was made using DUN.
lastCallRXOK-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.

mdmCall-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)
lastCallData-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

8.1278.2 Field Documentation

8.1278.2.1 uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason

8.1278.2.2 dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate

8.1278.2.3 connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus

8.1278.2.4 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech

8.1278.2.5 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus

8.1278.2.6 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech

8.1278.2.7 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt

8.1278.2.8 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt

8.1278.2.9 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive

8.1278.2.10 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount

8.1278.2.11 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount

8.1279 unpack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- UnPackGetProfileSettingOut * pProfileSettings
- uint8_t ProfileType
- uint16_t Tlvresult

8.1279.1 Detailed Description

This structure contains the profile setting information of parameter pOutput for API unpack_wds_SLQSGetProfile-Settings

Parameters

SlqsProfile3GPP	3GPP profile
SlqsProfile3GP-P2	3GPP2 profile

8.1279.2 Field Documentation

8.1279.2.1 UnPackGetProfileSettingOut* unpack_wds_SLQSGetProfileSettings_t::pProfileSettings

8.1279.2.2 uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType

8.1279.2.3 uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult

8.1280 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

8.1280.1 Detailed Description

Parameters

IPv4	ipv4 address
ProfileName	profile name
PDPType	PDP type
APNName	APN name
PrimaryDNSV4	primary dns IPV4
SecondaryDNS-V4	secondary dns IPV4
UMTSGranted-QoS	UMTS Granted Qos
GPRSGranted-QoS	GPRS Granted QoS
Username	username for authentication process
Authentication	authentication for authentication process
ProfielID	profile ID
GWAddressV4	Gateway IPv4
SubnetMaskV4	Subnet mask IPV4
PCSCFAddrPC-O	PCSCF address PCO
ServerAddrList	PCSCF server address list IPV4
PCSCFFQDN-AddrList	PCSCF FQDN address list IPV4
PrimaryDNSV6	Primary DNS IPV6
SecondryDNS-V6	Secondry DNS IPV6
Mtu	Maximum Transfer Unit
DomainList	domain list
IPFamily-Preference	ip family preference, it could be IPV4 or IPV6
IMCNflag	IM control flag, value: TRUE or FALSE
Technology	technology on which current packet data session is in progress • Values: – 32767 - CDMA – 32764 - UMTS – 30592 - EPC – 30584 - modem link local
IPV6AddrInfo	address information IPV6
IPV6GWAddr-Info	gateway address information IPV6

8.1280.2 Field Documentation

8.1280.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

8.1280.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

8.1280.2.3 struct **wds_DomainNameList** unpack_wds_SLQSGetRuntimeSettings_t::DomainList

8.1280.2.4 struct **wds_GPRSQoS** unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

8.1280.2.5 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4

8.1280.2.6 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag

8.1280.2.7 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference

8.1280.2.8 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4

8.1280.2.9 struct **wds_IPV6AddressInfo** unpack_wds_SLQSGetRuntimeSettings_t::IPV6AddrInfo

8.1280.2.10 struct **wds_IPV6GWAddressInfo** unpack_wds_SLQSGetRuntimeSettings_t::IPV6GWAddrInfo

8.1280.2.11 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu

8.1280.2.12 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO

8.1280.2.13 struct **wds_PCSCFFQDNAddressList** unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList

8.1280.2.14 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPType

8.1280.2.15 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4

8.1280.2.16 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]

8.1280.2.17 struct **wds_ProfileIdentifier** unpack_wds_SLQSGetRuntimeSettings_t::ProfileID

8.1280.2.18 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]

8.1280.2.19 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4

8.1280.2.20 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]

8.1280.2.21 struct **wds_PCSCFIPv4ServerAddressList** unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList

8.1280.2.22 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4

8.1280.2.23 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology

8.1280.2.24 **LibPackUMTSQoS** unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS

8.1280.2.25 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]

8.1281 unpack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint16_t * pExtErrorCode

8.1281.1 Detailed Description

This structure contains out parameters for unpack_wds_SLQSModifyProfile

Parameters

pExtErrorCode	• The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_-REG_xxx. • 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.

8.1281.2 Field Documentation

8.1281.2.1 uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode

8.1282 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint16_t Tlvresult

8.1282.1 Detailed Description

Parameters

Tlvresult	unpack result

8.1282.2 Field Documentation

8.1282.2.1 uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

8.1283 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

8.1283.1 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.

8.1283.2 Field Documentation

8.1283.2.1 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID

8.1283.2.2 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status

8.1283.2.3 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily

8.1283.2.4 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd

8.1283.2.5 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason

8.1283.2.6 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName

8.1283.2.7 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason

8.1283.2.8 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType

8.1284 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

8.1284.1 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	Data bearer technology Extended SO Mask

8.1284.2 Field Documentation

8.1284.2.1 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail

8.1284.2.2 wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]

8.1284.2.3 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail

8.1284.2.4 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail

8.1284.2.5 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechExtAvail

8.1284.2.6 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtRatValue

8.1284.2.7 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtSoMask

8.1284.2.8 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology

8.1284.2.9 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechnologyExt

8.1284.2.10 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail

8.1284.2.11 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus

8.1284.2.12 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail

8.1284.2.13 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus

8.1284.2.14 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen

8.1284.2.15 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork

8.1284.2.16 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask

8.1284.2.17 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes

8.1284.2.18 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts

8.1284.2.19 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask

8.1284.2.20 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes

8.1284.2.21 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts

8.1284.2.22 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail

8.1285 unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

Data Fields

- wdsDhcpv4HwConfig * pHwConfig
- wdsDhcpv4OptionList * pRequestOptionList

8.1285.1 Detailed Description

WDS SWI DHCPv4 Config Structure

Parameters

pHWConfig	• pointer to HW Config structure
pRequestOption-List	• pointer to Option List structure to be sent in DHCP request

8.1285.2 Field Documentation

8.1285.2.1 wdsDhcpv4HwConfig* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig

8.1285.2.2 wdsDhcpv4OptionList* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList

8.1286 unpack_wds_SLQSSGetLoopback_t Struct Reference

Data Fields

- uint8_t ByteLoopbackMode
- uint8_t ByteLoopbackMultiplier

8.1286.1 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.

8.1286.2 Field Documentation

8.1286.2.1 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode

8.1286.2.2 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier

8.1287 unpack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

- uint32_t * psid
- uint32_t * pFailureReason
- uint32_t * pVerboseFailReasonType
- uint32_t * pVerboseFailureReason

8.1287.1 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

8.1287.2 Field Documentation

8.1287.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

8.1287.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

8.1287.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

8.1287.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

8.1288 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]

8.1288.1 Detailed Description

This structure contains the unpack parameters retrieved by the API unpack_wds_SLQSWdsSwiPDPRuntime-Settings

Parameters

contextId	(optional) • Context Identifier – 0xFF - Not Available
bearerId	(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
apnName	(optional) • APN name associated with the context id – NULL terminated by default.
ipv4Address	(optional) • IPv4 Address – 0xFFFF - Not Available

ipv4GWAddress	(optional) • IPv4 Gateway Address – 0xFFFF - Not Available
prDNSIPv4-Address	(optional) • Primary DNS IPv4 Address – 0xFFFF - Not Available
seDNSIPv4-Address	(optional) • Secondary DNS IPv4 Address – 0xFFFF - Not Available
ipv6Address	(optional) • IPv6 Address • See ipv6AddressInfo for more information
ipv6GWAddress	(optional) • IPv6 Gateway Address • See ipv6AddressInfo for more information
prDNSIPv6-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
seDNSIPv6-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
prPCSCFIPv4-Address	(optional) • Primary PCSCF IPv4 Address
sePCSCFIPv4-Address	(optional) • Secondary PCSCF IPv4 Address
prPCSCFIPv6-Address	(optional) • Primary PCSCF IPv6 Address • This is an 8-element array of 16-bit numbers, each of which is in big-endian format
sePCSCFIPv6-Address	(optional) • Secondary PCSCF IPv6 Address • This is an 8-element array of 16-bit numbers, each of which is in big-endian format

8.1288.2 Field Documentation

8.1288.2.1 int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]

8.1288.2.2 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId

8.1288.2.3 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.1288.2.4 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address

8.1288.2.5 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress

8.1288.2.6 struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address

8.1288.2.7 struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress

8.1288.2.8 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address

8.1288.2.9 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]

8.1288.2.10 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address

8.1288.2.11 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]

8.1288.2.12 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address

8.1288.2.13 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]

8.1288.2.14 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address

8.1288.2.15 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]

8.1289 UnPackGetProfileSettingOut Struct Reference

Data Fields

- UnpackQmiProfileInfo curProfile
- uint16_t * pExtErrCode

8.1289.1 Detailed Description

This structure contains the profile settings output

Parameters

curProfile	• Structure containing details of the current profile
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.

8.1289.2 Field Documentation

8.1289.2.1 UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile

8.1289.2.2 uint16_t* UnPackGetProfileSettingOut::pExtErrCode

8.1290 unpackWdsProfileParam Union Reference

Data Fields

- LibpackProfile3GPP SlqsProfile3GPP
- LibpackProfile3GPP2 SlqsProfile3GPP2

8.1290.1 Detailed Description

This union WdsProfileParam consists of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

Parameters

SlqsProfile3GPP	3GPP profile
SlqsProfile3GP-P2	3GPP2 profile

8.1290.2 Field Documentation

8.1290.2.1 LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP

8.1290.2.2 LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2

8.1291 urlAddr Struct Reference

Data Fields

- CHAR address [256]
- BYTE TlvPresent

8.1291.1 Detailed Description

This structure contains LOC Get Server URL address field.

Parameters

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

8.1291.2 Field Documentation

8.1291.2.1 CHAR urlAddr::address[256]

8.1291.2.2 BYTE urlAddr::TlvPresent

8.1292 USBCompConfig Struct Reference

Data Fields

- BYTE * pUSBComp

8.1292.1 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

8.1292.2 Field Documentation

8.1292.2.1 BYTE* USBCompConfig::pUSBComp

8.1293 USBCompParams Struct Reference

Data Fields

- BYTE * pUSBComp
- BYTE * pNumSupUSBComps
- BYTE * pSupUSBComps

8.1293.1 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

8.1293.2 Field Documentation

8.1293.2.1 BYTE* USBCompParams::pNumSupUSBComps

8.1293.2.2 BYTE* USBCompParams::pSupUSBComps

8.1293.2.3 BYTE* USBCompParams::pUSBComp

8.1294 USSDNoWaitIndicationInfo Struct Reference

Data Fields

- BYTE * pError
- BYTE * pFailureCause
- struct USSInfo * pUSSDData
- alphaIDInfo * pAlphaIdentifier

8.1294.1 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.

8.1294.2 Field Documentation

8.1294.2.1 alphaIDInfo* USSDNoWaitIndicationInfo::pAlphaIdentifier

8.1294.2.2 BYTE* USSDNoWaitIndicationInfo::pError

8.1294.2.3 BYTE* USSDNoWaitIndicationInfo::pFailureCause

8.1294.2.4 struct USSInfo* USSDNoWaitIndicationInfo::pUSSDData

8.1295 USSDRespFNetwork Struct Reference

Data Fields

- char * pTypeCode
- char * pRespData

8.1295.1 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

8.1295.2 Field Documentation

8.1295.2.1 char* USSDRespFNetwork::pRespData

8.1295.2.2 char* USSDRespFNetwork::pTypeCode

8.1296 USSInfo Struct Reference

Data Fields

- BYTE ussDCS
- BYTE ussLen
- BYTE ussData [182]

8.1296.1 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

8.1296.2 Field Documentation

8.1296.2.1 BYTE USSInfo::ussData[182]

8.1296.2.2 BYTE USSInfo::ussDCS

8.1296.2.3 BYTE USSInfo::ussLen

8.1297 USSResp Struct Reference

Data Fields

- WORD * pfailureCause
- alphaIDInfo * pAlphaIDInfo
- struct USSInfo * pUSSDInfo
- BYTE * pCcResultType
- BYTE * pCallId
- ccSUPSType * pCCSuppsType

8.1297.1 Field Documentation

8.1297.1.1 alphaIDInfo* USSResp::pAlphaIDInfo

8.1297.1.2 BYTE* USSResp::pCallId

8.1297.1.3 BYTE* USSResp::pCcResultType

8.1297.1.4 ccSUPSType* USSResp::pCCSuppsType

8.1297.1.5 WORD* USSResp::pfailureCause

8.1297.1.6 struct USSInfo* USSResp::pUSSDInfo

8.1298 UUSInfo Struct Reference

Data Fields

- BYTE UUSType
- BYTE UUSDcs
- BYTE UUSDatalen
- BYTE UUSData [255]

8.1298.1 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

8.1298.2 Field Documentation

8.1298.2.1 BYTE UUSInfo::UUSData[255]

8.1298.2.2 BYTE UUSInfo::UUSDatalen

8.1298.2.3 BYTE UUSInfo::UUSDcs

8.1298.2.4 BYTE UUSInfo::UUSType

8.1299 verifyUIMPIN Struct Reference

Data Fields

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

8.1299.1 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.

8.1299.2 Field Documentation

8.1299.2.1 BYTE verifyUIMPIN::pinID

8.1299.2.2 BYTE verifyUIMPIN::pinLen

8.1299.2.3 BYTE verifyUIMPIN::pinVal[255]

8.1300 voice_airTimer Struct Reference

Data Fields

- uint8_t namID
- uint32_t airTimerValue

8.1300.1 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.

8.1300.2 Field Documentation

8.1300.2.1 uint32_t voice_airTimer::airTimerValue

8.1300.2.2 uint8_t voice_airTimer::namID

8.1301 voice_allCallsAlphaIDInfo Struct Reference

Data Fields

- uint8_t callID
- voice_alphaIDInfo AlphaIDInfo

8.1301.1 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.

8.1301.2 Field Documentation

8.1301.2.1 voice_alphaIDInfo voice_allCallsAlphaIDInfo::AlphaIDInfo

8.1301.2.2 uint8_t voice_allCallsAlphaIDInfo::callID

8.1302 voice_allCallsDiagInfo Struct Reference

Data Fields

- uint8_t callID
- voice_diagInfo DiagInfo

8.1302.1 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.

8.1302.2 Field Documentation

8.1302.2.1 uint8_t voice_allCallsDiagInfo::callID

8.1302.2.2 voice_diagInfo voice_allCallsDiagInfo::DiagInfo

8.1303 voice_allCallsUUSInfo Struct Reference

Data Fields

- uint8_t callID
- voice_UUSInfo uusInfo

8.1303.1 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.

8.1303.2 Field Documentation

8.1303.2.1 uint8_t voice_allCallsUUSInfo::callID

8.1303.2.2 voice_UUSInfo voice_allCallsUUSInfo::uusInfo

8.1304 voice_alphaIDInfo Struct Reference

Data Fields

- uint8_t alphaDcs
- uint8_t alphaLen
- uint8_t alphaText [255]

8.1304.1 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

8.1304.2 Field Documentation

8.1304.2.1 uint8_t voice_alphaIDInfo::alphaDcs

8.1304.2.2 uint8_t voice_alphaIDInfo::alphaLen

8.1304.2.3 uint8_t voice_alphaIDInfo::alphaText[255]

8.1305 voice_arrAlertingPattern Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint32_t alertingPattern [20]

8.1305.1 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

8.1305.2 Field Documentation

8.1305.2.1 uint32_t voice_arrAlertingPattern::alertingPattern[20]

8.1305.2.2 uint8_t voice_arrAlertingPattern::callID[20]

8.1305.2.3 uint8_t voice_arrAlertingPattern::numInstances

8.1306 voice_arrAlertingType Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint8_t AlertingType [20]

8.1306.1 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

8.1306.2 Field Documentation

8.1306.2.1 uint8_t voice_arrAlertingType::AlertingType[20]

8.1306.2.2 uint8_t voice_arrAlertingType::callID[20]

8.1306.2.3 uint8_t voice_arrAlertingType::numInstances

8.1307 voice_arrAlphaID Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsAlphaIDInfo allCallsAlphaIDInfoArr [20]

8.1307.1 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.

8.1307.2 Field Documentation

8.1307.2.1 voice_allCallsAlphaIDInfo voice_arrAlphaID::allCallsAlphaIDInfoArr[20]

8.1307.2.2 uint8_t voice_arrAlphaID::numInstances

8.1308 voice_arrCalledPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo CalledPartyNum [20]

8.1308.1 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.

8.1308.2 Field Documentation

8.1308.2.1 voice_peerNumberInfo voice_arrCalledPartyNum::CalledPartyNum[20]

8.1308.2.2 uint8_t voice_arrCalledPartyNum::numInstances

8.1309 voice_arrCallEndReason Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t callEndReason [20]

8.1309.1 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

8.1309.2 Field Documentation

8.1309.2.1 uint16_t voice_arrCallEndReason::callEndReason[20]

8.1309.2.2 uint8_t voice_arrCallEndReason::callID[20]

8.1309.2.3 uint8_t voice_arrCallEndReason::numInstances

8.1310 voice_arrCallInfo Struct Reference

Data Fields

- uint8_t numInstances

- voice_getAllCallInformation getAllCallInfo [20]

8.1310.1 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_getAllCallInformation for more information.

8.1310.2 Field Documentation

8.1310.2.1 voice_getAllCallInformation voice_arrCallInfo::getAllCallInfo[20]

8.1310.2.2 uint8_t voice_arrCallInfo::numInstances

8.1311 voice_arrConnectPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo ConnectedPartyNum [20]

8.1311.1 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.

8.1311.2 Field Documentation

8.1311.2.1 voice_peerNumberInfo voice_arrConnectPartyNum::ConnectedPartyNum[20]

8.1311.2.2 uint8_t voice_arrConnectPartyNum::numInstances

8.1312 voice_arrDiagInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsDiagInfo DiagInfo [20]

8.1312.1 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.

8.1312.2 Field Documentation

8.1312.2.1 voice_allCallsDiagInfo voice_arrDiagInfo::DiagInfo[20]

8.1312.2.2 uint8_t voice_arrDiagInfo::numInstances

8.1313 voice_arrRedirPartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_peerNumberInfo RedirPartyNum [20]

8.1313.1 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.

8.1313.2 Field Documentation

8.1313.2.1 uint8_t voice_arrRedirPartyNum::numInstances

8.1313.2.2 voice_peerNumberInfo voice_arrRedirPartyNum::RedirPartyNum[20]

8.1314 voice_arrRemotePartyName Struct Reference

Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyName GetAllCallRmtPtyName [20]

8.1314.1 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.

8.1314.2 Field Documentation

8.1314.2.1 voice_getAllCallRmtPtyName voice_arrRemotePartyName::GetAllCallRmtPtyName[20]

8.1314.2.2 uint8_t voice_arrRemotePartyName::numInstances

8.1315 voice_arrRemotePartyNum Struct Reference

Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyNum RmtPtyNum [20]

8.1315.1 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.

8.1315.2 Field Documentation

8.1315.2.1 uint8_t voice_arrRemotePartyNum::numInstances

8.1315.2.2 voice_getAllCallRmtPtyNum voice_arrRemotePartyNum::RmtPtyNum[20]

8.1316 voice_arrSvcOption Struct Reference

Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t srvOption [20]

8.1316.1 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.

8.1316.2 Field Documentation

8.1316.2.1 uint8_t voice_arrSvcOption::callID[20]

8.1316.2.2 uint8_t voice_arrSvcOption::numInstances

8.1316.2.3 uint16_t voice_arrSvcOption::srvOption[20]

8.1317 voice_arrUUSInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_allCallsUUSInfo AllCallsUUSInfo [20]

8.1317.1 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.

8.1317.2 Field Documentation

8.1317.2.1 voice_allCallsUUSInfo voice_arrUUSInfo::AllCallsUUSInfo[20]

8.1317.2.2 uint8_t voice_arrUUSInfo::numInstances

8.1318 voice_burstDTMFInfo Struct Reference

Data Fields

- uint8_t * pCallID
- uint8_t digitCnt
- uint8_t pDigitBuff [255]

8.1318.1 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

8.1318.2 Field Documentation

8.1318.2.1 uint8_t voice_burstDTMFInfo::digitCnt

8.1318.2.2 uint8_t* voice_burstDTMFInfo::pCallID

8.1318.2.3 uint8_t voice_burstDTMFInfo::pDigitBuff[255]

8.1319 voice_calledPartySubAdd Struct Reference

Data Fields

- uint8_t extBit
- uint8_t subAddrType
- uint8_t oddEvenInd
- uint8_t subAddrLen
- uint8_t subAddr [255]

8.1319.1 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.

8.1319.2 Field Documentation

8.1319.2.1 uint8_t voice_calledPartySubAdd::extBit

8.1319.2.2 uint8_t voice_calledPartySubAdd::oddEvenInd

8.1319.2.3 uint8_t voice_calledPartySubAdd::subAddr[255]

8.1319.2.4 uint8_t voice_calledPartySubAdd::subAddrLen

8.1319.2.5 uint8_t voice_calledPartySubAdd::subAddrType

8.1320 voice_callFwdTypeAndPlan Struct Reference

Data Fields

- uint8_t numberType
- uint8_t numberPlan

8.1320.1 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

8.1320.2 Field Documentation

8.1320.2.1 uint8_t voice_callFwdTypeAndPlan::numberPlan

8.1320.2.2 uint8_t voice_callFwdTypeAndPlan::numberType

8.1321 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]

8.1321.1 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.

8.1321.2 Field Documentation

8.1321.2.1 uint8_t voice_callFWExtInfo::noReplyTimer

8.1321.2.2 uint8_t voice_callFWExtInfo::number[255]

8.1321.2.3 uint8_t voice_callFWExtInfo::numLen

8.1321.2.4 uint8_t voice_callFWExtInfo::numPlan

8.1321.2.5 uint8_t voice_callFWExtInfo::numType

8.1321.2.6 uint8_t voice_callFWExtInfo::PI

8.1321.2.7 uint8_t voice_callFWExtInfo::SI

8.1321.2.8 uint8_t voice_callFWExtInfo::SvcClass

8.1321.2.9 uint8_t voice_callFWExtInfo::SvcStatus

8.1322 voice_callFWInfo Struct Reference

Data Fields

- uint8_t SvcStatus
- uint8_t SvcClass
- uint8_t numLen
- uint8_t number [255]
- uint8_t noReplyTimer

8.1322.1 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.

8.1322.2 Field Documentation

8.1322.2.1 uint8_t voice_callFWInfo::noReplyTimer

8.1322.2.2 uint8_t voice_callFWInfo::number[255]

8.1322.2.3 uint8_t voice_callFWInfo::numLen

8.1322.2.4 uint8_t voice_callFWInfo::SvcClass

8.1322.2.5 uint8_t voice_callFWInfo::SvcStatus

8.1323 voice_callInfo Struct Reference

Data Fields

- uint8_t callID
- uint8_t callState
- uint8_t callType
- uint8_t direction
- uint8_t mode

8.1323.1 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

8.1323.2 Field Documentation

8.1323.2.1 uint8_t voice_callInfo::callID

8.1323.2.2 uint8_t voice_callInfo::callState

8.1323.2.3 uint8_t voice_callInfo::callType

8.1323.2.4 uint8_t voice_callInfo::direction

8.1323.2.5 uint8_t voice_callInfo::mode

8.1324 voice_ccSUPSType Struct Reference

Data Fields

- uint8_t svcType
- uint8_t reason

8.1324.1 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.

8.1324.2 Field Documentation

8.1324.2.1 uint8_t voice_ccSUPSType::reason

8.1324.2.2 uint8_t voice_ccSUPSType::svcType

8.1325 voice_CLIPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1325.1 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

8.1325.2 Field Documentation

8.1325.2.1 uint8_t voice_CLIPResp::ActiveStatus

8.1325.2.2 uint8_t voice_CLIPResp::ProvisionStatus

8.1326 voice_CLIRResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1326.1 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

8.1326.2 Field Documentation

8.1326.2.1 uint8_t voice_CLIRResp::ActiveStatus

8.1326.2.2 uint8_t voice_CLIRResp::ProvisionStatus

8.1327 voice_CNAPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1327.1 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

8.1327.2 Field Documentation

8.1327.2.1 uint8_t voice_CNAPResp::ActiveStatus

8.1327.2.2 uint8_t voice_CNAPResp::ProvisionStatus

8.1328 voice_COLPResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1328.1 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

8.1328.2 Field Documentation

8.1328.2.1 uint8_t voice_COLPResp::ActiveStatus

8.1328.2.2 uint8_t voice_COLPResp::ProvisionStatus

8.1329 voice_COLRResp Struct Reference

Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

8.1329.1 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

8.1329.2 Field Documentation

8.1329.2.1 uint8_t voice_COLRResp::ActiveStatus

8.1329.2.2 uint8_t voice_COLRResp::ProvisionStatus

8.1330 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]

8.1330.1 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.

8.1330.2 Field Documentation

8.1330.2.1 uint8_t voice_connectNumInfo::callerID[81]

8.1330.2.2 uint8_t voice_connectNumInfo::callerIDLen

8.1330.2.3 uint8_t voice_connectNumInfo::numPlan

8.1330.2.4 uint8_t voice_connectNumInfo::numPresInd

8.1330.2.5 uint8_t voice_connectNumInfo::numType

8.1330.2.6 uint8_t voice_connectNumInfo::screeningInd

8.1331 voice_CUGInfo Struct Reference

Data Fields

- uint16_t CUGIndex
- uint8_t SuppPrefCUG
- uint8_t SuppOA

8.1331.1 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

8.1331.2 Field Documentation

8.1331.2.1 uint16_t voice_CUGInfo::CUGIndex

8.1331.2.2 uint8_t voice_CUGInfo::SuppOA

8.1331.2.3 uint8_t voice_CUGInfo::SuppPrefCUG

8.1332 voice_curAMRConfig Struct Reference

Data Fields

- uint8_t gsmAmrStat
- uint8_t wcdmaAmrStat

8.1332.1 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

8.1332.2 Field Documentation

8.1332.2.1 uint8_t voice_curAMRConfig::gsmAmrStat

8.1332.2.2 uint8_t voice_curAMRConfig::wcdmaAmrStat

8.1333 voice_diagInfo Struct Reference

Data Fields

- uint8_t diagInfoLen
- uint8_t diagnosticInfo [255]

8.1333.1 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.

8.1333.2 Field Documentation

8.1333.2.1 uint8_t voice_diagInfo::diagInfoLen

8.1333.2.2 uint8_t voice_diagInfo::diagnosticInfo[255]

8.1334 voice_DTMFLengths Struct Reference

Data Fields

- uint8_t DTMFPulseWidth
- uint8_t DTMFInterdigitInterval

8.1334.1 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

8.1334.2 Field Documentation

8.1334.2.1 uint8_t voice_DTMFLengths::DTMFInterdigitInterval

8.1334.2.2 uint8_t voice_DTMFLengths::DTMFPulseWidth

8.1335 voice_getAllCallInformation Struct Reference

Data Fields

- voice_callInfo CallInfo
- uint8_t isEmpty
- uint8_t ALS

8.1335.1 Detailed Description

This structure contains information related to call state change.

Parameters

CallInfo	• See voice_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

8.1335.2 Field Documentation

8.1335.2.1 uint8_t voice_getAllCallInformation::ALS

8.1335.2.2 voice_callInfo voice_getAllCallInformation::CallInfo

8.1335.2.3 uint8_t voice_getAllCallInformation::isEmpty

8.1336 voice_getAllCallRmtPtyName Struct Reference

Data Fields

- uint8_t callID
- voice_remotePartyName RemotePartyName

8.1336.1 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.

8.1336.2 Field Documentation

8.1336.2.1 uint8_t voice_getAllCallRmtPtyName::callID

8.1336.2.2 voice_remotePartyName voice_getAllCallRmtPtyName::RemotePartyName

8.1337 voice_getAllCallRmtPtyNum Struct Reference

Data Fields

- uint8_t callID
- voice_remotePartyNum RemotePartyNum

8.1337.1 Detailed Description

This structure contains information for All Call Remote Party Numbers

Parameters

callID	• Unique call identifier for the call.
RemoteParty-Num	• See voice_remotePartyNum for more information.

8.1337.2 Field Documentation

8.1337.2.1 uint8_t voice_getAllCallRmtPtyNum::callID

8.1337.2.2 voice_remotePartyNum voice_getAllCallRmtPtyNum::RemotePartyNum

8.1338 voice_getCallFWExtInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_callFWExtInfo CallFWExtInfo [20]

8.1338.1 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.

8.1338.2 Field Documentation

8.1338.2.1 voice_callFWExtInfo voice_getCallFWExtInfo::CallFWExtInfo[20]

8.1338.2.2 uint8_t voice_getCallFWExtInfo::numInstances

8.1339 voice_getCallFWInfo Struct Reference

Data Fields

- uint8_t numInstances
- voice_callFWInfo CallFWInfo [20]

8.1339.1 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.

8.1339.2 Field Documentation

8.1339.2.1 voice_callFWInfo voice_getCallFWInfo::CallFWInfo[20]

8.1339.2.2 uint8_t voice_getCallFWInfo::numInstances

8.1340 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]

8.1340.1 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.

8.1340.2 Field Documentation

8.1340.2.1 uint8_t voice_peerNumberInfo::callID

8.1340.2.2 uint8_t voice_peerNumberInfo::number[81]

8.1340.2.3 uint8_t voice_peerNumberInfo::numLen

8.1340.2.4 uint8_t voice_peerNumberInfo::numPI

8.1340.2.5 uint8_t voice_peerNumberInfo::numPlan

8.1340.2.6 uint8_t voice_peerNumberInfo::numSI

8.1340.2.7 uint8_t voice_peerNumberInfo::numType

8.1341 voice_prefVoiceSO Struct Reference

Data Fields

- uint8_t namID
- uint8_t evrcCapability
- uint16_t homePageVoiceSO
- uint16_t homeOrigVoiceSO
- uint16_t roamOrigVoiceSO

8.1341.1 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

8.1341.2 Field Documentation

8.1341.2.1 uint8_t voice_prefVoiceSO::evrcCapability

8.1341.2.2 uint16_t voice_prefVoiceSO::homeOrigVoiceSO

8.1341.2.3 uint16_t voice_prefVoiceSO::homePageVoiceSO

8.1341.2.4 uint8_t voice_prefVoiceSO::namID

8.1341.2.5 uint16_t voice_prefVoiceSO::roamOrigVoiceSO

8.1342 voice_remotePartyName Struct Reference

Data Fields

- uint8_t namePI
- uint8_t codingScheme
- uint8_t nameLen
- uint8_t callerName [255]

8.1342.1 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.

8.1342.2 Field Documentation

8.1342.2.1 uint8_t voice_remotePartyName::callerName[255]

8.1342.2.2 uint8_t voice_remotePartyName::codingScheme

8.1342.2.3 uint8_t voice_remotePartyName::nameLen

8.1342.2.4 uint8_t voice_remotePartyName::namePI

8.1343 voice_remotePartyNum Struct Reference

Data Fields

- uint8_t presentationInd
- uint8_t numLen
- uint8_t remPartyNumber [81]

8.1343.1 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.

8.1343.2 Field Documentation

8.1343.2.1 uint8_t voice_remotePartyNum::numLen

8.1343.2.2 uint8_t voice_remotePartyNum::presentationInd

8.1343.2.3 uint8_t voice_remotePartyNum::remPartyNumber[81]

8.1344 voice_roamTimer Struct Reference

Data Fields

- uint8_t namID
- uint32_t roamTimerValue

8.1344.1 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.

8.1344.2 Field Documentation

8.1344.2.1 uint8_t voice_roamTimer::namID

8.1344.2.2 uint32_t voice_roamTimer::roamTimerValue

8.1345 voice_USSInfo Struct Reference

Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

8.1345.1 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

8.1345.2 Field Documentation

8.1345.2.1 uint8_t voice_USSInfo::ussData[182]

8.1345.2.2 uint8_t voice_USSInfo::ussDCS

8.1345.2.3 uint8_t voice_USSInfo::ussLen

8.1346 voice_UUSInfo Struct Reference

Data Fields

- uint8_t UUSType
- uint8_t UUSDcs
- uint8_t UUSDatalen
- uint8_t UUSData [255]

8.1346.1 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

8.1346.2 Field Documentation

8.1346.2.1 uint8_t voice_UUSInfo::UUSData[255]

8.1346.2.2 uint8_t voice_UUSInfo::UUSDatalen

8.1346.2.3 uint8_t voice_UUSInfo::UUSDcs

8.1346.2.4 uint8_t voice_UUSInfo::UUSType

8.1347 voiceALSSelectLineInfo Struct Reference

Data Fields

- BYTE lineValue

8.1347.1 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

8.1347.2 Field Documentation

8.1347.2.1 BYTE voiceALSSelectLineInfo::lineValue

8.1348 voiceALSSetLineSwitchInfo Struct Reference

Data Fields

- BYTE switchOption

8.1348.1 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

8.1348.2 Field Documentation

8.1348.2.1 BYTE voiceALSSetLineSwitchInfo::switchOption

8.1349 voiceAnswerCall Struct Reference

Data Fields

- BYTE * pCallId

8.1349.1 Detailed Description

Contains the parameters passed for SLQSVoiceAnswerCall.

Parameters

pCallId[IN/OUT]	• Unique call identifier for the call that must be answered.

8.1349.2 Field Documentation

8.1349.2.1 BYTE* voiceAnswerCall::pCallId

8.1350 voiceBindSubscriptionInfo Struct Reference

Data Fields

- BYTE subsType

8.1350.1 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

8.1350.2 Field Documentation

8.1350.2.1 BYTE voiceBindSubscriptionInfo::subsType

8.1351 voiceBurstDTMFInfo Struct Reference

Data Fields

- burstDTMFInfo BurstDTMFInfo
- DTMFLengths * pBurstDTMFLengths

8.1351.1 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

8.1351.2 Field Documentation

8.1351.2.1 burstDTMFInfo voiceBurstDTMFInfo::BurstDTMFInfo

8.1351.2.2 DTMFLengths* voiceBurstDTMFInfo::pBurstDTMFLengths

8.1352 voiceCallInfoReq Struct Reference

Data Fields

- BYTE callID

8.1352.1 Detailed Description

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

Parameters

callID	• Call identifier for the call queried for information.

8.1352.2 Field Documentation

8.1352.2.1 BYTE voiceCallInfoReq::callID

8.1353 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

8.1353.1 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

8.1353.2 Field Documentation

8.1353.2.1 ULONG* voiceCallInfoResp::pAlertingPattern

8.1353.2.2 BYTE* voiceCallInfoResp::pAlertType

8.1353.2.3 alphaIDInfo* voiceCallInfoResp::pAlphaIDInfo

8.1353.2.4 callInfo* voiceCallInfoResp::pCallInfo

8.1353.2.5 connectNumInfo* voiceCallInfoResp::pConnectNumInfo

8.1353.2.6 diagInfo* voiceCallInfoResp::pDiagInfo

8.1353.2.7 BYTE* voiceCallInfoResp::pOTASPStatus

8.1353.2.8 remotePartyName* voiceCallInfoResp::pRemotePartyName

8.1353.2.9 remotePartyNum* voiceCallInfoResp::pRemotePartyNum

8.1353.2.10 WORD* voiceCallInfoResp::pSrvOpt

8.1353.2.11 UUSInfo* voiceCallInfoResp::pUUSInfo

8.1353.2.12 BYTE* voiceCallInfoResp::pVoicePrivacy

8.1354 voiceCallRequestParams Struct Reference

Data Fields

- BYTE callNumber [81]
- BYTE * pCallType
- BYTE * pCLIRType
- UUSInfo * pUUSInFo
- CUGInfo * pCUGInfo
- BYTE * pEmergencyCategory

- calledPartySubAdd * pCallPartySubAdd
- ULONG * pSvcType

8.1354.1 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]
pCall-Type(optional)	• the type of call to be dialed. CALL_TYPE_VOICE is automatically selected if this parameter is not provided. When CALL_TYPE_NON_STD_OTASP is selected, the call is sent as a nonstandard OTASP call regardless of the digit string Call type values are: – 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
pCLIR-Type(optional)	• CLIR type values are: – 0x01 - CLIR_SUPPRESSION - Suppression – 0x02 - CLIR_INVOCATION - Invocation
pUUSIn-Fo(optional)	• Pointer to structure of UUSInfo – See UUSInfo for more information
pCUG-Info(optional)	• Pointer to structure of CUGInfo – See CUGInfo for more information
pEmergency-Category(optional)	• Bit mask of emergency number categories. This is only applicable when the call type is set to Emergency. – 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
pCallPartySub-Add(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

8.1354.2 Field Documentation

8.1354.2.1 BYTE voiceCallRequestParams::callNumber[81]

8.1354.2.2 calledPartySubAdd* voiceCallRequestParams::pCallPartySubAdd

8.1354.2.3 BYTE* voiceCallRequestParams::pCallType

8.1354.2.4 BYTE* voiceCallRequestParams::pCLIRType

8.1354.2.5 CUGInfo* voiceCallRequestParams::pCUGInfo

8.1354.2.6 BYTE* voiceCallRequestParams::pEmergencyCategory

8.1354.2.7 ULONG* voiceCallRequestParams::pSvcType

8.1354.2.8 UUSInfo* voiceCallRequestParams::pUUSInFo

8.1355 voiceCallResponseParams Struct Reference

Data Fields

- BYTE * pCallID
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- ccSUPSType * pCCSUPSType

8.1355.1 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

8.1355.2 Field Documentation

8.1355.2.1 alphaIDInfo∗ voiceCallResponseParams::pAlphaIDInfo

8.1355.2.2 BYTE∗ voiceCallResponseParams::pCallID

8.1355.2.3 BYTE∗ voiceCallResponseParams::pCCResultType

8.1355.2.4 ccSUPSType∗ voiceCallResponseParams::pCCSUPSType

8.1356 voiceContDTMFinfo Struct Reference

Data Fields

- BYTE ∗ pCallID
- BYTE DTMFdigit

8.1356.1 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.

8.1356.2 Field Documentation

8.1356.2.1 BYTE voiceContDTMFinfo::DTMFdigit

8.1356.2.2 BYTE* voiceContDTMFinfo::pCallID

8.1357 voiceDTMFEventInfo Struct Reference

Data Fields

- DTMFInfo DTMFInformation
- BYTE * pOnLength
- BYTE * pOffLength

8.1357.1 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

8.1357.2 Field Documentation

8.1357.2.1 DTMFInfo voiceDTMFEventInfo::DTMFInformation

8.1357.2.2 BYTE* voiceDTMFEventInfo::pOffLength

8.1357.2.3 BYTE* voiceDTMFEventInfo::pOnLength

8.1358 voiceFlashInfo Struct Reference

Data Fields

- BYTE * pCallID
- BYTE * pFlashPayLd
- BYTE * pFlashType

8.1358.1 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

8.1358.2 Field Documentation

8.1358.2.1 BYTE* voiceFlashInfo::pCallID

8.1358.2.2 BYTE* voiceFlashInfo::pFlashPayLd

8.1358.2.3 BYTE* voiceFlashInfo::pFlashType

8.1359 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

8.1359.1 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

pVoice-Privacy(optional)	• Voice Privacy. • Values. – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available
pArrCallEnd-Reason(optional)	• See arrCallEndReason for more information.
pArrAlphaI-D(optional)	• See arrAlphaID for more information.
pArrConnect-Party-Num(optional)	• See arrConnectPartyNum for more information.
pArrDiag-Info(optional)	• See arrDiagInfo for more information.
pArrCalledParty-Num(optional)	• See arrCalledPartyNum for more information.
pArrRedirParty-Num(optional)	• See arrRedirPartyNum for more information.
pArrAlerting-Pattern(optional)	• See arrAlertingPattern for more information.

8.1359.2 Field Documentation

8.1359.2.1 arrAlertingPattern∗ voiceGetAllCallInfo::pArrAlertingPattern

8.1359.2.2 arrAlertingType∗ voiceGetAllCallInfo::pArrAlertingType

8.1359.2.3 arrAlphaID∗ voiceGetAllCallInfo::pArrAlphaID

8.1359.2.4 arrCalledPartyNum∗ voiceGetAllCallInfo::pArrCalledPartyNum

8.1359.2.5 arrCallEndReason∗ voiceGetAllCallInfo::pArrCallEndReason

8.1359.2.6 arrCallInfo∗ voiceGetAllCallInfo::pArrCallInfo

8.1359.2.7 arrConnectPartyNum∗ voiceGetAllCallInfo::pArrConnectPartyNum

8.1359.2.8 arrDiagInfo∗ voiceGetAllCallInfo::pArrDiagInfo

8.1359.2.9 arrRedirPartyNum∗ voiceGetAllCallInfo::pArrRedirPartyNum

8.1359.2.10 arrRemotePartyName∗ voiceGetAllCallInfo::pArrRemotePartyName

8.1359.2.11 arrRemotePartyNum∗ voiceGetAllCallInfo::pArrRemotePartyNum

8.1359.2.12 arrSvcOption* voiceGetAllCallInfo::pArrSvcOption

8.1359.2.13 arrUUSInfo* voiceGetAllCallInfo::pArrUUSInfo

8.1359.2.14 BYTE* voiceGetAllCallInfo::pOTASPStatus

8.1359.2.15 BYTE* voiceGetAllCallInfo::pVoicePrivacy

8.1360 voiceGetCallBarringReq Struct Reference

Data Fields

- BYTE reason
- BYTE * pSvcClass

8.1360.1 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

8.1360.2 Field Documentation

8.1360.2.1 BYTE* voiceGetCallBarringReq::pSvcClass

8.1360.2.2 BYTE voiceGetCallBarringReq::reason

8.1361 voiceGetCallBarringResp Struct Reference

Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1361.1 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.

8.1361.2 Field Documentation

8.1361.2.1 alphaIDInfo* voiceGetCallBarringResp::pAlphaIDInfo

8.1361.2.2 BYTE* voiceGetCallBarringResp::pCallID

8.1361.2.3 BYTE* voiceGetCallBarringResp::pCCResType

8.1361.2.4 ccSUPSType* voiceGetCallBarringResp::pCCSUPSType

8.1361.2.5 WORD* voiceGetCallBarringResp::pFailCause

8.1361.2.6 BYTE* voiceGetCallBarringResp::pSvcClass

8.1362 voiceGetCallFWReq Struct Reference

Data Fields

- BYTE Reason
- BYTE * pSvcClass

8.1362.1 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.

8.1362.2 Field Documentation

8.1362.2.1 BYTE* voiceGetCallFWReq::pSvcClass

8.1362.2.2 BYTE voiceGetCallFWReq::Reason

8.1363 voiceGetCallFWResp Struct Reference

Data Fields

- getCallFWInfo * pGetCallFWInfo
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType
- getCallFWExtInfo * pGetCallFWExtInfo

8.1363.1 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.

8.1363.2 Field Documentation

8.1363.2.1 alphaIDInfo* voiceGetCallFWResp::pAlphaIDInfo

8.1363.2.2 BYTE* voiceGetCallFWResp::pCallID

8.1363.2.3 BYTE* voiceGetCallFWResp::pCCResType

8.1363.2.4 ccSUPSType* voiceGetCallFWResp::pCCSUPSType

8.1363.2.5 WORD* voiceGetCallFWResp::pFailCause

8.1363.2.6 getCallFWExtInfo* voiceGetCallFWResp::pGetCallFWExtInfo

8.1363.2.7 getCallFWInfo* voiceGetCallFWResp::pGetCallFWInfo

8.1364 voiceGetCallWaitInfo Struct Reference

Data Fields

- BYTE * pSvcClass
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1364.1 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.

8.1364.2 Field Documentation

8.1364.2.1 alphaIDInfo* voiceGetCallWaitInfo::pAlphaIDInfo

8.1364.2.2 BYTE* voiceGetCallWaitInfo::pCallID

8.1364.2.3 BYTE* voiceGetCallWaitInfo::pCCResType

8.1364.2.4 ccSUPSType* voiceGetCallWaitInfo::pCCSUPSType

8.1364.2.5 WORD* voiceGetCallWaitInfo::pFailCause

8.1364.2.6 BYTE* voiceGetCallWaitInfo::pSvcClass

8.1365 voiceGetCLIPResp Struct Reference

Data Fields

- CLIPResp * pCLIPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1365.1 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.

8.1365.2 Field Documentation

8.1365.2.1 alphaIDInfo* voiceGetCLIPResp::pAlphaIDInfo

8.1365.2.2 BYTE* voiceGetCLIPResp::pCallID

8.1365.2.3 BYTE* voiceGetCLIPResp::pCCResType

8.1365.2.4 ccSUPSType* voiceGetCLIPResp::pCCSUPSType

8.1365.2.5 CLIPResp* voiceGetCLIPResp::pCLIPResp

8.1365.2.6 WORD* voiceGetCLIPResp::pFailCause

8.1366 voiceGetCLIRResp Struct Reference

Data Fields

- CLIRResp * pCLIRResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1366.1 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.

8.1366.2 Field Documentation

8.1366.2.1 alphaIDInfo* voiceGetCLIRResp::pAlphaIDInfo

8.1366.2.2 BYTE* voiceGetCLIRResp::pCallID

8.1366.2.3 BYTE* voiceGetCLIRResp::pCCResType

8.1366.2.4 ccSUPSType* voiceGetCLIRResp::pCCSUPSType

8.1366.2.5 CLIRResp* voiceGetCLIRResp::pCLIRResp

8.1366.2.6 WORD* voiceGetCLIRResp::pFailCause

8.1367 voiceGetCNAPResp Struct Reference

Data Fields

- CNAPResp * pCNAPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1367.1 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.

8.1367.2 Field Documentation

8.1367.2.1 alphaIDInfo* voiceGetCNAPResp::pAlphaIDInfo

8.1367.2.2 BYTE* voiceGetCNAPResp::pCallID

8.1367.2.3 BYTE* voiceGetCNAPResp::pCCResType

8.1367.2.4 ccSUPSType* voiceGetCNAPResp::pCCSUPSType

8.1367.2.5 CNAPResp* voiceGetCNAPResp::pCNAPResp

8.1367.2.6 WORD* voiceGetCNAPResp::pFailCause

8.1368 voiceGetCOLPResp Struct Reference

Data Fields

- COLPResp * pCOLPResp
- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1368.1 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.

8.1368.2 Field Documentation

8.1368.2.1 alphaIDInfo∗ voiceGetCOLPResp::pAlphaIDInfo

8.1368.2.2 BYTE∗ voiceGetCOLPResp::pCallID

8.1368.2.3 BYTE∗ voiceGetCOLPResp::pCCResType

8.1368.2.4 ccSUPSType∗ voiceGetCOLPResp::pCCSUPSType

8.1368.2.5 COLPResp∗ voiceGetCOLPResp::pCOLPResp

8.1368.2.6 WORD∗ voiceGetCOLPResp::pFailCause

8.1369 voiceGetCOLRResp Struct Reference

Data Fields

- COLRResp ∗ pCOLRResp
- WORD ∗ pFailCause
- alphaIDInfo ∗ pAlphaIDInfo
- BYTE ∗ pCCResType
- BYTE ∗ pCallID
- ccSUPSType ∗ pCCSUPSType

8.1369.1 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.

8.1369.2 Field Documentation

8.1369.2.1 alphaIDInfo* voiceGetCOLRResp::pAlphaIDInfo

8.1369.2.2 BYTE* voiceGetCOLRResp::pCallID

8.1369.2.3 BYTE* voiceGetCOLRResp::pCCResType

8.1369.2.4 ccSUPSType* voiceGetCOLRResp::pCCSUPSType

8.1369.2.5 COLRResp* voiceGetCOLRResp::pCOLRResp

8.1369.2.6 WORD* voiceGetCOLRResp::pFailCause

8.1370 voiceGetConfigReq Struct Reference

Data Fields

- BYTE * pAutoAnswer
- BYTE * pAirTimer
- BYTE * pRoamTimer
- BYTE * pTTYMode
- BYTE * pPrefVoiceSO
- BYTE * pAMRStatus
- BYTE * pPrefVoicePrivacy
- BYTE * pNamID
- BYTE * pVoiceDomainPref

8.1370.1 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.

8.1370.2 Field Documentation

8.1370.2.1 BYTE* voiceGetConfigReq::pAirTimer

8.1370.2.2 BYTE* voiceGetConfigReq::pAMRStatus

8.1370.2.3 BYTE* voiceGetConfigReq::pAutoAnswer

8.1370.2.4 BYTE* voiceGetConfigReq::pNamID

8.1370.2.5 BYTE* voiceGetConfigReq::pPrefVoicePrivacy

8.1370.2.6 BYTE* voiceGetConfigReq::pPrefVoiceSO

8.1370.2.7 BYTE* voiceGetConfigReq::pRoamTimer

8.1370.2.8 BYTE* voiceGetConfigReq::pTTYMode

8.1370.2.9 BYTE* voiceGetConfigReq::pVoiceDomainPref

8.1371 voiceGetConfigResp Struct Reference

Data Fields

- BYTE * pAutoAnswerStat
- airTimer * pAirTimerCnt
- roamTimer * pRoamTimerCnt
- BYTE * pCurrTTYMode
- prefVoiceSO * pCurPrefVoiceSO
- curAMRConfig * pCurAMRConfig
- BYTE * pCurVoicePrivacyPref
- BYTE * pCurVoiceDomainPref

8.1371.1 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.

8.1371.2 Field Documentation

8.1371.2.1 airTimer∗ voiceGetConfigResp::pAirTimerCnt

8.1371.2.2 BYTE∗ voiceGetConfigResp::pAutoAnswerStat

8.1371.2.3 curAMRConfig∗ voiceGetConfigResp::pCurAMRConfig

8.1371.2.4 prefVoiceSO∗ voiceGetConfigResp::pCurPrefVoiceSO

8.1371.2.5 BYTE∗ voiceGetConfigResp::pCurrTTYMode

8.1371.2.6 BYTE∗ voiceGetConfigResp::pCurVoiceDomainPref

8.1371.2.7 BYTE∗ voiceGetConfigResp::pCurVoicePrivacyPref

8.1371.2.8 roamTimer∗ voiceGetConfigResp::pRoamTimerCnt

8.1372 voiceIndicationRegisterInfo Struct Reference

Data Fields

- BYTE ∗ pRegDTMFEvents
- BYTE ∗ pRegVoicePrivacyEvents
- BYTE ∗ pSuppsNotifEvents

8.1372.1 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.

8.1372.2 Field Documentation

8.1372.2.1 BYTE* voiceIndicationRegisterInfo::pRegDTMFEvents

8.1372.2.2 BYTE* voiceIndicationRegisterInfo::pRegVoicePrivacyEvents

8.1372.2.3 BYTE* voiceIndicationRegisterInfo::pSuppsNotifEvents

8.1373 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

8.1373.1 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

8.1373.2 Field Documentation

8.1373.2.1 BYTE voiceInfoRec::callID

8.1373.2.2 calledPartyInfo* voiceInfoRec::pCalledPartyInfo

8.1373.2.3 callerIDInfo* voiceInfoRec::pCallerIDInfo

8.1373.2.4 BYTE* voiceInfoRec::pCallerNameInfo

8.1373.2.5 connectNumInfo* voiceInfoRec::pCallingPartyInfo

8.1373.2.6 BYTE* voiceInfoRec::pCallWaitInd

8.1373.2.7 BYTE* voiceInfoRec::pCLIRCause

8.1373.2.8 connectNumInfo* voiceInfoRec::pConnectNumInfo

8.1373.2.9 BYTE* voiceInfoRec::pDispInfo

8.1373.2.10 BYTE* voiceInfoRec::pExtDispInfo

8.1373.2.11 extDispRecInfo* voiceInfoRec::pExtDispRecInfo

8.1373.2.12 lineCtrlInfo* voiceInfoRec::pLineCtrlInfo

8.1373.2.13 NSSAudioCtrl* voiceInfoRec::pNSSAudioCtrl

8.1373.2.14 BYTE* voiceInfoRec::pNSSRelease

8.1373.2.15 redirNumInfo* voiceInfoRec::pRedirNumInfo

8.1373.2.16 signalInfo* voiceInfoRec::pSignalInfo

8.1374 voiceManageCallsReq Struct Reference

Data Fields

- BYTE SUPSType
- BYTE * pCallID

8.1374.1 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

8.1374.2 Field Documentation

8.1374.2.1 BYTE* voiceManageCallsReq::pCallID

8.1374.2.2 BYTE voiceManageCallsReq::SUPSType

8.1375 voiceManageCallsResp Struct Reference

Data Fields

- WORD * pFailCause

8.1375.1 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

8.1375.2 Field Documentation

8.1375.2.1 WORD* voiceManageCallsResp::pFailCause

8.1376 voiceOrigUSSDNoWaitInfo Struct Reference

Data Fields

- struct USSInfo USSInformation

8.1376.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

Parameters

USSInformation	• See USSInfo for more information.

8.1376.2 Field Documentation

8.1376.2.1 struct USSInfo voiceOrigUSSDNoWaitInfo::USSInformation

8.1377 voiceOTASPStatusInfo Struct Reference

Data Fields

- BYTE callID
- BYTE OTASPStatus

8.1377.1 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)

8.1377.2 Field Documentation

8.1377.2.1 BYTE voiceOTASPStatusInfo::callID

8.1377.2.2 BYTE voiceOTASPStatusInfo::OTASPStatus

8.1378 voicePrivacyInfo Struct Reference

Data Fields

- BYTE callID
- BYTE voicePrivacy

8.1378.1 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

8.1378.2 Field Documentation

8.1378.2.1 BYTE voicePrivacyInfo::callID

8.1378.2.2 BYTE voicePrivacyInfo::voicePrivacy

8.1379 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

8.1379.1 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.

8.1379.2 Field Documentation

8.1379.2.1 arrCallInfo voiceSetAllCallStatusCbkInfo::arrCallInfomation

8.1379.2.2 arrAlertingPattern* voiceSetAllCallStatusCbkInfo::pArrAlertingPattern

8.1379.2.3 arrAlertingType* voiceSetAllCallStatusCbkInfo::pArrAlertingType

8.1379.2.4 arrAlphaID* voiceSetAllCallStatusCbkInfo::pArrAlphaID

8.1379.2.5 arrCalledPartyNum* voiceSetAllCallStatusCbkInfo::pArrCalledPartyNum

8.1379.2.6 arrCallEndReason* voiceSetAllCallStatusCbkInfo::pArrCallEndReason

8.1379.2.7 arrConnectPartyNum* voiceSetAllCallStatusCbkInfo::pArrConnectPartyNum

8.1379.2.8 arrDiagInfo* voiceSetAllCallStatusCbkInfo::pArrDiagInfo

8.1379.2.9 arrRedirPartyNum* voiceSetAllCallStatusCbkInfo::pArrRedirPartyNum

8.1379.2.10 arrRemotePartyName* voiceSetAllCallStatusCbkInfo::pArrRemotePartyName

8.1379.2.11 arrRemotePartyNum* voiceSetAllCallStatusCbkInfo::pArrRemotePartyNum

8.1379.2.12 arrSvcOption* voiceSetAllCallStatusCbkInfo::pArrSvcOption

8.1380 voiceSetCallBarringPwdInfo Struct Reference

Data Fields

- BYTE Reason
- BYTE oldPasswd [4]
- BYTE newPasswd [4]
- BYTE newPasswdAgain [4]

8.1380.1 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.

8.1380.2 Field Documentation

8.1380.2.1 BYTE voiceSetCallBarringPwdInfo::newPasswd[4]

8.1380.2.2 BYTE voiceSetCallBarringPwdInfo::newPasswdAgain[4]

8.1380.2.3 BYTE voiceSetCallBarringPwdInfo::oldPasswd[4]

8.1380.2.4 BYTE voiceSetCallBarringPwdInfo::Reason

8.1381 voiceSetCallBarringPwdResp Struct Reference

Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1381.1 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.

8.1381.2 Field Documentation

8.1381.2.1 alphaIDInfo* voiceSetCallBarringPwdResp::pAlphaIDInfo

8.1381.2.2 BYTE* voiceSetCallBarringPwdResp::pCallID

8.1381.2.3 BYTE* voiceSetCallBarringPwdResp::pCCResType

8.1381.2.4 ccSUPSType* voiceSetCallBarringPwdResp::pCCSUPSType

8.1381.2.5 WORD* voiceSetCallBarringPwdResp::pFailCause

8.1382 voiceSetConfigReq Struct Reference

Data Fields

- BYTE * pAutoAnswer

- airTimer * pAirTimerConfig
- roamTimer * pRoamTimerConfig
- BYTE * pTTYMode
- prefVoiceSO * pPrefVoiceSO
- BYTE * pPrefVoiceDomain

8.1382.1 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.

8.1382.2 Field Documentation

8.1382.2.1 airTimer* voiceSetConfigReq::pAirTimerConfig

8.1382.2.2 BYTE* voiceSetConfigReq::pAutoAnswer

8.1382.2.3 BYTE* voiceSetConfigReq::pPrefVoiceDomain

8.1382.2.4 prefVoiceSO* voiceSetConfigReq::pPrefVoiceSO

8.1382.2.5 roamTimer* voiceSetConfigReq::pRoamTimerConfig

8.1382.2.6 BYTE* voiceSetConfigReq::pTTYMode

8.1383 voiceSetConfigResp Struct Reference

Data Fields

- BYTE * pAutoAnsStatus
- BYTE * pAirTimerStatus
- BYTE * pRoamTimerStatus
- BYTE * pTTYConfigStatus
- BYTE * pPrefVoiceSOStatus
- BYTE * pVoiceDomainPrefStatus

8.1383.1 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.

8.1383.2 Field Documentation

8.1383.2.1 BYTE* voiceSetConfigResp::pAirTimerStatus

8.1383.2.2 BYTE* voiceSetConfigResp::pAutoAnsStatus

8.1383.2.3 BYTE* voiceSetConfigResp::pPrefVoiceSOStatus

8.1383.2.4 BYTE* voiceSetConfigResp::pRoamTimerStatus

8.1383.2.5 BYTE* voiceSetConfigResp::pTTYConfigStatus

8.1383.2.6 BYTE* voiceSetConfigResp::pVoiceDomainPrefStatus

8.1384 voiceSetPrefPrivacy Struct Reference

Data Fields

- BYTE privacyPref

8.1384.1 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

8.1384.2 Field Documentation

8.1384.2.1 BYTE voiceSetPrefPrivacy::privacyPref

8.1385 voiceSetSUPSServiceReq Struct Reference

Data Fields

- BYTE voiceSvc
- BYTE reason
- BYTE * pServiceClass
- BYTE * pCallBarringPasswd
- BYTE * pCallForwardingNumber
- BYTE * pTimerVal
- callFwdTypeAndPlan * pCallFwdTypeAndPlan

8.1385.1 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

8.1385.2 Field Documentation

8.1385.2.1 BYTE* voiceSetSUPSServiceReq::pCallBarringPasswd

8.1385.2.2 BYTE* voiceSetSUPSServiceReq::pCallForwardingNumber

8.1385.2.3 callFwdTypeAndPlan* voiceSetSUPSServiceReq::pCallFwdTypeAndPlan

8.1385.2.4 BYTE* voiceSetSUPSServiceReq::pServiceClass

8.1385.2.5 BYTE* voiceSetSUPSServiceReq::pTimerVal

8.1385.2.6 BYTE voiceSetSUPSServiceReq::reason

8.1385.2.7 BYTE voiceSetSUPSServiceReq::voiceSvc

8.1386 voiceSetSUPSServiceResp Struct Reference

Data Fields

- WORD * pFailCause
- alphaIDInfo * pAlphaIDInfo
- BYTE * pCCResultType
- BYTE * pCallID
- ccSUPSType * pCCSUPSType

8.1386.1 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

8.1386.2 Field Documentation

8.1386.2.1 alphaIDInfo* voiceSetSUPSServiceResp::pAlphaIDInfo

8.1386.2.2 BYTE* voiceSetSUPSServiceResp::pCallID

8.1386.2.3 BYTE* voiceSetSUPSServiceResp::pCCResultType

8.1386.2.4 ccSUPSType* voiceSetSUPSServiceResp::pCCSUPSType

8.1386.2.5 WORD* voiceSetSUPSServiceResp::pFailCause

8.1387 voiceStopContDTMFinfo Struct Reference

Data Fields

- BYTE callID

8.1387.1 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

8.1387.2 Field Documentation

8.1387.2.1 BYTE voiceStopContDTMFinfo::callID

8.1388 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

8.1388.1 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

8.1388.2 Field Documentation

8.1388.2.1 alphaIDInfo* voiceSUPSInfo::pAlphaIDInfo

8.1388.2.2 BYTE* voiceSUPSInfo::pCallBarPasswd

8.1388.2.3 getCallFWInfo* voiceSUPSInfo::pCallFwdInfo

8.1388.2.4 BYTE* voiceSUPSInfo::pCallFWNum

8.1388.2.5 BYTE* voiceSUPSInfo::pCallFWTimerVal

8.1388.2.6 BYTE* voiceSUPSInfo::pCallID

8.1388.2.7 CLIPResp* voiceSUPSInfo::pCLIPstatus

8.1388.2.8 CLIRResp* voiceSUPSInfo::pCLIRstatus

8.1388.2.9 CNAPResp* voiceSUPSInfo::pCNAPstatus

8.1388.2.10 COLPResp* voiceSUPSInfo::pCOLPstatus

8.1388.2.11 COLRResp* voiceSUPSInfo::pCOLRstatus

8.1388.2.12 BYTE* voiceSUPSInfo::pDataSrc

8.1388.2.13 WORD* voiceSUPSInfo::pFailCause

8.1388.2.14 newPwdData* voiceSUPSInfo::pNewPwdData

8.1388.2.15 BYTE* voiceSUPSInfo::pReason

8.1388.2.16 BYTE* voiceSUPSInfo::pSvcClass

8.1388.2.17 struct USSInfo* voiceSUPSInfo::pUSSInfo

8.1388.2.18 SUPSInfo voiceSUPSInfo::SUPSInformation

8.1389 voiceSUPSNotification Struct Reference

Data Fields

- BYTE callID
- BYTE notifType
- WORD * pCUGIndex
- ECTNum * pECTNum

8.1389.1 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

8.1389.2 Field Documentation

8.1389.2.1 BYTE voiceSUPSNotification::callID

8.1389.2.2 BYTE voiceSUPSNotification::notifType

8.1389.2.3 WORD* voiceSUPSNotification::pCUGIndex

8.1389.2.4 ECTNum* voiceSUPSNotification::pECTNum

8.1390 wcdmaCellInfo Struct Reference

Data Fields

- WORD psc
- SHORT cpich_rscp
- SHORT cpich_ecno
- SHORT srxlev

8.1390.1 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.

8.1390.2 Field Documentation

8.1390.2.1 SHORT wcdmaCellInfo::cpich_ecno

8.1390.2.2 SHORT wcdmaCellInfo::cpich_rscp

8.1390.2.3 WORD wcdmaCellInfo::psc

8.1390.2.4 SHORT wcdmaCellInfo::srxlev

8.1391 WCDMAECIOThresh Struct Reference

Data Fields

- BYTE WCDMAECIOThreshListLen
- WORD * pWCDMAECIOThreshList

8.1391.1 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)

8.1391.2 Field Documentation

8.1391.2.1 WORD* WCDMAECIOThresh::pWCDMAECIOThreshList

8.1391.2.2 BYTE WCDMAECIOThresh::WCDMAECIOThreshListLen

8.1392 WCDMAInfoLTENeighborCell Struct Reference

Data Fields

- ULONG wcdmaRRCState
- BYTE umtsLTENbrCellLen
- umtsLTENbrCell UMTSLTENbrCell [255]

8.1392.1 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.

8.1392.2 Field Documentation

8.1392.2.1 umtsLTENbrCell WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.1392.2.2 BYTE WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.1392.2.3 ULONG WCDMAInfoLTENeighborCell::wcdmaRRCState

8.1393 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

8.1393.1 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.

8.1393.2 Field Documentation

8.1393.2.1 BYTE wcdmaLongMsgDecodingParams::Date[0x09]

8.1393.2.2 BOOL* wcdmaLongMsgDecodingParams::pIsUDHPresent

8.1393.2.3 BYTE* wcdmaLongMsgDecodingParams::pMessage

8.1393.2.4 BYTE* wcdmaLongMsgDecodingParams::pPartNum

8.1393.2.5 BYTE* wcdmaLongMsgDecodingParams::pReferenceNum

8.1393.2.6 CHAR* wcdmaLongMsgDecodingParams::pScAddr

8.1393.2.7 BYTE* wcdmaLongMsgDecodingParams::pScAddrLength

8.1393.2.8 CHAR* wcdmaLongMsgDecodingParams::pSenderAddr

8.1393.2.9 BYTE* wcdmaLongMsgDecodingParams::pSenderAddrLength

8.1393.2.10 CHAR* wcdmaLongMsgDecodingParams::pTextMsg

8.1393.2.11 BYTE* wcdmaLongMsgDecodingParams::pTextMsgLength

8.1393.2.12 BYTE* wcdmaLongMsgDecodingParams::pTotalNum

8.1393.2.13 BYTE wcdmaLongMsgDecodingParams::Time[0x09]

8.1394 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]

8.1394.1 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

8.1394.2 Field Documentation

8.1394.2.1 BYTE wcdmaMsgDecodingParams::Date[0x09]

8.1394.2.2 BYTE* wcdmaMsgDecodingParams::pMessage

8.1394.2.3 CHAR* wcdmaMsgDecodingParams::pScAddr

8.1394.2.4 BYTE* wcdmaMsgDecodingParams::pScAddrLength

8.1394.2.5 CHAR* wcdmaMsgDecodingParams::pSenderAddr

8.1394.2.6 BYTE* wcdmaMsgDecodingParams::pSenderAddrLength

8.1394.2.7 CHAR* wcdmaMsgDecodingParams::pTextMsg

8.1394.2.8 BYTE* wcdmaMsgDecodingParams::pTextMsgLength

8.1394.2.9 BYTE wcdmaMsgDecodingParams::Time[0x09]

8.1395 wcdmaMsgEncodingParams Struct Reference

Data Fields

- ULONG messageSize
- CHAR * pDestAddr
- CHAR * pTextMsg
- CHAR * pPDUMessage
- BYTE alphabet

8.1395.1 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

8.1395.2 Field Documentation

8.1395.2.1 BYTE wcdmaMsgEncodingParams::alphabet

8.1395.2.2 ULONG wcdmaMsgEncodingParams::messageSize

8.1395.2.3 CHAR* wcdmaMsgEncodingParams::pDestAddr

8.1395.2.4 CHAR* wcdmaMsgEncodingParams::pPDUMessage

8.1395.2.5 CHAR* wcdmaMsgEncodingParams::pTextMsg

8.1396 WCDMARSSIThresh Struct Reference

Data Fields

- BYTE WCDMARSSIThreshListLen
- WORD * pWCDMARSSIThreshList

8.1396.1 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)

8.1396.2 Field Documentation

8.1396.2.1 WORD* WCDMARSSIThresh::pWCDMARSSIThreshList

8.1396.2.2 BYTE WCDMARSSIThresh::WCDMARSSIThreshListLen

8.1397 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

8.1397.1 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

8.1397.2 Field Documentation

8.1397.2.1 ULONG WCDMASysInfo::cellId

8.1397.2.2 BYTE WCDMASysInfo::cellIdValid

8.1397.2.3 BYTE WCDMASysInfo::hsCallStatus

8.1397.2.4 BYTE WCDMASysInfo::hsCallStatusValid

8.1397.2.5 BYTE WCDMASysInfo::hsInd

8.1397.2.6 BYTE WCDMASysInfo::hsIndValid

8.1397.2.7 WORD WCDMASysInfo::lac

8.1397.2.8 BYTE WCDMASysInfo::lacValid

8.1397.2.9 BYTE WCDMASysInfo::MCC[3]

8.1397.2.10 BYTE WCDMASysInfo::MNC[3]

8.1397.2.11 BYTE WCDMASysInfo::networkIdValid

8.1397.2.12 WORD WCDMASysInfo::psc

8.1397.2.13 BYTE WCDMASysInfo::pscValid

8.1397.2.14 BYTE WCDMASysInfo::regRejectInfoValid

8.1397.2.15 BYTE WCDMASysInfo::rejCause

8.1397.2.16 BYTE WCDMASysInfo::rejectSrvDomain

8.1397.2.17 sysInfoCommon WCDMASysInfo::sysInfoWCDMA

8.1398 wcdmaUARFCN Struct Reference

Data Fields

- BYTE status
- ULONG uarfcn

8.1398.1 Detailed Description

This structure contains the parameters for WCDMA UARFCN.

Parameters

status	• 0 - Disable • 1 - Enable
uarfcn	• UARFCN to which UE is locked

8.1398.2 Field Documentation

8.1398.2.1 BYTE wcdmaUARFCN::status

8.1398.2.2 ULONG wcdmaUARFCN::uarfcn

8.1399 wds_currNetworkInfo Struct Reference

Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

8.1399.1 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

8.1399.2 Field Documentation

8.1399.2.1 uint8_t wds_currNetworkInfo::NetworkType

8.1399.2.2 uint32_t wds_currNetworkInfo::RATMask

8.1399.2.3 uint32_t wds_currNetworkInfo::SOMask

8.1400 wds_DataULongLongTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint64_t ullData

8.1400.1 Detailed Description

Parameters

TlvPresent	a flag indicates whether TLV presented or not
ulData	unsigend long long data in 64bit

8.1400.2 Field Documentation

8.1400.2.1 uint8_t wds_DataULongLongTlv::TlvPresent

8.1400.2.2 uint64_t wds_DataULongLongTlv::ullData

8.1401 wds_DataULongTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t ulData

8.1401.1 Detailed Description

Parameters

TlvPresent	a flag indicates whether TLV presented or not
ulData	unsigend long data in 32bit

8.1401.2 Field Documentation

8.1401.2.1 uint8_t wds_DataULongTlv::TlvPresent

8.1401.2.2 uint32_t wds_DataULongTlv::ulData

8.1402 wds_DHCPLeaseOptTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t numOpt
- wds_DHCPOpt optList [30]
- uint8_t optListData [2048]

8.1402.1 Detailed Description

DHCP lease option information

Parameters

TlvPresent	a flag indicates whether TLV presented or not
numOpt	number of sets of wds_DHCPOpt
optList	option list
optListData	option list data

8.1402.2 Field Documentation

8.1402.2.1 uint8_t wds_DHCPLeaseOptTlv::numOpt

8.1402.2.2 wds_DHCPOpt wds_DHCPLeaseOptTlv::optList[30]

8.1402.2.3 uint8_t wds_DHCPLeaseOptTlv::optListData[2048]

8.1402.2.4 uint8_t wds_DHCPLeaseOptTlv::TlvPresent

8.1403 wds_DHCPLeaseStateTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t leaseState

8.1403.1 Detailed Description

DHCP lease state information

Parameters

TlvPresent	a flag indicates whether TLV presented or not
leaseState	lease state • 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

8.1403.2 Field Documentation

8.1403.2.1 uint8_t wds_DHCPLeaseStateTlv::leaseState

8.1403.2.2 uint8_t wds_DHCPLeaseStateTlv::TlvPresent

8.1404 wds_DHCPOpt Struct Reference

Data Fields

- uint8_t optCode
- uint8_t optValLen
- uint8_t * pOptVal

8.1404.1 Detailed Description

DHCP option code information

Parameters

optCode	option code • values: – 0 - 255
optValLen	length of option code • values: – 0 - 255
pOptVal	option value

8.1404.2 Field Documentation

8.1404.2.1 uint8_t wds_DHCPOpt::optCode

8.1404.2.2 uint8_t wds_DHCPOpt::optValLen

8.1404.2.3 uint8_t* wds_DHCPOpt::pOptVal

8.1405 wds_DHCPProfileIdTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint8_t profileType
- uint8_t profileId

8.1405.1 Detailed Description

DHCP profile id TLV information

Parameters

TlvPresent	a flag indicates whether TLV presented or not
profileType	identifying the type of the profile • 0 - 3GPP
profileId	index identifying the profile • 1-24 valid for 3GPP profile type (9x30 and onwards)

8.1405.2 Field Documentation

8.1405.2.1 uint8_t wds_DHCPProfileIdTlv::profileId

8.1405.2.2 uint8_t wds_DHCPProfileIdTlv::profileType

8.1405.2.3 uint8_t wds_DHCPProfileIdTlv::TlvPresent

8.1406 wds_DHCPv4HWConfig Struct Reference

Data Fields

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

8.1406.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

Parameters

hwType	• HW Type 1 - Ethernet 20 - Serial
chaddrlen	• chaddrlen
chaddr	• chaddr. Max size 16 bytes

8.1406.2 Field Documentation

8.1406.2.1 uint8_t wds_DHCPv4HWConfig::chaddr[16]

8.1406.2.2 uint8_t wds_DHCPv4HWConfig::chaddrLen

8.1406.2.3 uint8_t wds_DHCPv4HWConfig::hwType

8.1407 wds_DHCPv4Option Struct Reference

Data Fields

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

8.1407.1 Detailed Description

WDS SWI DHCPv4 Option Structure

Parameters

optCode	• Option code – 0 - 255
optValLen	• Option value length – 0 - 255
optVal	• Option value

8.1407.2 Field Documentation

8.1407.2.1 uint8_t wds_DHCPv4Option::optCode

8.1407.2.2 uint8_t wds_DHCPv4Option::optVal[255]

8.1407.2.3 uint8_t wds_DHCPv4Option::optValLen

8.1408 wds_DHCPv4OptionList Struct Reference

Data Fields

- uint8_t numOpt
- wds_DHCPv4Option * pOptList

8.1408.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

Parameters

numOpt	• number of options – 0 - 255
pOptList	• pointer to list of DHCP Options

8.1408.2 Field Documentation

8.1408.2.1 uint8_t wds_DHCPv4OptionList::numOpt

8.1408.2.2 wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList

8.1409 wds_DHCPv4ProfileId Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileId

8.1409.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

Parameters

profileType	• 0 for 3GPP
profileId	• 1 to 24 for 3GPP profile

8.1409.2 Field Documentation

8.1409.2.1 uint8_t wds_DHCPv4ProfileId::profileId

8.1409.2.2 uint8_t wds_DHCPv4ProfileId::profileType

8.1410 wds_Domain Struct Reference

Data Fields

- uint16_t domainLen
- uint8_t domainName [256]

8.1410.1 Detailed Description

This structure contains the DomainName Information

Parameters

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

8.1410.2 Field Documentation

8.1410.2.1 uint16_t wds_Domain::domainLen

8.1410.2.2 uint8_t wds_Domain::domainName[256]

8.1411 wds_DomainNameList Struct Reference

Data Fields

- uint8_t numInstances
- struct wds_Domain domain [10]

8.1411.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

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

8.1411.2 Field Documentation

8.1411.2.1 struct wds_Domain wds_DomainNameList::domain[10]

8.1411.2.2 uint8_t wds_DomainNameList::numInstances

8.1412 wds_GPRSQoS Struct Reference

Data Fields

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

8.1412.1 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

8.1412.2 Field Documentation

8.1412.2.1 uint32_t wds_GPRSQoS::delayClass

8.1412.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

8.1412.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

8.1412.2.4 uint32_t wds_GPRSQoS::precedenceClass

8.1412.2.5 uint32_t wds_GPRSQoS::reliabilityClass

8.1413 wds_IPv4AdTlv Struct Reference

Data Fields

- uint8_t TlvPresent
- uint32_t IPv4Addr

8.1413.1 Detailed Description

IPv4 address TLV information

Parameters

TlvPresent	a flag indicates whether TLV presented or not
IPv4Addr	IPv4 address

8.1413.2 Field Documentation

8.1413.2.1 uint32_t wds_IPv4AdTlv::IPv4Addr

8.1413.2.2 uint8_t wds_IPv4AdTlv::TlvPresent

8.1414 wds_IPV6AddressInfo Struct Reference

Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

8.1414.1 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.

8.1414.2 Field Documentation

8.1414.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

8.1414.2.2 uint8_t wds_IPV6AddressInfo::IPV6PrefixLen

8.1415 wds_IPV6GWAddressInfo Struct Reference

Data Fields

- uint8_t gwV6PrefixLen
- uint16_t gwAddressV6 [8]

8.1415.1 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.

8.1415.2 Field Documentation

8.1415.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

8.1415.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

8.1416 wds_PCSCFFQDNAddress Struct Reference

Data Fields

- uint16_t fqdnLen
- uint8_t fqdnAddr [256]

8.1416.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

Parameters

fqdnLen	• length of the received FQDN address
fqdnAddr	• FQDN address(Max 256 characters)

8.1416.2 Field Documentation

8.1416.2.1 uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]

8.1416.2.2 uint16_t wds_PCSCFFQDNAddress::fqdnLen

8.1417 wds_PCSCFFQDNAddressList Struct Reference

Data Fields

- uint8_t numInstances
- struct wds_PCSCFFQDNAddress pcsfFQDNAddress [10]

8.1417.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

Parameters

numInstances	• Number of FQDN addresses received
pcsfFQDN-Address	• FQDN address information(Max 10 addresses)

8.1417.2 Field Documentation

8.1417.2.1 uint8_t wds_PCSCFFQDNAddressList::numInstances

8.1417.2.2 struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]

8.1418 wds_PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- uint8_t numInstances
- uint32_t pscsfIPv4Addr [64]

8.1418.1 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)

8.1418.2 Field Documentation

8.1418.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

8.1418.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

8.1419 wds_ProfileIdentifier Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileIndex

8.1419.1 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

8.1419.2 Field Documentation

8.1419.2.1 uint8_t wds_ProfileIdentifier::profileIndex

8.1419.2.2 uint8_t wds_ProfileIdentifier::profileType

8.1420 wds_profileInfo Union Reference

Data Fields

- LibPackprofile_3GPP SlqsProfile3GPP
- LibPackprofile_3GPP2 SlqsProfile3GPP2

8.1420.1 Detailed Description

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

Parameters

SlqsProfile3GPP	3GPP profile
SlqsProfile3GP-P2	3GPP2 profile

8.1420.2 Field Documentation

8.1420.2.1 LibPackprofile_3GPP wds_profileInfo::SlqsProfile3GPP

8.1420.2.2 LibPackprofile_3GPP2 wds_profileInfo::SlqsProfile3GPP2

8.1421 wds_TrStatInd Struct Reference

Data Fields

- uint8_t statsPeriod
- uint32_t statsMask

8.1421.1 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.

8.1421.2 Field Documentation

8.1421.2.1 uint32_t wds_TrStatInd::statsMask

8.1421.2.2 uint8_t wds_TrStatInd::statsPeriod

8.1422 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

8.1422.1 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.

8.1422.2 Field Documentation

8.1422.2.1 uint8_t wds_UMTSMinQoS::deliveryErrSDU

8.1422.2.2 uint32_t wds_UMTSMinQoS::grntDownlinkBitrate

8.1422.2.3 uint32_t wds_UMTSMinQoS::grntUplinkBitrate

8.1422.2.4 uint32_t wds_UMTSMinQoS::maxDownlinkBitrate

8.1422.2.5 uint32_t wds_UMTSMinQoS::maxSDUSize

8.1422.2.6 uint32_t wds_UMTSMinQoS::maxUplinkBitrate

8.1422.2.7 uint8_t wds_UMTSMinQoS::qosDeliveryOrder

8.1422.2.8 uint8_t wds_UMTSMinQoS::resBerRatio

8.1422.2.9 uint8_t wds_UMTSMinQoS::sduErrorRatio

8.1422.2.10 uint8_t wds_UMTSMinQoS::trafficClass

8.1422.2.11 uint32_t wds_UMTSMinQoS::trafficPriority

8.1422.2.12 uint32_t wds_UMTSMinQoS::transferDelay

8.1423 WdsByteTotals Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsByteTotalsElmnts ByteTotalsElmntsV4
- struct WdsByteTotalsElmnts ByteTotalsElmntsV6

8.1423.1 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.

8.1423.2 Field Documentation

8.1423.2.1 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV4

8.1423.2.2 struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV6

8.1423.2.3 ULONG* WdsByteTotals::pV4sessionId

8.1423.2.4 ULONG* WdsByteTotals::pV6sessionId

8.1424 WdsByteTotalsElmnts Struct Reference

Data Fields

- ULONGLONG * pTXTotalBytes
- ULONGLONG * pRXTotalBytes

8.1424.1 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.

8.1424.2 Field Documentation

8.1424.2.1 ULONGLONG* WdsByteTotalsElmnts::pRXTotalBytes

8.1424.2.2 ULONGLONG* WdsByteTotalsElmnts::pTXTotalBytes

8.1425 WdsClientLeaseChange Struct Reference

Data Fields

- BYTE * pEnableNotification

8.1425.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

Parameters

pEnable-Notification	[IN] • Enable Notification or not

8.1425.2 Field Documentation

8.1425.2.1 BYTE* WdsClientLeaseChange::pEnableNotification

8.1426 WdsConnectionRate Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsConnectionRateElmnts ConnRateElmntsV4
- struct WdsConnectionRateElmnts ConnRateElmntsV6

8.1426.1 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.

8.1426.2 Field Documentation

8.1426.2.1 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV4

8.1426.2.2 struct WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV6

8.1426.2.3 ULONG* WdsConnectionRate::pV4sessionId

8.1426.2.4 ULONG* WdsConnectionRate::pV6sessionId

8.1427 WdsConnectionRateElmnts Struct Reference

Data Fields

- ULONG * pCurrentChannelTXRate
- ULONG * pCurrentChannelRXRate
- ULONG * pMaxChannelTXRate
- ULONG * pMaxChannelRXRate

8.1427.1 Detailed Description

WDS Connection rates request data structure for individual session

Parameters

pCurrent-ChannelTX-Rate[OUT]	• Instantaneous channel Tx rate in bits per second.
pCurrent-ChannelRX-Rate[OUT]	• Instantaneous channel Rx rate in bits per second.
pMaxChannelT-XRate[OUT]	• Maximum Tx rate that can be assigned to the device by the • serving system in bits per second
pMaxChannelR-XRate[OUT]	• Maximum Rx rate that can be assigned to the device by the • serving system in bits per second

8.1427.2 Field Documentation

8.1427.2.1 ULONG* WdsConnectionRateElmnts::pCurrentChannelRXRate

8.1427.2.2 ULONG* WdsConnectionRateElmnts::pCurrentChannelTXRate

8.1427.2.3 ULONG* WdsConnectionRateElmnts::pMaxChannelRXRate

8.1427.2.4 ULONG* WdsConnectionRateElmnts::pMaxChannelTXRate

8.1428 WdsDHCPv4ClientLeaseInd Struct Reference

Data Fields

- WdsDHCPv4ProfileId * pProfileId
- BYTE * pLeaseState
- ULONG * pIPv4Addr
- DHCPOptionList * pOptList

8.1428.1 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

8.1428.2 Field Documentation

8.1428.2.1 ULONG* WdsDHCPv4ClientLeaseInd::pIPv4Addr

8.1428.2.2 BYTE* WdsDHCPv4ClientLeaseInd::pLeaseState

8.1428.2.3 DHCPOptionList* WdsDHCPv4ClientLeaseInd::pOptList

8.1428.2.4 WdsDHCPv4ProfileId* WdsDHCPv4ClientLeaseInd::pProfileId

8.1429 WdsDHCPv4Config Struct Reference

Data Fields

- WdsDHCPv4ProfileId * pProfileId
- WdsDHCPv4HWConfig * pHwConfig
- WdsDHCPv4OptionList * pRequestOptionList

8.1429.1 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

8.1429.2 Field Documentation

8.1429.2.1 WdsDHCPv4HWConfig* WdsDHCPv4Config::pHwConfig

8.1429.2.2 WdsDHCPv4ProfileId* WdsDHCPv4Config::pProfileId

8.1429.2.3 WdsDHCPv4OptionList* WdsDHCPv4Config::pRequestOptionList

8.1430 wdsDhcpv4HwConfig Struct Reference

Data Fields

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

8.1430.1 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

8.1430.2 Field Documentation

8.1430.2.1 uint8_t wdsDhcpv4HwConfig::chaddr[16]

8.1430.2.2 uint8_t wdsDhcpv4HwConfig::chaddrLen

8.1430.2.3 uint8_t wdsDhcpv4HwConfig::hwType

8.1431 WdsDHCPv4HWConfig Struct Reference

Data Fields

- BYTE hwType
- BYTE chaddrLen
- BYTE chaddr [16]

8.1431.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

Parameters

hwType	• HW Type 1 - Ethernet 20 - Serial
chaddrlen	• chaddrlen
chaddr	• chaddr. Max size 16 bytes

8.1431.2 Field Documentation

8.1431.2.1 BYTE WdsDHCPv4HWConfig::chaddr[16]

8.1431.2.2 BYTE WdsDHCPv4HWConfig::chaddrLen

8.1431.2.3 BYTE WdsDHCPv4HWConfig::hwType

8.1432 WdsDHCPv4Option Struct Reference

Data Fields

- BYTE optCode
- BYTE optValLen
- BYTE optVal [255]

8.1432.1 Detailed Description

WDS SWI DHCPv4 Option Structure

Parameters

optCode	• Option code – 0 - 255
optValLen	• Option value length – 0 - 255
optVal	• Option value

8.1432.2 Field Documentation

8.1432.2.1 BYTE WdsDHCPv4Option::optCode

8.1432.2.2 BYTE WdsDHCPv4Option::optVal[255]

8.1432.2.3 BYTE WdsDHCPv4Option::optValLen

8.1433 wdsDhcpv4Option Struct Reference

Data Fields

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

8.1433.1 Detailed Description

Parameters

optCode	Option code • 0 - 255
optValLen	Option value length • 0 - 255
optVal	Option Value

8.1433.2 Field Documentation

8.1433.2.1 uint8_t wdsDhcpv4Option::optCode

8.1433.2.2 uint8_t wdsDhcpv4Option::optVal[255]

8.1433.2.3 uint8_t wdsDhcpv4Option::optValLen

8.1434 wdsDhcpv4OptionList Struct Reference

Data Fields

- uint8_t numOpt
- wdsDhcpv4Option * pOptList

8.1434.1 Detailed Description

Parameters

numOpt	number of options • 0 - 255
pOptList	pointer to list of DHCP Options

8.1434.2 Field Documentation

8.1434.2.1 uint8_t wdsDhcpv4OptionList::numOpt

8.1434.2.2 wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList

8.1435 WdsDHCPv4OptionList Struct Reference

Data Fields

- BYTE numOpt
- WdsDHCPv4Option * pOptList

8.1435.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

Parameters

numOpt	• number of options – 0 - 255
pOptList	• pointer to list of DHCP Options

8.1435.2 Field Documentation

8.1435.2.1 BYTE WdsDHCPv4OptionList::numOpt

8.1435.2.2 WdsDHCPv4Option* WdsDHCPv4OptionList::pOptList

8.1436 wdsDhcpv4ProfileId Struct Reference

Data Fields

- uint8_t profileType
- uint8_t profileId

8.1436.1 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)

8.1436.2 Field Documentation

8.1436.2.1 uint8_t wdsDhcpv4ProfileId::profileId

8.1436.2.2 uint8_t wdsDhcpv4ProfileId::profileType

8.1437 WdsDHCPv4ProfileId Struct Reference

Data Fields

- BYTE profileType
- BYTE profileId

8.1437.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

Parameters

profileType	• 0 for 3GPP
profileId	• 1 to 24 for 3GPP profile

8.1437.2 Field Documentation

8.1437.2.1 BYTE WdsDHCPv4ProfileId::profileId

8.1437.2.2 BYTE WdsDHCPv4ProfileId::profileType

8.1438 WDSGetLoopbackData Struct Reference

Data Fields

- BYTE ByteLoopbackMode

- BYTE ByteLoopbackMultiplier

8.1438.1 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.

8.1438.2 Field Documentation

8.1438.2.1 BYTE WDSGetLoopbackData::ByteLoopbackMode

8.1438.2.2 BYTE WDSGetLoopbackData::ByteLoopbackMultiplier

8.1439 WdsIpAddressInfoReq Struct Reference

Data Fields

- ULONG * pv4sessionId
- ULONG * pv6sessionId
- QmiWdsIpAddressInfo ip

8.1439.1 Detailed Description

Parameters

pv4sessionId[OUT]	• IPv4 session id
pv6sessionId[OUT]	• IPv6 session id
ip[OUT]	• IP address information see QmiWdsIpAddressInfo for more details

8.1439.2 Field Documentation

8.1439.2.1 QmiWdsIpAddressInfo WdsIpAddressInfoReq::ip

8.1439.2.2 ULONG* WdsIpAddressInfoReq::pv4sessionId

8.1439.2.3 ULONG* WdsIpAddressInfoReq::pv6sessionId

8.1440 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

8.1440.1 Detailed Description

WDS Pkt Statistics request data structure for individual session

Parameters

pTXPacketSuccesses	• Number of transmitted Packets without error.
pRXPacketSuccesses	• Number of received Packets without error.
pTXPacketErrors	• Number of outgoing packets with framing errors.
pRXPacketErrors	• Number of incoming packets with framing errors.

pTXPacket-Overflows	• Number of packets dropped because Tx buffer overflowed (out of memory).
pRXPacket-Overflows	• Number of packets dropped because Rx buffer overflowed (out of memory).
pTXOkBytes-Count	• Number of bytes transmitted without error.
pRXOkBytes-Count	• Number of bytes received without error.
pTXOKBytes-LastCall	• Number of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
pRXOKBytes-LastCall	• Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
pTXDropped-Count	• Number of outgoing packets dropped.
pRXDropped-Count	• Number of incoming packets dropped.

8.1440.2 Field Documentation

8.1440.2.1 ULONG* WdsPktStatisticsElmnts::pRXDroppedCount

8.1440.2.2 ULONGLONG* WdsPktStatisticsElmnts::pRXOkBytesCount

8.1440.2.3 ULONGLONG* WdsPktStatisticsElmnts::pRXOKBytesLastCall

8.1440.2.4 ULONG* WdsPktStatisticsElmnts::pRXPacketErrors

8.1440.2.5 ULONG* WdsPktStatisticsElmnts::pRXPacketOverflows

8.1440.2.6 ULONG* WdsPktStatisticsElmnts::pRXPacketSuccesses

8.1440.2.7 ULONG* WdsPktStatisticsElmnts::pTXDroppedCount

8.1440.2.8 ULONGLONG* WdsPktStatisticsElmnts::pTXOkBytesCount

8.1440.2.9 ULONGLONG* WdsPktStatisticsElmnts::pTXOKBytesLastCall

8.1440.2.10 ULONG* WdsPktStatisticsElmnts::pTXPacketErrors

8.1440.2.11 ULONG* WdsPktStatisticsElmnts::pTXPacketOverflows

8.1440.2.12 ULONG* WdsPktStatisticsElmnts::pTXPacketSuccesses

8.1441 WdsPktStatisticsReq Struct Reference

Data Fields

- ULONG * pStatMask

8.1441.1 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

8.1441.2 Field Documentation

8.1441.2.1 ULONG* WdsPktStatisticsReq::pStatMask

8.1442 WdsPktStatisticsResp Struct Reference

Data Fields

- ULONG * pV4sessionId
- ULONG * pV6sessionId
- struct WdsPktStatisticsElmnts PktStatElmntsV4
- struct WdsPktStatisticsElmnts PktStatElmntsV6

8.1442.1 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.

8.1442.2 Field Documentation

8.1442.2.1 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV4

8.1442.2.2 struct WdsPktStatisticsElmnts WdsPktStatisticsResp::PktStatElmntsV6

8.1442.2.3 ULONG* WdsPktStatisticsResp::pV4sessionId

8.1442.2.4 ULONG* WdsPktStatisticsResp::pV6sessionId

8.1443 WdsProfileParam Union Reference

Data Fields

- struct Profile3GPP SlqsProfile3GPP
- struct Profile3GPP2 SlqsProfile3GPP2

8.1443.1 Detailed Description

This union WdsProfileParam consist of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

Parameters

SlqsProfile3GPP	• 3GPP profile
SlqsProfile3GP-P2	• 3GPP2 profile

8.1443.2 Field Documentation

8.1443.2.1 struct Profile3GPP WdsProfileParam::SlqsProfile3GPP

8.1443.2.2 struct Profile3GPP2 WdsProfileParam::SlqsProfile3GPP2

8.1444 WdsRunTimeSettings Struct Reference

Data Fields

- ULONG * v4sessionId
- ULONG * v6sessionId
- struct qmiWdsRunTimeSettings rts

8.1444.1 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.

8.1444.2 Field Documentation

8.1444.2.1 struct qmiWdsRunTimeSettings WdsRunTimeSettings::rts

8.1444.2.2 ULONG* WdsRunTimeSettings::v4sessionId

8.1444.2.3 ULONG* WdsRunTimeSettings::v6sessionId

8.1445 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

8.1445.1 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.

8.1445.2 Field Documentation

8.1445.2.1 BYTE* wdsSetEventReportReq::pCurrChannelRateInd

8.1445.2.2 BYTE* wdsSetEventReportReq::pCurrDataBearerTechInd

8.1445.2.3 BYTE* wdsSetEventReportReq::pCurrPrefDataSysInd

8.1445.2.4 BYTE* wdsSetEventReportReq::pDataBearerTechInd

8.1445.2.5 BYTE* wdsSetEventReportReq::pDataCallStatusChangeInd

8.1445.2.6 BYTE* wdsSetEventReportReq::pDataSystemStatusChangeInd

8.1445.2.7 BYTE* wdsSetEventReportReq::pDormancyStatusInd

8.1445.2.8 BYTE* wdsSetEventReportReq::pEVDOPageMonPerChangeInd

8.1445.2.9 BYTE* wdsSetEventReportReq::pMIPStatusInd

8.1445.2.10 TrStatInd* wdsSetEventReportReq::pTransferStatInd

8.1446 WDSSetLoopbackData Struct Reference

Data Fields

- BYTE * pLoopbackMode
- BYTE * pLoopbackMultiplier

8.1446.1 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.

8.1446.2 Field Documentation

8.1446.2.1 BYTE* WDSSetLoopbackData::pLoopbackMode

8.1446.2.2 BYTE* WDSSetLoopbackData::pLoopbackMultiplier

8.1447 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

8.1447.1 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.

8.1447.2 Field Documentation

8.1447.2.1 unsigned long WDSSWICurrentChannelRates::current_channel_rx_rate

8.1447.2.2 unsigned long WDSSWICurrentChannelRates::current_channel_tx_rate

8.1447.2.3 unsigned long WDSSWICurrentChannelRates::max_channel_rx_rate

8.1447.2.4 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,
  eIMSA =0x21,
  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 ()
- void liteqmi_helper_decode7bitAsciiEncString (uint8_t *encoded_str, uint8_t encoded_len, uint8_t *decoded_str)

Variables

- logger glog
- uint8_t gloglvl

9.2.1 Macro Definition Documentation

9.2.1.1 #define DEAULT_LOC_TIMEOUT_IN_SEC 2

9.2.1.2 #define MINREQBKLEN (2048)

9.2.1.3 #define MSGID_AND_LEN (4)

9.2.1.4 #define MSGID_DONT_CARE (0xffff)

9.2.1.5 #define SDK_VALIDATE_INPUT_PACK_PARAM(*pCtx, pBuf, pLen*)

Value:

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
        {\
        liteqmi_log(eLOG_DEBUG,"[  pack] %s parameter NULL\n",__func__); \
        return eQCWWAN_ERR_INVALID_ARG; \
        }
```

9.2.1.6 #define SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID(*pCtx, pBuf, pLen*)

Value:

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
        {\
        liteqmi_log(eLOG_DEBUG,"[  pack] %s parameter NULL\n",__func__); \
        return eQCWWAN_ERR_INVALID_ARG; \
        } \
    if (pCtx->xid == 0) \
       return eQCWWAN_ERR_INVALID_XID; \
    pBuf[0] = eREQ; \
    pBuf[1] = pCtx->xid & 0xff; \
    pBuf[2] = pCtx->xid >> 8;
```

9.2.1.7 #define 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

eIMSA

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_helper_decode7bitAsciiEncString (uint8_t * *encoded_str,* uint8_t *encoded_len,* uint8_t * *decoded_str*)

decode 7bit encoded string

Parameters

in	*encoded_str*	7 bit encoded ASCII
in	*encoded_len*	encoded length
out	*decoded_str*	decoded ASCII string

9.2.4.10 void liteqmi_log (uint8_t *lvl,* const char * *fmt,* ...)

9.2.4.11 unsigned unpack_result_code_only (uint8_t * *pMdmResp*)

common handler for unpacking response with TLV type 0x02 only

9.2.5 Variable Documentation

9.2.5.1 logger glog

9.2.5.2 uint8_t gloglvl

9.3 dms.h File Reference

Data Structures

- struct unpack_dms_GetModelID_t
- struct unpack_dms_GetIMSI_t
- struct unpack_dms_GetFirmwareInfo_t
- struct unpack_dms_GetPower_t
- struct unpack_dms_GetSerialNumbers_t
- struct unpack_dms_GetHardwareRevision_t
- struct unpack_dms_SLQSGetBandCapability_t
- struct unpack_dms_GetDeviceCapabilities_t
- struct unpack_dms_GetFirmwareRevisions_t
- struct unpack_dms_GetFirmwareRevision_t
- struct unpack_dms_GetDeviceSerialNumbers_t
- struct unpack_dms_GetPRLVersion_t
- struct unpack_dms_GetNetworkTime_t
- struct unpack_dms_GetVoiceNumber_t
- struct unpack_dms_GetDeviceHardwareRev_t
- struct unpack_dms_GetFSN_t
- struct unpack_dms_GetDeviceCap_t
- struct pack_dms_SetPower_t
- struct unpack_dms_SetPower_t
- struct unpack_dms_GetBandCapability_t
- struct unpack_dms_GetUSBComp_t
- struct pack_dms_SetUSBComp_t
- struct unpack_dms_SetUSBComp_t
- struct pack_dms_SetCustFeature_t
- struct unpack_dms_SetCustFeature_t
- struct unpack_dms_GetCustFeature_t
- struct unpack_dms_SetFirmwarePreference_t

- struct unpack_dms_GetCrashAction_t
- struct pack_dms_SetCrashAction_t
- struct unpack_dms_SetCrashAction_t
- struct unpack_dms_GetDeviceMfr_t
- struct pack_dms_SetEventReport_t
- struct unpack_dms_SetEventReport_t
- struct dms_OperatingModeTlv
- struct dms_ActivationStatusTlv
- struct unpack_dms_SetEventReport_ind_t
- struct pack_dms_UIMGetICCID_t
- struct unpack_dms_UIMGetICCID_t
- struct pack_dms_SetCustFeaturesV2_t
- struct unpack_dms_SetCustFeaturesV2_t
- struct pack_dms_GetCustFeaturesV2_t
- struct DMSgetCustomInput
- struct DMScustSettingInfo
- struct DMScustSettingList
- struct DMSgetCustomFeatureV2
- struct unpack_dms_GetCustFeaturesV2_t
- struct unpack_dms_GetActivationState_t
- struct image_info_t
- struct unpack_dms_SLQSSwiGetFirmwareCurr_t
- struct pack_dms_SLQSSwiSetDyingGaspCfg_t
- struct unpack_dms_SLQSSwiSetDyingGaspCfg_t
- struct unpack_dms_SLQSSwiClearDyingGaspStatistics_t
- struct packgetDyingGaspStatistics
- struct unpack_dms_SLQSSwiGetDyingGaspStatistics_t
- struct packgetDyingGaspCfg
- struct unpack_dms_SLQSSwiGetDyingGaspCfg_t
- struct unpack_dms_SLQSDmsSwiGetResetInfo_t
- struct unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t
- struct pack_dms_SLQSDmsSwiIndicationRegister_t
- struct unpack_dms_SLQSDmsSwiIndicationRegister_t
- struct unpack_dms_SLQSSwiGetFwUpdateStatus_t
- struct unpack_dms_GetManufacturer_t
- struct unpack_dms_GetOfflineReason_t
- struct pack_dms_SetActivationStatusCallback_t
- struct unpack_dms_SetActivationStatusCallback_t
- struct pack_dms_UIMSetPINProtection_t
- struct unpack_dms_UIMSetPINProtection_t
- struct pack_dms_UIMUnblockPIN_t
- struct pack_dms_UIMVerifyPIN_t
- struct pack_dms_UIMChangePIN_t
- struct pack_dms_UIMGetControlKeyStatus_t
- struct unpack_dms_UIMGetControlKeyStatus_t
- struct unpack_dms_UIMGetPINStatus_t
- struct pack_dms_UIMSetControlKeyProtection_t
- struct unpack_dms_UIMSetControlKeyProtection_t
- struct pack_dms_UIMUnblockControlKey_t
- struct unpack_dms_UIMUnblockControlKey_t
- struct pack_dms_ResetToFactoryDefaults_t
- struct unpack_dms_ResetToFactoryDefaults_t
- struct pack_dms_ActivateAutomatic_t
- struct eriDataparams
- struct unpack_dms_SLQSGetERIFile_t

- struct unpack_dms_SLQSUIMGetState_t
- struct pack_dms_SLQSSwiGetCrashInfo_t
- struct crashInformation
- struct crashInfoParams
- struct unpack_dms_SLQSSwiGetCrashInfo_t
- struct unpack_dms_SLQSSwiGetHostDevInfo_t
- struct pack_dms_SLQSSwiSetHostDevInfo_t
- struct unpack_dms_SLQSSwiSetHostDevInfo_t
- struct unpack_dms_SLQSSwiGetOSInfo_t
- struct pack_dms_SLQSSwiSetOSInfo_t
- struct unpack_dms_SLQSSwiSetOSInfo_t
- struct unpack_dms_SLQSSwiGetSerialNoExt_t

Macros

- #define DMS_UINT8_MAX_STRING_SZ 255
- #define DMS_MAX_CUST_ID_LEN 64
- #define DMS_MAX_CUST_VALUE_LEN 8
- #define DMS_IMGDETAILS_LEN 16
- #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255
- #define DMS_MAX_FWUPDATE_REF_STR_SZ 15
- #define DMS_SLQSFWINFO_MODELID_SZ 20
- #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85
- #define DMS_SLQSFWINFO_APPVERSION_SZ 85
- #define DMS_SLQSFWINFO_SKU_SZ 15
- #define DMS_SLQSFWINFO_PACKAGEID_SZ 85
- #define DMS_SLQSFWINFO_CARRIER_SZ 20
- #define DMS_SLQSFWINFO_PRIVERSION_SZ 16
- #define DMS_SLQSFWINFO_CUR_CARR_NAME 17
- #define DMS_SLQSFWINFO_CUR_CARR_REV 13
- #define MAX_BUILD_ID_LEN 255
- #define UNIQUE_ID_LEN 16
- #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160
- #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20
- #define SPC_SIZE 6
- #define CK_MAX_SIZE 8
- #define ACT_CODE_MAX_SIZE 81
- #define ERI_DATA_MAX_SIZE 1024
- #define MEID_MAX_SIZE 8
- #define DMS_PM_ONLINE 0x00 /∗ Online ∗/
- #define DMS_PM_LOW 0x01 /∗ Low Power ∗/
- #define DMS_PM_FACTORY 0x02 /∗ Factory Test Mode ∗/
- #define DMS_PM_OFFLINE 0x03 /∗ Offline ∗/
- #define DMS_PM_RESET 0x04 /∗ Reset ∗/
- #define DMS_PM_SHUT_DOWN 0x05 /∗ Shut Down ∗/
- #define DMS_PM_PERSISTENT_LOW 0x06 /∗ Persistent Low Power ∗/
- #define DMS_SET_REPORT_ENABLE 1
- #define DMS_SET_REPORT_DISABLE 0
- #define DMS_SWI_SET_IND_ENABLE 1
- #define DMS_SWI_SET_IND_DISABLE 0

Functions

- int pack_dms_GetIMSI (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetIMSI (uint8_t *pResp, uint16_t respLen, unpack_dms_GetIMSI_t *pOutput)
- int pack_dms_GetModelID (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetModelID (uint8_t *pResp, uint16_t respLen, unpack_dms_GetModelID_t *pOutput)
- int pack_dms_GetFirmwareInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetFirmwareInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareInfo_t *pOutput)
- int pack_dms_GetPower (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetPower (uint8_t *pResp, uint16_t respLen, unpack_dms_GetPower_t *pOutput)
- int pack_dms_GetSerialNumbers (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetSerialNumbers (uint8_t *pResp, uint16_t respLen, unpack_dms_GetSerialNumbers_t *pOutput)
- int pack_dms_GetHardwareRevision (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetHardwareRevision (uint8_t *pResp, uint16_t respLen, unpack_dms_GetHardwareRevision_t *pOutput)
- int pack_dms_SLQSGetBandCapability (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_SLQSGetBandCapability (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSGetBandCapability_t *pOutput)
- int pack_dms_GetDeviceCapabilities (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetDeviceCapabilities (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceCapabilities_t *pOutput)
- int pack_dms_GetFirmwareRevisions (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetFirmwareRevisions (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareRevisions_t *pOutput)
- int pack_dms_GetFirmwareRevision (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetFirmwareRevision (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareRevision_t *pOutput)
- int pack_dms_GetDeviceSerialNumbers (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetDeviceSerialNumbers (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceSerialNumbers_t *pOutput)
- int pack_dms_GetPRLVersion (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetPRLVersion (uint8_t *pResp, uint16_t respLen, unpack_dms_GetPRLVersion_t *pOutput)
- int pack_dms_GetNetworkTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetNetworkTime (uint8_t *pResp, uint16_t respLen, unpack_dms_GetNetworkTime_t *pOutput)
- int pack_dms_GetVoiceNumber (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetVoiceNumber (uint8_t *pResp, uint16_t respLen, unpack_dms_GetVoiceNumber_t *pOutput)
- int pack_dms_GetDeviceHardwareRev (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetDeviceHardwareRev (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceHardwareRev_t *pOutput)
- int pack_dms_GetFSN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetFSN (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFSN_t *pOutput)
- int pack_dms_GetDeviceCap (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetDeviceCap (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceCap_t *pOutput)
- int pack_dms_SetPower (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetPower_t *reqArg)
- int unpack_dms_SetPower (uint8_t *pResp, uint16_t respLen, unpack_dms_SetPower_t *pOutput)
- int pack_dms_GetBandCapability (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetBandCapability (uint8_t *pResp, uint16_t respLen, unpack_dms_GetBandCapability_t *pOutput)
- int pack_dms_GetUSBComp (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetUSBComp (uint8_t *pResp, uint16_t respLen, unpack_dms_GetUSBComp_t *pOutput)

- int pack_dms_SetUSBComp (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetUSBComp_t *reqArg)
- int unpack_dms_SetUSBComp (uint8_t *pResp, uint16_t respLen, unpack_dms_SetUSBComp_t *pOutput)
- int pack_dms_SetCustFeature (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetCustFeature_t *reqArg)
- int unpack_dms_SetCustFeature (uint8_t *pResp, uint16_t respLen, unpack_dms_SetCustFeature_t *pOutput)
- int pack_dms_GetCustFeature (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetCustFeature (uint8_t *pResp, uint16_t respLen, unpack_dms_GetCustFeature_t *pOutput)
- int pack_dms_SetFirmwarePreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_SetFirmwarePreference (uint8_t *pResp, uint16_t respLen, unpack_dms_SetFirmwarePreference_t *pOutput)
- int pack_dms_GetCrashAction (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetCrashAction (uint8_t *pResp, uint16_t respLen, unpack_dms_GetCrashAction_t *pOutput)
- int pack_dms_SetCrashAction (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetCrashAction_t reqArg)
- int unpack_dms_SetCrashAction (uint8_t *pResp, uint16_t respLen, unpack_dms_SetCrashAction_t *pOutput)
- int pack_dms_GetDeviceMfr (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_GetDeviceMfr (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceMfr_t *pOutput)
- int pack_dms_SetEventReport (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetEventReport_t *reqArg)
- int unpack_dms_SetEventReport (uint8_t *pResp, uint16_t respLen, unpack_dms_SetEventReport_t *pOutput)
- int unpack_dms_SetEventReport_ind (uint8_t *pResp, uint16_t respLen, unpack_dms_SetEventReport_ind_t *pOutput)
- int pack_dms_UIMGetICCID (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMGetICCID_t *reqArg)
- int unpack_dms_UIMGetICCID (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMGetICCID_t *pOutput)
- int pack_dms_SetCustFeaturesV2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetCustFeaturesV2_t *reqArg)
- int unpack_dms_SetCustFeaturesV2 (uint8_t *pResp, uint16_t respLen, unpack_dms_SetCustFeaturesV2_t *pOutput)
- int pack_dms_GetCustFeaturesV2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_GetCustFeaturesV2_t *reqArg)
- int unpack_dms_GetCustFeaturesV2 (uint8_t *pResp, uint16_t respLen, unpack_dms_GetCustFeaturesV2_t *pOutput)
- int pack_dms_GetActivationState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_GetActivationState (uint8_t *pResp, uint16_t respLen, unpack_dms_GetActivationState_t *pOutput)
- int pack_dms_SLQSSwiGetFirmwareCurr (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSSwiGetFirmwareCurr (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetFirmwareCurr_t *pOutput)
- int pack_dms_SLQSSwiSetDyingGaspCfg (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiSetDyingGaspCfg_t *reqArg)
- int unpack_dms_SLQSSwiSetDyingGaspCfg (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiSetDyingGaspCfg_t *pOutput)
- int pack_dms_SLQSSwiClearDyingGaspStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSSwiClearDyingGaspStatistics (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiClearDyingGaspStatistics_t *pOutput)
- int pack_dms_SLQSSwiGetDyingGaspStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSSwiGetDyingGaspStatistics (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspStatistics_t *pOutput)
- int pack_dms_SLQSSwiGetDyingGaspCfg (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)

- int unpack_dms_SLQSSwiGetDyingGaspCfg (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspCfg_t *pOutput)
- int pack_dms_SLQSDmsSwiGetResetInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSDmsSwiGetResetInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetResetInfo_t *pOutput)
- int unpack_dms_SLQSDmsSwiGetResetInfo_Ind (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t *pOutput)
- int pack_dms_SLQSDmsSwiIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSDmsSwiIndicationRegister_t *reqArg)
- int unpack_dms_SLQSDmsSwiIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiIndicationRegister_t *pOutput)
- int pack_dms_SLQSSwiGetFwUpdateStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSSwiGetFwUpdateStatus (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetFwUpdateStatus_t *pOutput)
- int pack_dms_GetManufacturer (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_GetManufacturer (uint8_t *pResp, uint16_t respLen, unpack_dms_GetManufacturer_t *pOutput)
- int pack_dms_GetOfflineReason (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_GetOfflineReason (uint8_t *pResp, uint16_t respLen, unpack_dms_GetOfflineReason_t *pOutput)
- int pack_dms_SetActivationStatusCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetActivationStatusCallback_t *reqArg)
- int unpack_dms_SetActivationStatusCallback (uint8_t *pResp, uint16_t respLen, unpack_dms_SetActivationStatusCallback_t *pOutput)
- int pack_dms_UIMSetPINProtection (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMSetPINProtection_t *pReq)
- int unpack_dms_UIMSetPINProtection (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)
- int pack_dms_UIMUnblockPIN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMUnblockPIN_t *pReq)
- int unpack_dms_UIMUnblockPIN (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)
- int pack_dms_UIMVerifyPIN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMVerifyPIN_t *pReq)
- int unpack_dms_UIMVerifyPIN (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)
- int pack_dms_UIMChangePIN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMChangePIN_t *pReq)
- int unpack_dms_UIMChangePIN (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)
- int pack_dms_UIMGetControlKeyStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMGetControlKeyStatus_t *pReq)
- int unpack_dms_UIMGetControlKeyStatus (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMGetControlKeyStatus_t *pOutput)
- int pack_dms_UIMGetPINStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_UIMGetPINStatus (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMGetPINStatus_t *pOutput)
- int pack_dms_UIMSetControlKeyProtection (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMSetControlKeyProtection_t *pReq)
- int unpack_dms_UIMSetControlKeyProtection (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetControlKeyProtection_t *pOutput)
- int pack_dms_UIMUnblockControlKey (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMUnblockControlKey_t *pReq)
- int unpack_dms_UIMUnblockControlKey (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMUnblockControlKey_t *pOutput)

- int pack_dms_ResetToFactoryDefaults (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_ResetToFactoryDefaults_t *pReq)
- int unpack_dms_ResetToFactoryDefaults (uint8_t *pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t *pOutput)
- int pack_dms_ValidateSPC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_ResetToFactoryDefaults_t *pReq)
- int unpack_dms_ValidateSPC (uint8_t *pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t *pOutput)
- int pack_dms_ActivateAutomatic (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_ActivateAutomatic_t *pReq)
- int unpack_dms_ActivateAutomatic (uint8_t *pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t *pOutput)
- int pack_dms_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*)

Requests the device to perform automatic service activation pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
out	*pReq*	requeset parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.2 int pack_dms_GetActivationState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get Activation State pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.3 int pack_dms_GetBandCapability (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Band Capability pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.4 int pack_dms_GetCrashAction (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Crash Action pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.5 int pack_dms_GetCustFeature (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use pack_dms_GetCustFeatures-V2() instead for EM74xx/MC74xx.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.6 int pack_dms_GetCustFeaturesV2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_GetCustFeaturesV2_t * *reqArg*)

Get Custom Feature pack This function queries the modem for a list of supported features. This function is for firmware version 2.0 and newer. Currently supported Customization features:

- GPIOSARENABLE
- GPSSEL
- IMSWITCHHIDE

- IPV6ENABLE
- WAKEHOSTEN

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.7 int pack_dms_GetDeviceCap (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Device Capabilities pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.8 int pack_dms_GetDeviceCapabilities (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get device capability pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.9 int pack_dms_GetDeviceHardwareRev (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Hardware Revision of the device pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.10 int pack_dms_GetDeviceMfr (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Device Manufacture Name pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.11 int pack_dms_GetDeviceSerialNumbers (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Device Serial Number pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.12 int pack_dms_GetFirmwareInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get firmware info pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.13 int pack_dms_GetFirmwareRevision (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Firmware Revision pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.14 int pack_dms_GetFirmwareRevisions (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Firmware Revisions pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.15 int pack_dms_GetFSN (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get FSN pack. This API get the Factory Sequence Number of the device.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.16 int pack_dms_GetHardwareRevision (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get hardware revision of the device pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.17 int pack_dms_GetIMSI (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get IMSI pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_-xx_xx and all EM74xx firmware versions. Please use pack_uim_ReadTransparent()(EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.18 int pack_dms_GetManufacturer (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To get device manufacturer information pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.19 int pack_dms_GetModelID (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get model id pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.20 int pack_dms_GetNetworkTime (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Network Time pack Returns the current time of the device based on the value supported by the network.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

To get operating mode offline reason pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.22 int pack_dms_GetPower (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get operating mode of the device pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.23 int pack_dms_GetPRLVersion (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get PRL Versions pack. Return version of the active Preferred Roaming List (PRL) in use by the device.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.24 int pack_dms_GetSerialNumbers (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get serial numbers pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.25 int pack_dms_GetUSBComp (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get USB Comp pack This API queries the modem's USB interface configuration and supported configuration parameters.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.26 int pack_dms_GetVoiceNumber (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Voice Number pack. Returns the voice number in use by the device.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.27 int pack_dms_ResetToFactoryDefaults (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_ResetToFactoryDefaults_t * *pReq*)

Resets to default factory settings of the device pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
out	*pReq*	requeset parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.28 int pack_dms_SetActivationStatusCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetActivationStatusCallback_t * *reqArg*)

Set activation status pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.29 int pack_dms_SetCrashAction (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetCrashAction_t *reqArg*)

Set Crash Action pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.30 int pack_dms_SetCustFeature (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetCustFeature_t * *reqArg*)

Set Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use pack_dms_SetCustFeatures-V2() instead for EM74xx/MC74xx.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.31 int pack_dms_SetCustFeaturesV2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetCustFeaturesV2_t * *reqArg*)

Set Cust Features pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.32 int pack_dms_SetEventReport (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetEventReport_t * *reqArg*)

Set Event Report pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.33 int pack_dms_SetFirmwarePreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Set Firmware Preference pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.34 int pack_dms_SetPower (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetPower_t * *reqArg*)

Set Power pack Sets the operating mode of the device.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.35 int pack_dms_SetUSBComp (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetUSBComp_t * *reqArg*)

Set USB Comp pack This API is used to change the modem's USB interface configuration thus allowing a device to have multiple USB compositions. Devices will, by default, be configured to support a minimal set of interfaces to reduce end user modem installation time. Developers and some customers, however, require access to a custom set of interfaces. A reset is required for any change in the USB composition to take effect.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.36 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
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.38 int pack_dms_SLQSGetBandCapability (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get band capability of the device pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter(NULL)

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.39 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

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 Get the Crash Information from the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
out	*pReq*	requeset parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.42 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 get host dev information pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.47 int pack_dms_SLQSSwiGetOSInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To Get Host OS Info pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.48 int pack_dms_SLQSSwiGetSerialNoExt (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To get serial number extension pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.49 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
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

9.3.2.50 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*)

Returns the UIM state pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API unpack_uim_GetCardStatus/ unpack_uim_GetCardStatus for new firmware versions and new modules

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.53 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,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

9.3.2.56 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*)

Requests the device to perform automatic service activation pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
out	*pReq*	requeset parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.63 int unpack_dms_ActivateAutomatic (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_ResetToFactoryDefaults_t * *pOutput*)

Requests the device to perform automatic service activation unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.64 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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*)

Get Custom Feature unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.69 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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 of the device unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.72 int unpack_dms_GetDeviceMfr (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceMfr_t * *pOutput*)

Get Device Manufacture Name unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.73 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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 This API get the Factory Sequence Number of the device.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.78 int unpack_dms_GetHardwareRevision (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetHardware-Revision_t * *pOutput*)

Get hardware revision of the device unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.79 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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. Returns the current time of the device based on the value supported by the network.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.83 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	*respLen*	qmi response length
out	*pOutput*	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 operating mode of the device unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.85 int unpack_dms_GetPRLVersion (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetPRLVersion_t * *pOutput*)

Get PRL Versions unpack Return version of the active Preferred Roaming List (PRL) in use by the device.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.86 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	*respLen*	qmi response length
out	*pOutput*	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 This API queries the modem's USB interface configuration and supported configuration parameters.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.88 int unpack_dms_GetVoiceNumber (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetVoiceNumber_t * *pOutput*)

Get Voice Number unpack Returns the voice number in use by the device.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.89 int unpack_dms_ResetToFactoryDefaults (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_ResetToFactoryDefaults_t * *pOutput*)

Resets to default factory settings of the device unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.90 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	unpacked response. Not used

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.92 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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. Sets the operating mode of the device.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.98 int unpack_dms_SetUSBComp (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetUSBComp_t * *pOutput*)

Set USB Comp unpack This API is used to change the modem's USB interface configuration thus allowing a device to have multiple USB compositions. Devices will, by default, be configured to support a minimal set of interfaces to reduce end user modem installation time. Developers and some customers, however, require access to a custom set of interfaces. A reset is required for any change in the USB composition to take effect.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.99 int unpack_dms_SLQSDmsSwiGetResetInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSDmsSwiGetResetInfo_t * *pOutput*)

To get reset info unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.100 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.101 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.102 int unpack_dms_SLQSGetBandCapability (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSGetBandCapability_t * *pOutput*)

Get band capability of the device unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.103 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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 the Crash Information from the device unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.106 int unpack_dms_SLQSSwiGetDyingGaspCfg (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetDyingGaspCfg_t * *pOutput*)

Get Dying GASP Config unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.107 int unpack_dms_SLQSSwiGetDyingGaspStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetDyingGaspStatistics_t * *pOutput*)

Get Dying GASP Statistics unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.108 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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 information unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.111 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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 set Host Dev Info unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.115 int unpack_dms_SLQSSwiSetOSInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiSetOSInfo_t * *pOutput*)

To set Host OS Info unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.116 int unpack_dms_SLQSUIMGetState (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSUIMGetState_t * *pOutput*)

Returns the UIM state unpack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API unpack_uim_GetCardStatus / unpack_uim_GetCardStatus for new firmware versions and new modules

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.117 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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.3.2.123 int unpack_dms_UIMUnblockControlKey (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMUnblockControlKey_t * *pOutput*)

To get UIM Unblock Control Key unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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	*respLen*	qmi response length
out	*pOutput*	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*)

Requests the device to perform automatic service activation unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.4 fms.h File Reference

Data Structures

- struct CarrierImage_t
- struct pack_fms_GetImagesPreference_t
- struct FMSImageElement
- struct FMSPrefImageList
- struct unpack_fms_GetImagesPreference_t
- struct pack_fms_GetStoredImages_t
- struct FMSImageIdElement
- struct FMSImageIDEntries
- struct FMSImageList
- struct unpack_fms_GetStoredImages_t
- struct pack_fms_SetImagesPreference_t
- struct unpack_fms_SetImagesPreference_t

Macros

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

Functions

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

9.4.1 Macro Definition Documentation

9.4.1.1 #define FMS_FW_PRI_BUILD_MATCH_LEN 11

9.4.1.2 #define FMS_GOBI_LISTENTRIES_MAX 2

9.4.1.3 #define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100

9.4.1.4 #define FMS_GOBI_MBN_IMG_ID_STR_LEN 16

9.4.1.5 #define FMS_IMAGE_ID_BUILD_ID_LEN 32

9.4.1.6 #define FMS_IMAGE_ID_IMG_ID_LEN 16

9.4.1.7 #define FMS_IMAGE_ID_MAX_ENTRIES 2

9.4.1.8 #define FMS_IMAGE_ID_PRI_IMGTYPE 0x01

9.4.1.9 #define FMS_MAX_IMAGE_ID_ELEMENT 50

9.4.1.10 #define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255

9.4.2 Function Documentation

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

This API distills valid Firmware/PRI combinations from GetStoredImages result

Parameters

in	*pStoredImage-List*	• image list returned from GetStoredImages • 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,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

9.4.2.3 int pack_fms_GetStoredImages (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_GetStoredImages_t * *reqArg*)

Get Images Preference pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

9.4.2.4 int pack_fms_SetImagesPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_SetImagesPreference_t * *reqArg*)

Set Images Preference pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

9.4.2.5 int unpack_fms_GetImagesPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_GetImagesPreference_t * *pOutput*)

Get Images Preference unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

9.4.2.6 int unpack_fms_GetStoredImages (uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_GetStoredImages_t * *pOutput*)

Get Images Preference unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

9.4.2.7 int unpack_fms_SetImagesPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_SetImagesPreference_t * *pOutput*)

Set Images Preference unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

9.5 loc.h File Reference

Data Structures

- struct loc_LocApplicationInfo
- struct loc_SV
- struct loc_SVInfo
- struct loc_GnssData
- struct loc_CellDb
- struct loc_ClkInfo
- struct loc_BdsSV
- struct loc_BdsSVInfo
- struct pack_loc_EventRegister_t
- struct unpack_loc_EventRegister_t
- struct pack_loc_SetExtPowerState_t
- struct unpack_loc_SetExtPowerState_t
- struct pack_loc_Start_t
- struct unpack_loc_Start_t
- struct pack_loc_Stop_t
- struct unpack_loc_Stop_t

- struct pack_loc_SetOperationMode_t
- struct unpack_loc_SetOperationMode_t
- struct pack_loc_Delete_Assist_Data_t
- struct unpack_loc_Delete_Assist_Data_t
- struct loc_precisionDilution
- struct loc_sensorDataUsage
- struct loc_svUsedforFix
- struct loc_gpsTime
- struct unpack_loc_PositionRpt_Ind_t
- struct unpack_loc_EngineState_Ind_t
- struct unpack_loc_SetExtPowerConfig_Ind_t
- struct unpack_loc_SLQSLOCGetBestAvailPos_t
- struct pack_loc_SLQSLOCGetBestAvailPos_t
- struct unpack_loc_BestAvailPos_Ind_t
- struct unpack_loc_SetOperationMode_Ind_t
- struct unpack_loc_DeleteAssistData_Ind_t
- struct loc_satelliteInfo
- struct unpack_loc_GnssSvInfo_Ind_t
- struct pack_loc_SLQSLOCInjectUTCTime_t
- struct altSrcInfo_t
- struct pack_loc_SLQSLOCInjectPosition_t
- struct pack_loc_SLQSLOCSetCradleMountConfig_t
- struct sensorData_t
- struct tempData_t
- struct pack_loc_SLQSLOCInjectSensorData_t
- struct unpack_loc_EventNMEA_Ind_t
- struct pack_loc_SLQSLOCGetServer_t
- struct loc_IPv4Info
- struct loc_IPv6Info
- struct loc_urlAddr
- struct unpack_loc_GetServer_Ind_t

Macros

- #define LOC_UINT8_MAX_STRING_SZ 255
- #define LOCEVENTMASKPOSITIONREPORT 0x00000001
- #define LOCEVENTMASKGNSSSVINFO 0x00000002
- #define LOCEVENTMASKNMEA 0x00000004
- #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008
- #define LOCEVENTMASKINJECTTIMEREQ 0x00000010
- #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020
- #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040
- #define LOCEVENTMASKENGINESTATE 0x00000080
- #define LOCEVENTMASKFIXSESSIONSTATE 0x00000100
- #define LOCEVENTMASKWIFIREQ 0x00000200
- #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400
- #define LOCEVENTMASKTIMESYNCREQ 0x00000800
- #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000
- #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000
- #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000
- #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000
- #define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000
- #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000
- #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000

- #define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000
- #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000
- #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000
- #define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000
- #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000
- #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000
- #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF
- #define MAX_SENSOR_DATA_LEN 64
- #define MAX_TEMP_DATA_LEN 64
- #define MAX_LOC_NMEA_STR_LEN 201

Enumerations

- enum {
  eQMI_LOC_SESS_STATUS_SUCCESS =0,
  eQMI_LOC_SESS_STATUS_IN_PROGRESS =1,
  eQMI_LOC_SESS_STATUS_FAILURE =2,
  eQMI_LOC_SESS_STATUS_TIMEOUT =3 }

Functions

- int pack_loc_EventRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_EventRegister_t *reqArg)
- int unpack_loc_EventRegister (uint8_t *pResp, uint16_t respLen, unpack_loc_EventRegister_t *pOutput)
- int pack_loc_SetExtPowerState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SetExtPowerState_t *reqArg)
- int unpack_loc_SetExtPowerState (uint8_t *pResp, uint16_t respLen, unpack_loc_SetExtPowerState_t *pOutput)
- int pack_loc_Start (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Start_t *reqArg)
- int unpack_loc_Start (uint8_t *pResp, uint16_t respLen, unpack_loc_Start_t *pOutput)
- int pack_loc_Stop (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Stop_t *reqArg)
- int unpack_loc_Stop (uint8_t *pResp, uint16_t respLen, unpack_loc_Stop_t *pOutput)
- int pack_loc_SetOperationMode (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SetOperationMode_t *reqArg)
- int unpack_loc_SetOperationMode (uint8_t *pResp, uint16_t respLen, unpack_loc_SetOperationMode_t *pOutput)
- int pack_loc_DeleteAssistData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Delete_Assist_Data_t *reqArg)
- int unpack_loc_DeleteAssistData (uint8_t *pResp, uint16_t respLen, unpack_loc_Delete_Assist_Data_t *pOutput)
- int unpack_loc_PositionRpt_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_PositionRpt_Ind_t *pOutput)
- int unpack_loc_EngineState_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_EngineState_Ind_t *pOutput)
- int unpack_loc_SetExtPowerConfig_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_SetExtPowerConfig_Ind_t *pOutput)
- int pack_loc_SLQSLOCGetBestAvailPos (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCGetBestAvailPos_t *reqArg)
- int unpack_loc_SLQSLOCGetBestAvailPos (uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCGetBestAvailPos_t *pOutput)
- int unpack_loc_BestAvailPos_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_BestAvailPos_Ind_t *pOutput)
- int unpack_loc_SetOperationMode_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_SetOperationMode_Ind_t *pOutput)
- int unpack_loc_DeleteAssistData_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_DeleteAssistData_Ind_t *pOutput)

- int unpack_loc_GnssSvInfo_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_GnssSvInfo_Ind_t *pOutput)
- int pack_loc_SLQSLOCInjectUTCTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectUTCTime_t *reqArg)
- int unpack_loc_SLQSLOCInjectUTCTime (uint8_t *pResp, uint16_t respLen)
- int pack_loc_SLQSLOCInjectPosition (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectPosition_t *reqArg)
- int unpack_loc_SLQSLOCInjectPosition (uint8_t *pResp, uint16_t respLen)
- int pack_loc_SLQSLOCSetCradleMountConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCSetCradleMountConfig_t *reqArg)
- int unpack_loc_SLQSLOCSetCradleMountConfig (uint8_t *pResp, uint16_t respLen)
- int pack_loc_SLQSLOCInjectSensorData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectSensorData_t *reqArg)
- int unpack_loc_SLQSLOCInjectSensorData (uint8_t *pResp, uint16_t respLen)
- int unpack_loc_EventNMEA_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_EventNMEA_Ind_t *pOutput)
- int pack_loc_SLQSLOCGetServer (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCGetServer_t *reqArg)
- int unpack_loc_SLQSLOCGetServer (uint8_t *pResp, uint16_t respLen)
- int unpack_loc_GetServer_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_GetServer_Ind_t *pOutput)

9.5.1 Macro Definition Documentation

9.5.1.1 #define LOC_UINT8_MAX_STRING_SZ 255

9.5.1.2 #define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000

The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.

9.5.1.3 #define LOCEVENTMASKENGINESTATE 0x00000080

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

9.5.1.4 #define LOCEVENTMASKFIXSESSIONSTATE 0x00000100

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

9.5.1.5 #define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.

9.5.1.6 #define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence.

9.5.1.7 #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000

The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, for example, if GPS is turned off or if the network is unavailable.

9.5.1.8 #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000

The control point must enable this mask to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_ CONSTELL_REPORT_CONFIG(Not yet supported).

9.5.1.9 #define LOCEVENTMASKGNSSSVINFO 0x00000002

The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate.

9.5.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040

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

9.5.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020

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

9.5.1.12 #define LOCEVENTMASKINJECTTIMEREQ 0x00000010

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

9.5.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000

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

9.5.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF

Invalid Event Mask

9.5.1.15 #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000

The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing.

9.5.1.16 #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000

The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.

9.5.1.17 #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000

The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data.

9.5.1.18 #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000

The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.

9.5.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008

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

9.5.1.20 #define LOCEVENTMASKNMEA 0x00000004

The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.

9.5.1.21 #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000

The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.

9.5.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001

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

9.5.1.23 #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.).

9.5.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000

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

9.5.1.25 #define LOCEVENTMASKTIMESYNCREQ 0x00000800

The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.

9.5.1.26 #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).

9.5.1.27 #define LOCEVENTMASKWIFIREQ 0x00000200

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

9.5.1.28 #define MAX_LOC_NMEA_STR_LEN 201

9.5.1.29 #define MAX_SENSOR_DATA_LEN 64

9.5.1.30 #define MAX_TEMP_DATA_LEN 64

9.5.2 Enumeration Type Documentation

9.5.2.1 anonymous enum

Enumerator

eQMI_LOC_SESS_STATUS_SUCCESS

eQMI_LOC_SESS_STATUS_IN_PROGRESS

eQMI_LOC_SESS_STATUS_FAILURE

eQMI_LOC_SESS_STATUS_TIMEOUT

9.5.3 Function Documentation

9.5.3.1 int pack_loc_DeleteAssistData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_Delete_Assist_Data_t * *reqArg*)

Delete Assistant Data pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.2 int pack_loc_EventRegister (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_EventRegister_t * *reqArg*)

Event Register pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.3 int pack_loc_SetExtPowerState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SetExtPowerState_t * *reqArg*)

Set Ext Power State pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

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

Set Operation Mode pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.5 int pack_loc_SLQSLOCGetBestAvailPos (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCGetBestAvailPos_t * *reqArg*)

Get Best Avail position pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.6 int pack_loc_SLQSLOCGetServer (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCGetServer_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through unpack_loc_GetServer_Ind callback If successful, the callback also contains the A-GPS server address.

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.7 int pack_loc_SLQSLOCInjectPosition (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectPosition_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.8 int pack_loc_SLQSLOCInjectSensorData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectSensorData_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.9 int pack_loc_SLQSLOCInjectUTCTime (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectUTCTime_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.10 int pack_loc_SLQSLOCSetCradleMountConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCSetCradleMountConfig_t * *reqArg*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.11 int pack_loc_Start (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_Start_t * *reqArg*)

LOC Start pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.12 int pack_loc_Stop (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_Stop_t * *reqArg*)

Loc Stop pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.13 int unpack_loc_BestAvailPos_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_BestAvailPos_Ind_t * *pOutput*)

Loc Best Avial position Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.14 int unpack_loc_DeleteAssistData (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Delete_Assist_Data_t * *pOutput*)

Delete Assistant Data unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.15 int unpack_loc_DeleteAssistData_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_DeleteAssistData_Ind_t * *pOutput*)

Unpack the status of delete the location engine assistance data

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.16 int unpack_loc_EngineState_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EngineState_Ind_t * *pOutput*)

Loc Engine State Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.17 int unpack_loc_EventNMEA_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EventNMEA_Ind_t * *pOutput*)

Unpack the NMEA Event Indication.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.18 int unpack_loc_EventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EventRegister_t * *pOutput*)

Event Register unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.19 int unpack_loc_GetServer_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_GetServer_Ind_t * *pOutput*)

Unpack the Get Server Indication.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.20 int unpack_loc_GnssSvInfo_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_GnssSvInfo_Ind_t * *pOutput*)

Unpack the GNSS SV Info Indication.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.21 int unpack_loc_PositionRpt_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_PositionRpt_Ind_t * *pOutput*)

Loc Position Report Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.22 int unpack_loc_SetExtPowerConfig_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerConfig_Ind_t * *pOutput*)

Loc Set External Power Configure Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.23 int unpack_loc_SetExtPowerState (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerState_t * *pOutput*)

Set Ext Power State unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.24 int unpack_loc_SetOperationMode (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetOperationMode_t * *pOutput*)

Set Operation Mode unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.25 int unpack_loc_SetOperationMode_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetOperationMode_Ind_t * *pOutput*)

Unpack the engine to use the specified operation mode.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.26 int unpack_loc_SLQSLOCGetBestAvailPos (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCGetBestAvailPos_t * *pOutput*)

Get Best Avail position unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.27 int unpack_loc_SLQSLOCGetServer (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_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

9.5.3.29 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

9.5.3.30 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

9.5.3.31 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

9.5.3.32 int unpack_loc_Start (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Start_t * *pOutput*)

Loc Start unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.5.3.33 int unpack_loc_Stop (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Stop_t * *pOutput*)

Loc Stop unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6 nas.h File Reference

Data Structures

- struct unpack_nas_GetSignalStrengths_t
- struct unpack_nas_SLQSGetSysSelectionPref_t
- struct nas_netSelectionPref
- struct nas_acqOrderPref
- struct nas_CSGID
- struct pack_nas_SLQSSetSysSelectionPref_t
- struct 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*)

Retrieves information about the access overload class (ACCOLC) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.2 int pack_nas_GetANAAAAuthenticationStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

AN-AAA authentication status of the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.3 int pack_nas_GetCDMANetworkParameters (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.4 int pack_nas_GetHomeNetwork (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get home network pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.5 int pack_nas_GetHomeNetwork3GPP2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get Home Network 3GPP2 Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.6 int pack_nas_GetNetworkPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

9.6.3.7 int pack_nas_GetRFInfo (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen*)

get rf info pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.8 int pack_nas_GetServingNetwork (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.9 int pack_nas_GetServingNetworkCapabilities (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.10 int pack_nas_GetSignalStrengths (pack_qmi_t * *pCtx,* uint8_t * *pReq,* uint16_t * *pLen*)

get signal strengths pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.11 int pack_nas_InitiateDomainAttach (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_InitiateDomainAttach_t * *pReqParam*)

Initiates a domain attach/detach of the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.12 int pack_nas_PerformNetworkScan (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.13 int pack_nas_SetACCOLC (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetACCOLC_t *reqParam*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.14 int pack_nas_SetCDMANetworkParameters (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetCDMANetworkParameters_t * *pReqParam*)

CDMA Network Parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.15 int pack_nas_SetLURejectCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pBenable*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pBenable*	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.16 int pack_nas_SetNetworkPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SetNetworkPreference_t * reqArg)

Sets the network registration preference. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use pack_nas_SLQSSetSysSelectionPref() for new firmware versions and new modules

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

9.6.3.17 int pack_nas_SetRFInfoCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pBenable*	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.18 int pack_nas_SLQSConfigSigInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSConfigSigInfo_t * pReqParam)

Config Sig Info Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.19 int pack_nas_SLQSGetErrorRate (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Get Error Rate Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.20 int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

9.6.3.21 int pack_nas_SLQSGetNetworkTime (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.22 int pack_nas_SLQSGetOperatorNameData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Operator Name Data Parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.23 int pack_nas_SLQSGetPLMNName (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSGetPLMNName_t * *reqArg*)

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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.25 int pack_nas_SLQSGetSignalStrength (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint16_t *reqMask*)

Queries the current signal strength as measured by the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use unpack_nas_SLQSNasGetSigInfo() for new firmware versions and new modules

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqMask*	request mask for fetching extra signal info

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.26 int pack_nas_SLQSGetSysInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.27 int pack_nas_SLQSGetSysSelectionPref (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Queries the different system selection preferences of the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.28 int pack_nas_SLQSInitiateNetworkRegistration (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSInitiateNetworkRegistration_t * *pReqParam*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.29 int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasConfigSigInfo2_t * *pReqParam*)

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.30 int pack_nas_SLQSNasGet3GPP2Subscription (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasGet3GPP2Subscription_t * *reqArg*)

3GPP2 Subscription Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.31 int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.33 int pack_nas_SLQSNasGetSigInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get sig info pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.34 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*)

Used to Registers/De-registers for different NAS (Network access service) indications.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

This function is used by a control point to register/deregister for different QMI_NAS indications. The control point's registration state variables, controlling registration for indications, are modified to reflect the settings indicated in the parameters that are present in the request message. At least one optional parameter must be present in the request.

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.36 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	*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

9.6.3.37 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.38 int pack_nas_SLQSNasSwiModemStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

This function requests the device to return the current status of modem.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.39 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.40 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.41 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.42 int pack_nas_SLQSSetSysSelectionPref (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSysSelectionPref_t * pReqParam)

Sets the different system selection preferences of the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values4780

9.6.3.43 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	*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

9.6.3.44 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	*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

9.6.3.45 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	*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

9.6.3.46 int pack_nas_SLQSSwiGetLteCQI (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

9.6.3.47 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.49 int pack_nas_SLQSSwiPSDetach (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSSwiPSDetach_t * *pReqParam*)

PS Connection Detach Parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.50 int unpack_nas_GetACCOLC (uint8_t * *pResp*, uint16_t *respLen*, uint8_t * *pAccolc*)

Retrieves information about the access overload class (ACCOLC) unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pAccolc*	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*)

AN-AAA authentication status of the device unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pAuthStatus*	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*)

Retrieves information about the home network of the device unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.54 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	*pOutput*	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*)

Sets the network registration preference unpack

Parameters

in	*pResp*	qmi resp from modem
in	*respLen*	qmi resp length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.56 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	*pOutput*	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*)

Provides information about the system that provides service to the device. 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 unpack_nas_SLQSGetSysInfo() for new firmware versions and new modules

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.58 int unpack_nas_GetServingNetworkCapabilities (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetServingNetworkCapabilities_t * *pOutput*)

Returns information regarding the data capabilities of the system that currently provides service to the device.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.59 int unpack_nas_GetSignalStrengths (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetSignalStrengths_t * *pOutput*)

Returns the available signal strengths (in dBm) as measured by the device in an array. This will also provides the corresponding radio interfaces in an array.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	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*)

Performs scan for available networks and scans for RAT info as well.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.62 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	*pOutput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.65 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	*pOutput*	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.68 int unpack_nas_SetNetworkPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNetworkPreference_t * *pOutput*)

Sets the network registration preference unpack

Parameters

in	*pResp*	qmi resp from modem
in	*respLen*	qmi resp length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.69 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	*pOutput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.71 int unpack_nas_SetServingSystemCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetServingSystemCallback_ind_t * *pOutput*)

Set Serving system indication unpack

Parameters

in	*pResp*	qmi indication from modem
in	*respLen*	qmi indication length
out	*pOutput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.72 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	*pOutput*	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*)

Get LTE CPHY Carrier Info unpack

Parameters

in	*pResp*	qmi resp from modem
in	*respLen*	qmi resp length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.75 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	*pOutput*	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_SLQSGetServing-System_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

9.6.3.79 int unpack_nas_SLQSGetSignalStrength (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSignal-Strength_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

9.6.3.80 int unpack_nas_SLQSGetSysInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysInfo_t * *pOutput*)

Provides the system information. This API is preferred when trying to get the service status info and serving system info. The function unpack_nas_SLQSGetServingSystem_t() reports similar NAS information, but it is deprecated. Please refer to the header description of unpack_nas_SLQSGetServingSystem_t() for more information.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

This API queries current serving system information, including registration information and system property. The registration information for all RATs specified in the mode capability setting are included regardless of registration status. The RAT-specific system property are included only for RATs that are specified in the mode capability setting and which are not in either No Service or Power Save modes.

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.81 int unpack_nas_SLQSGetSysSelectionPref (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysSelectionPref_t * *pOutput*)

Queries the different system selection preferences of the device unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.82 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	*pOutput*	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	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

This command queries the signal strength information for currently active RATs. Information is reported only if the corresponding RATs have signal strength values to be reported. If no signal strength information is available for any RAT, the response message contains only the mandatory response message

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.88 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	*pOutput*	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.91 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	*pOutput*	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.92 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	*pOutput*	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*)

This function requests the device to return the current status of modem.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.95 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	*pOutput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.96 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	*pOutput*	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	*pOutput*	network timer indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.99 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*)

Sets the different system selection preferences of the device unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.102 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	*pOutput*	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.103 int unpack_nas_SLQSSwiGetHDRPersonality (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetHDRPersonality_t * *pOutput*)

HDR Personality Parameters unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.6.3.104 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	*pOutput*	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	*pOutput*	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	*pOutput*	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	*pOutput*	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	*pOutput*	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
- struct pack_pds_StartPDSTrackingSessionExt_t
- struct pack_pds_PDSInjectTimeReference_t
- struct unpack_pds_GetXTRAAutomaticDownload_t
- struct pack_pds_SetXTRAAutomaticDownload_t
- struct unpack_pds_GetXTRANetwork_t
- struct pack_pds_SetXTRANetwork_t
- struct unpack_pds_GetXTRAValidity_t
- struct unpack_pds_GetServiceAutomaticTracking_t
- struct pack_pds_SetServiceAutomaticTracking_t
- struct pack_pds_ResetPDSData_t
- struct pack_pds_SLQSSetAGPSConfig_t
- struct pack_pds_SLQSPDSInjectAbsoluteTimeReference_t
- struct pack_pds_SLQSGetAGPSConfig_t
- struct unpack_pds_SLQSGetAGPSConfig_t
- struct pack_pds_SLQSPDSInjectPositionData_t
- struct unpack_pds_SLQSGetGPSStateInfo_t
- struct pack_pds_SLQSSetPositionMethodState_t
- struct pack_pds_SetEventReportCallback_t
- struct unpack_pds_SetEventReport_Ind_t
- struct unpack_pds_SetPdsState_Ind_t

Functions

- int pack_pds_GetPDSState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetPDSState (uint8_t *pResp, uint16_t respLen, unpack_pds_GetPDSState_t *pOutput)
- int pack_pds_SetPDSState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetPDSState_t *pReqParam)
- int unpack_pds_SetPDSState (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetPDSDefaults (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetPDSDefaults (uint8_t *pResp, uint16_t respLen, unpack_pds_GetPDSDefaults_t *pOutput)
- int pack_pds_SetPDSDefaults (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetPDSDefaults_t *pReqParam)
- int unpack_pds_SetPDSDefaults (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetPortAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetPortAutomaticTracking (uint8_t *pResp, uint16_t respLen, unpack_pds_GetPortAutomaticTracking_t *pOutput)
- int pack_pds_SetPortAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetPortAutomaticTracking_t *pReqParam)
- int unpack_pds_SetPortAutomaticTracking (uint8_t *pResp, uint16_t respLen)
- int pack_pds_StartPDSTrackingSessionExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_StartPDSTrackingSessionExt_t *pReqParam)
- int unpack_pds_StartPDSTrackingSessionExt (uint8_t *pResp, uint16_t respLen)
- int pack_pds_StopPDSTrackingSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_StopPDSTrackingSession (uint8_t *pResp, uint16_t respLen)
- int pack_pds_PDSInjectTimeReference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_PDSInjectTimeReference_t *pReqParam)
- int unpack_pds_PDSInjectTimeReference (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetXTRAAutomaticDownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRAAutomaticDownload (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRAAutomaticDownload_t *pOutput)
- int pack_pds_SetXTRAAutomaticDownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetXTRAAutomaticDownload_t *pReqParam)
- int unpack_pds_SetXTRAAutomaticDownload (uint8_t *pResp, uint16_t respLen)

- int pack_pds_GetXTRANetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRANetwork (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRANetwork_t *pOutput)
- int pack_pds_SetXTRANetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetXTRANetwork_t *pReqParam)
- int unpack_pds_SetXTRANetwork (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetXTRAValidity (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRAValidity (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRAValidity_t *pOutput)
- int pack_pds_ForceXTRADownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_ForceXTRADownload (uint8_t *pResp, uint16_t respLen)
- int pack_pds_GetServiceAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetServiceAutomaticTracking (uint8_t *pResp, uint16_t respLen, unpack_pds_GetServiceAutomaticTracking_t *pOutput)
- int pack_pds_SetServiceAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetServiceAutomaticTracking_t *pReqParam)
- int unpack_pds_SetServiceAutomaticTracking (uint8_t *pResp, uint16_t respLen)
- int pack_pds_ResetPDSData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_ResetPDSData_t *pReqParam)
- int unpack_pds_ResetPDSData (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSSetAGPSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSSetAGPSConfig_t *pReqParam)
- int unpack_pds_SLQSSetAGPSConfig (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSPDSInjectAbsoluteTimeReference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSPDSInjectAbsoluteTimeReference_t *pReqParam)
- int unpack_pds_SLQSPDSInjectAbsoluteTimeReference (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSGetAGPSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSGetAGPSConfig_t *pReqParam)
- int unpack_pds_SLQSGetAGPSConfig (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSGetAGPSConfig_t *pOutput)
- int pack_pds_SLQSPDSInjectPositionData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSPDSInjectPositionData_t *pReqParam)
- int unpack_pds_SLQSPDSInjectPositionData (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSPDSDeterminePosition (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_SLQSPDSDeterminePosition (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SLQSGetGPSStateInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_SLQSGetGPSStateInfo (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSGetGPSStateInfo_t *pOutput)
- int pack_pds_SLQSSetPositionMethodState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSSetPositionMethodState_t *pReqParam)
- int unpack_pds_SLQSSetPositionMethodState (uint8_t *pResp, uint16_t respLen)
- int pack_pds_SetEventReportCallback (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen, pack_pds_SetEventReportCallback_t *pReqParam)
- int unpack_pds_SetEventReportCallback (uint8_t *pResp, uint16_t respLen)
- int unpack_pds_SetEventReport_Ind (uint8_t *pResp, uint16_t respLen, unpack_pds_SetEventReport_Ind_t *pOutput)
- int unpack_pds_SetPdsState_Ind (uint8_t *pResp, uint16_t respLen, unpack_pds_SetPdsState_Ind_t *pOutput)

9.7.1 Function Documentation

9.7.1.1 int pack_pds_ForceXTRADownload (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Forces the XTRA database to be downloaded to the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.2 int pack_pds_GetPDSDefaults (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get the default tracking session configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.3 int pack_pds_GetPDSState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get current PDS state pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.4 int pack_pds_GetPortAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Returns the automatic tracking configuration for the NMEA COM port pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.5 int pack_pds_GetServiceAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

automatic tracking state for the service pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.6 int pack_pds_GetXTRAAutomaticDownload (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

XTRA automatic database download configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.7 int pack_pds_GetXTRANetwork (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

XTRA WWAN network preference parameter pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.8 int pack_pds_GetXTRAValidity (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

XTRA database validity period parameter pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.9 int pack_pds_PDSInjectTimeReference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_PDSInjectTimeReference_t * *pReqParam*)

Inject system time parameters pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.10 int pack_pds_ResetPDSData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_ResetPDSData_t * *pReqParam*)

Resets the specified PDS data pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.11 int pack_pds_SetEventReportCallback (pack_qmi_t * *pCtx*, uint8_t * *pReq*, uint16_t * *pLen*, pack_pds_SetEventReportCallback_t * *pReqParam*)

Enables/disables the PDS Event report callback function pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReq*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.12 int pack_pds_SetPDSDefaults (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPDSDefaults_t * *pReqParam*)

Sets the default tracking session configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.13 int pack_pds_SetPDSState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPDSState_t * *pReqParam*)

Sets the PDS state pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.14 int pack_pds_SetPortAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPortAutomaticTracking_t * *pReqParam*)

Sets the automatic tracking configuration for the NMEA COM port pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.15 int pack_pds_SetServiceAutomaticTracking (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetServiceAutomaticTracking_t * *pReqParam*)

Sets the automatic tracking state for the service pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.16 int pack_pds_SetXTRAAutomaticDownload (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetXTRAAutomaticDownload_t * *pReqParam*)

Sets the XTRA automatic database download configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.17 int pack_pds_SetXTRANetwork (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetXTRANetwork_t * *pReqParam*)

Sets the XTRA WWAN network preference pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.18 int pack_pds_SLQSGetAGPSConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSGetAGPSConfig_t * *pReqParam*)

Gets the PDS AGPS (MS-based) configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.19 int pack_pds_SLQSGetGPSStateInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Queries the MSM GPS server for receiver state information pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.20 int pack_pds_SLQSPDSDeterminePosition (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.21 int pack_pds_SLQSPDSInjectAbsoluteTimeReference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSPDSInjectAbsoluteTimeReference_t * *pReqParam*)

Injects a absolute time reference into the PDS engine pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.22 int pack_pds_SLQSPDSInjectPositionData (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSPDSInjectPositionData_t * *pReqParam*)

Injects position data into the PDS engine pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.23 int pack_pds_SLQSSetAGPSConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSSetAGPSConfig_t * *pReqParam*)

Sets the PDS AGPS (MS-based) configuration pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.24 int pack_pds_SLQSSetPositionMethodState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSSetPositionMethodState_t * *pReqParam*)

Sets the state of positioning methods for the device pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.25 int pack_pds_StartPDSTrackingSessionExt (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_StartPDSTrackingSessionExt_t * *pReqParam*)

Starts a PDS tracking session pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.26 int pack_pds_StopPDSTrackingSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Stops a PDS tracking session pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.27 int unpack_pds_ForceXTRADownload (uint8_t * *pResp*, uint16_t *respLen*)

Forces the XTRA database to be downloaded to the device unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.28 int unpack_pds_GetPDSDefaults (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPDSDefaults_t * *pOutput*)

Get the default tracking session configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.29 int unpack_pds_GetPDSState (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPDSState_t * *pOutput*)

Get current PDS state unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.30 int unpack_pds_GetPortAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPortAutomaticTracking_t * *pOutput*)

Returns the automatic tracking configuration for the NMEA COM port unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.31 int unpack_pds_GetServiceAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetServiceAutomaticTracking_t * *pOutput*)

automatic tracking state for the service unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.32 int unpack_pds_GetXTRAAutomaticDownload (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetXTRAAutomaticDownload_t * *pOutput*)

XTRA automatic database download configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.33 int unpack_pds_GetXTRANetwork (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetXTRANetwork_t * *pOutput*)

XTRA WWAN network preference parameter unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.34 int unpack_pds_GetXTRAValidity (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetXTRAValidity_t * *pOutput*)

XTRA database validity period parameter unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.35 int unpack_pds_PDSInjectTimeReference (uint8_t * *pResp*, uint16_t *respLen*)

Inject system time parameters unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.36 int unpack_pds_ResetPDSData (uint8_t * *pResp*, uint16_t *respLen*)

Resets the specified PDS data unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.37 int unpack_pds_SetEventReport_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetEventReport_Ind_t * *pOutput*)

PDS Event Report Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.38 int unpack_pds_SetEventReportCallback (uint8_t * *pResp*, uint16_t *respLen*)

Enables/disables the PDS Event report callback function unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.39 int unpack_pds_SetPDSDefaults (uint8_t * *pResp*, uint16_t *respLen*)

Sets the default tracking session configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.40 int unpack_pds_SetPDSState (uint8_t * *pResp*, uint16_t *respLen*)

Sets the PDS state unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.41 int unpack_pds_SetPdsState_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetPdsState_Ind_t * *pOutput*)

PDS State Indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.42 int unpack_pds_SetPortAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*)

Sets the automatic tracking configuration for the NMEA COM port unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.43 int unpack_pds_SetServiceAutomaticTracking (uint8_t * *pResp*, uint16_t *respLen*)

Sets the automatic tracking state for the service unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.44 int unpack_pds_SetXTRAAutomaticDownload (uint8_t * *pResp*, uint16_t *respLen*)

Sets the XTRA automatic database download configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.45 int unpack_pds_SetXTRANetwork (uint8_t * *pResp*, uint16_t *respLen*)

Sets the XTRA WWAN network preference unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.46 int unpack_pds_SLQSGetAGPSConfig (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSGetAGPSConfig_t * *pOutput*)

Gets the PDS AGPS (MS-based) configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.47 int unpack_pds_SLQSGetGPSStateInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSGetGPSStateInfo_t * *pOutput*)

Queries the MSM GPS server for receiver state information unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.48 int unpack_pds_SLQSPDSDeterminePosition (uint8_t * *pResp*, uint16_t *respLen*)

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.49 int unpack_pds_SLQSPDSInjectAbsoluteTimeReference (uint8_t * *pResp*, uint16_t *respLen*)

Injects a absolute time reference into the PDS engine unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.50 int unpack_pds_SLQSPDSInjectPositionData (uint8_t * *pResp*, uint16_t *respLen*)

Injects position data into the PDS engine unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.51 int unpack_pds_SLQSSetAGPSConfig (uint8_t * *pResp*, uint16_t *respLen*)

Sets the PDS AGPS (MS-based) configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.52 int unpack_pds_SLQSSetPositionMethodState (uint8_t * *pResp*, uint16_t *respLen*)

Sets the state of positioning methods for the device unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.53 int unpack_pds_StartPDSTrackingSessionExt (uint8_t * *pResp*, uint16_t *respLen*)

Starts a PDS tracking session unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.7.1.54 int unpack_pds_StopPDSTrackingSession (uint8_t * *pResp*, uint16_t *respLen*)

Stops a PDS tracking session unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.8 qaCbkCatEventReportInd.h File Reference

Data Structures

- struct CatEventIDDataTlv
- struct CatAlPhaIdentifierTlv
- struct CatEventListTlv
- struct CatRefreshTlv
- struct CatEndProactiveSessionTlv
- union currentCatEvent
- struct CatCommonEventTlv
- struct QmiCbkCatEventStatusReportInd

Macros

- #define QMI_MAX_CAT_EVENT_DATA_LENGTH 255
- #define QMI_CAN_COMMON_EVENT_TLV_NUMBER 11

Enumerations

- enum eQMI_CAT_EVENT_REPORT_IND_TLV {
  eTLV_CBK_DISPLAY_TEXT = 0x10,
  eTLV_CBK_GET_IN_KEY = 0x11,
  eTLV_CBK_GET_INPUT = 0x12,
  eTLV_CBK_SETUP_MENU = 0x13,
  eTLV_CBK_SELECT_ITEM = 0x14,
  eTLV_CBK_ALPHA_IDENTIFIER = 0x15,
  eTLV_CBK_SETUP_EVENT_LIST = 0x16,
  eTLV_CBK_SETUP_IDLE_MODE_TEXT = 0x17,
  eTLV_CBK_LANGUAGE_NOTIFICATION = 0x18,
  eTLV_CBK_REFRESH = 0x19,
  eTLV_CBK_END_PROACTIVE_SESSION = 0x1A }
- enum eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH {
  eTLV_SETUP_EVENT_LIST_LENGTH = 0x04,
  eTLV_REFRESH_LENGTH = 0x03,
  eTLV_END_PROACTIVE_SESSION_LENGTH = 0x01 }

Functions

- enum eQCWWANError UpkQmiCbkCatEventReportInd (BYTE *pMdmResp, struct 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 SLQSGetAudioVoITLBConfig (GetAudioVoITLBConfigReq * *pGetAudioVoITLBCfgReq*, GetAudioVoITLBConfigResp * *pGetAudioVoITLBCfgResp*)

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
- struct IPv4Info
- struct IPv6Info
- struct urlAddr
- struct QmiCbkLocGetServerInd

Macros

- #define SIGSTRENGTH_THRESHOLD_ARR_SZ 5
- #define QMI_WMS_MAX_PAYLOAD_LENGTH 256
- #define QMI_ETWS_MAX_PAYLOAD_LENGTH 1254 /∗ Qualcomm defined max ∗/
- #define QMI_MAX_VOICE_NUMBER_LENGTH 81
- #define MAX_NO_OF_UUSINFO 20
- #define MAXUSSDLENGTH 182
- #define MAX_NO_OF_CALLS 20
- #define CBK_ENABLE_EVENT 0x01
- #define CBK_DISABLE_EVENT 0x00
- #define CBK_NOCHANGE 0xFF
- #define MAX_NO_OF_APPLICATIONS 10
- #define MAX_NO_OF_SLOTS 5
- #define MAX_NO_OF_FILES 255
- #define MAX_PATH_LENGTH 255
- #define EVENT_MASK_CARD 0x00000001
- #define EVENT_MASK_PHY_SLOT_STATUS 0x00000010
- #define EVENT_MASK_DEREGISTER_ALL 0x00000000
- #define REGISTER_EVENT 0x01
- #define DEREGISTER_EVENT 0x00
- #define FIRST_INSTANCE 0x00
- #define SECOND_INSTANCE 0x01
- #define THIRD_INSTANCE 0x02
- #define INVALID_INSTACNE 0x08
- #define REGISTER_SRV 0x01
- #define DEREGISTER_SRV 0x00
- #define WDS_SRV 0x01
- #define NAS_SRV 0x02

- #define PDS_SRV 0x04
- #define VOICE_SRV 0x08
- #define NUM_OF_SET 0xFF
- #define IPV4 4
- #define IPV6 6
- #define IPV4V6 7
- #define LOC_EVENT_MASK_ENG_STATE 0x00000080
- #define LOC_EVENT_MASK_TIME_SYNC 0x00000800
- #define LOC_EVENT_MASK_INJECT_TIME 0x00000010
- #define LOC_EVENT_MASK_SENSOR_STREAM 0x00000400
- #define LOC_EVENT_POSITION_REPORT 0x00000001
- #define LOC_EVENT_MASK_GNSS_SV_INFO 0x00000002
- #define LOC_EVENT_MASK_NMEA 0x00000004
- #define MAX_NMEA_STR_LEN 201
- #define DHCP_MAX_NUM_OPTIONS 30
- #define DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */
- #define MAX_MITIGATION_DEV_ID_LEN 255
- #define MAX_URL_ADDR_STR_LEN 256
- #define MAX_RADIO_INTERFACE_LIST 255
- #define USSD_DCS_ASCII 0x01 /* ASCII coding scheme */
- #define USSD_DCS_8BIT 0x02 /* 8-bit coding scheme */
- #define USSD_DCS_UCS2 0x03 /* UCS2 coding scheme */

Typedefs

- typedef void(* tFNSLQSSessionState)(slqsSessionStateInfo *pSessionStateInfo)
- typedef void(* tFNSLQSWDSEvent)(slqsWdsEventInfo *pWdsEventInfo)
- typedef void(* tFNPower)(ULONG operatingMode)
- typedef void(* tFNActivationStatus)(ULONG activationStatus)
- typedef void(* tFNMobileIPStatus)(ULONG mipStatus)
- typedef void(* tFNRoamingIndicator)(ULONG roaming)
- typedef void(* tFNDataCapabilities)(BYTE dataCapsSize, BYTE *pDataCaps)
- typedef void(* tFNSignalStrength)(INT8 signalStrength, ULONG radioInterface)
- typedef void(* tFNRFInfo)(ULONG radioInterface, ULONG activeBandClass, ULONG activeChannel)
- typedef void(* tFNLUReject)(ULONG serviceDomain, ULONG rejectCause)
- typedef void(* tFNNewSMS)(ULONG storageType, ULONG messageIndex)
- typedef enum SMSEventType eSMSEventType
- typedef struct SMSMTMessage SMSMTMessageInfo
- typedef struct SMSTransferRouteMTMessage SMSTransferRouteMTMessageInfo
- typedef struct SMSMessageMode SMSMessageModeInfo
- typedef struct SMSEtwsMessage SMSEtwsMessageInfo
- typedef struct SMSEtwsPlmn SMSEtwsPlmnInfo
- typedef struct SMSCAddress SMSCAddressInfo
- typedef struct SMSOnIMS SMSOnIMSInfo
- typedef struct SMSEventInfo_s SMSEventInfo
- typedef void(* tFNSMSEvents)(SMSEventInfo *pSMSEventInfo)
- typedef void(* tFNNewNMEA)(LPCSTR pNMEA)
- typedef void(* tFNPDSState)(ULONG enabledStatus, ULONG trackingStatus)
- typedef void(* tFNCATEvent)(ULONG eventID, ULONG eventLen, BYTE *pEventData)
- typedef enum device_state_enum eDevState
- typedef void(* tFNDeviceStateChange)(eDevState device_state)
- typedef void(* tFNNet)(ULONG q_depth, BYTE isThrottle, BYTE instanceId)

- typedef void(* tFNFwDldCompletion)(ULONG fwdld_completion_status)
- typedef void(* tFNSLQSOMADMAlert)(ULONG eventType, BYTE *pEventFields)
- typedef void(* tFNOMADMState)(ULONG sessionState, ULONG failureReason)
- typedef void(* tFNServingSystem)(struct ServingSystemInfo *pServingSystem, struct RoamingInfo *pRoamingInfo)
- typedef void(* tFNBandPreference)(ULONGLONG band_pref)
- typedef void(* tFNUSSDRelease)(void)
- typedef void(* tFNUSSDNotification)(ULONG type, BYTE *pNetworkInfo)
- typedef void(* tFNSLQSSignalStrengths)(struct SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo)
- typedef void(* tFNSUPSNotification)(voiceSUPSNotification *pVoiceSUPSNotification)
- typedef void(* tFNSDKTerminated)(BYTE *psReason)
- typedef void(* tFNAllCallStatus)(voiceSetAllCallStatusCbkInfo *pVoiceSetAllCallStatusCbkInfo)
- typedef struct _transLayerInfoNotification transLayerNotification
- typedef void(* tFNtransLayerInfo)(transLayerNotification *pTransLayerNotification)
- typedef struct _transNWRegInfoNotification transNWRegInfoNotification
- typedef void(* tFNtransNWRegInfo)(transNWRegInfoNotification *pTransNWRegInfoNotification)
- typedef void(* tFNSysSelectionPref)(sysSelectPrefInfo *pSysSelectPrefInfo)
- typedef void(* tFNUIMRefresh)(UIMRefreshEvent *pUIMRefreshEvent)
- typedef void(* tFNUIMStatusChangeInfo)(UIMStatusChangeInfo *pUIMStatusChangeInfo)
- typedef void(* tFNPrivacyChange)(voicePrivacyInfo *pVoicePrivacyInfo)
- typedef void(* tFNDTMFEvent)(voiceDTMFEventInfo *pVoiceDTMFEventInfo)
- typedef void(* tFNSUPSInfo)(voiceSUPSInfo *pVoiceSUPSInfo)
- typedef void(* tFNSysInfo)(nasSysInfo *pNasSysInfo)
- typedef void(* tFNNetworkTime)(nasNetworkTime *pNasNetworkTime)
- typedef union sessionInfo sessionInformation
- typedef union sessionInfoExt sessionInformationExt
- typedef void(* tFNMemoryFull)(SMSMemoryInfo *pSMSMemoryFullInfo)
- typedef void(* tFNOTASPStatus)(voiceOTASPStatusInfo *pVoiceOTASPStatusInfo)
- typedef void(* tFNInfoRec)(voiceInfoRec *pVoiceInfoRec)
- typedef void(* tFNMessageWaiting)(msgWaitingInfo *pSMSMessageWaitingInfo)
- typedef void(* tFNSLQSQOSEvent)(BYTE instance, QosFlowInfo *pFlowInfo)
- typedef void(* tFNQosStatus)(BYTE instance, ULONG id, BYTE status, BYTE event, BYTE reason)
- typedef void(* tFNQosNWStatus)(BYTE status)
- typedef void(* tFNQosPriEvent)(WORD event)
- typedef void(* tFNSigInfo)(nasSigInfo *pNasSigInfo)
- typedef struct _modemTempNotification modemTempNotification
- typedef void(* tFNModemTempInfo)(modemTempNotification *pModemTempNotification)
- typedef struct _packetSrvStatus packetSrvStatus
- typedef void(* tFNPacketSrvState)(packetSrvStatus *pPacketSrvStatus)
- typedef void(* tFNHDRPersonaity)(HDRPersonalityInd *pHDRPers)
- typedef void(* tFNImsSIPConfig)(imsSIPConfigInfo *pImsSIPConfigInfo)
- typedef void(* tFNImsRegMgrConfig)(imsRegMgrConfigInfo *pImsRegMgrConfigInfo)
- typedef void(* tFNImsSMSConfig)(imsSMSConfigInfo *pImsSMSConfigInfo)
- typedef void(* tFNImsUserConfig)(imsUserConfigInfo *pImsUserConfigInfo)
- typedef void(* tFNImsVoIPConfig)(imsVoIPConfigInfo *pImsVoIPConfigInfo)
- typedef void(* tFNUSSDNoWaitIndication)(USSDNoWaitIndicationInfo *pNetworkInfo)
- typedef void(* tFNDUNCallInfo)(DUNCallInfoInd *pDUNCallInfo)
- typedef void(* tFNDataSysStatus)(CurrDataSysStat *pCurrDataSysStat)
- typedef struct SMSAsyncRawSend_s SMSAsyncRawSend
- typedef void(* tFNAsyncRawSend)(SMSAsyncRawSend *pSMSAsyncRawSend)
- typedef struct LteNasReleaseInfo_s LteNasReleaseInfo

- typedef struct SwiOTAMsg_s SwiOTAMsg
- typedef void(* tFNASwiOTAMsg)(SwiOTAMsg *pSwiOTAMsg)
- typedef void(* tFNNewGPS)(double dLongitude, double dLatitude, BYTE session_status, ULONG pos_src)
- typedef void(* tFNNewRMTransferStatistics)(QmiCbkWdsStatisticsIndState *pMsg)
- typedef void(* tFNDHCPv4ClientLeaseStatus)(BYTE instance, WdsDHCPv4ClientLeaseInd *pMsg)
- typedef void(* tFNSetCradleMount)(QmiCbkLocCradleMountInd *pSetLocCradleMount)
- typedef void(* tFNSetEventTimeSync)(QmiCbkLocEventTimeSyncInd *pSetLocEventTimeSync)
- typedef void(* tFNInjectTimeStatus)(QmiCbkLocInjectTimeInd *pLocInjectTime)
- typedef struct accelAcceptReady_s accelAcceptReady
- typedef struct gyroAcceptReady_s gyroAcceptReady
- typedef struct accelTempAcceptReady_s accelTempAcceptReady
- typedef struct gyroTempAcceptReady_s gyroTempAcceptReady
- typedef void(* tFNSensorStreaming)(QmiCbkLocSensorStreamingInd *pLocSensorStream)
- typedef void(* tFNInjectSensorData)(QmiCbkLocInjectSensorDataInd *pLocInjectSensorData)
- typedef struct precisionDilution_s precisionDilution
- typedef struct gpsTime_s gpsTime
- typedef struct sensorDataUsage_s sensorDataUsage
- typedef struct svUsedforFix_s svUsedforFix
- typedef void(* tFNEventPosition)(QmiCbkLocPositionReportInd *pLocPositionReport)
- typedef void(* tFNBestAvailPos)(QmiCbkLocBestAvailPosInd *pBestAvailPos)
- typedef void(* tFNOpMode)(ULONG mode)
- typedef void(* tFNImsaRegStatus)(imsaRegStatusInfo *pImsaRegStatusInfo)
- typedef void(* tFNImsaSvcStatus)(imsaSvcStatusInfo *pImsaSvcStatusInfo)
- typedef void(* tFNImsaRatStatus)(imsaRatStatusInfo *pImsaRatStatusInfo)
- typedef void(* tFNImsaPdpStatus)(imsaPdpStatusInfo *pImsaPdpStatusInfo)
- typedef void(* tFNGnssSvInfo)(gnssSvInfoNotification *pGnssSvInfoNotification)
- typedef void(* tFNDelAssistData)(delAssistDataStatus *pAssistDataNotification)
- typedef void(* tFNASwiLTECphyCaInfo)(QmiCbkNasLTECphyCaInfo *pQmiCbkNasLTECphyCaInfo)
- typedef void(* tFNRankIndicator)(RankIndicatorInd *pRankIndicatorInd)
- typedef void(* tFNInjectUTCTime)(QmiCbkLocInjectUTCTimeInd *pInjectUTCTimeNotification)
- typedef void(* tFNInjectPosition)(QmiCbkLocInjectPositionInd *pInjectPositionNotification)
- typedef void(* tFNCbkUimSlotStatusChangeInd)(UIMSlotStatusChangeInfo *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)
- typedef void(* tFNEventNmea)(CHAR *pNMEAData)
- typedef void(* tFNGetServer)(QmiCbkLocGetServerInd *pGetLocServerInfo)

Enumerations

- enum eQaQMIService {
  eQA_QMI_SVC_WDS = 0x01,
  eQA_QMI_SVC_NAS = 0x03,
  eQA_QMI_SVC_NA = 0xFF }
- enum SMSEventType {
  SMS_EVENT_MT_MESSAGE = 0x01,
  SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE = 0x02,
  SMS_EVENT_MESSAGE_MODE = 0x04,
  SMS_EVENT_ETWS = 0x08,
  SMS_EVENT_ETWS_PLMN = 0x10,
  SMS_EVENT_SMSC_ADDRESS = 0x20,
  SMS_EVENT_SMS_ON_IMS = 0x40 }
- enum device_state_enum {
  DEVICE_STATE_DISCONNECTED,
  DEVICE_STATE_READY,
  DEVICE_STATE_BOOT,
  DEVICE_STATE_SERIAL_ONLY }
- enum TYPE_OF_SERVICE_DOMAIN {
  SYS_SRV_DOMAIN_NO_SRV = 0x00,
  SYS_SRV_DOMAIN_CS_ONLY = 0x01,
  SYS_SRV_DOMAIN_PS_ONLY = 0x02,
  SYS_SRV_DOMAIN_CS_PS = 0x03,
  SYS_SRV_DOMAIN_CAMPED = 0x04 }

Functions

- ULONG SLQSSetSessionStateCallback (tFNSLQSSessionState pCallback)
- ULONG SLQSSetWdsEventCallback (tFNSLQSWDSEvent pCallback, BYTE interval, BYTE instanceid, BYTE ipfamily)
- ULONG SLQSSetWdsTransferStatisticCallback (tFNSLQSWDSEvent pXferStatsCb, BYTE interval, BYTE instanceid, BYTE ipfamily)
- ULONG iSLQSSetWdsFirstInstEventCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsSecondInstEventCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsThirdInstEventCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsXferStatsFirstInstCallback (tFNSLQSWDSEvent pCallback)
- ULONG iSLQSSetWdsXferStatsSecondInstCallback (tFNSLQSWDSEvent pCallback)
- ULONG SetPowerCallback (tFNPower pCallback)
- ULONG SetActivationStatusCallback (tFNActivationStatus pCallback)
- ULONG SetMobileIPStatusCallback (tFNMobileIPStatus pCallback)
- ULONG SetRoamingIndicatorCallback (tFNRoamingIndicator pCallback)
- ULONG SetDataCapabilitiesCallback (tFNDataCapabilities pCallback)
- ULONG SetSignalStrengthCallback (tFNSignalStrength pCallback, BYTE thresholdsSize, INT8 *pThresholds)
- ULONG iSetSignalStrengthCallback (tFNSignalStrength pCallback)
- ULONG SetRFInfoCallback (tFNRFInfo pCallback)
- ULONG SetLURejectCallback (tFNLUReject pCallback)
- ULONG SetNewSMSCallback (tFNNewSMS pCallback)
- ULONG SLQSSetSMSEventCallback (tFNSMSEvents pCallback)
- ULONG SetNMEACallback (tFNNewNMEA pCallback)
- ULONG SetPDSStateCallback (tFNPDSState pCallback)
- ULONG SetCATEventCallback (tFNCATEvent pCallback, ULONG eventMask, ULONG *pErrorMask)
- ULONG iSetCATEventCallback (tFNCATEvent pCallback)
- ULONG SetDeviceStateChangeCbk (tFNDeviceStateChange pCallback)

- ULONG SetNetChangeCbk (BYTE instance, tFNNet pCallback, ULONG loMark, ULONG hiMark, ULONG period)
- ULONG SetFwDldCompletionCbk (tFNFwDldCompletion pCallback)
- ULONG SetSLQSOMADMAlertCallback (tFNSLQSOMADMAlert pCallback)
- ULONG SetSLQSOMADMAlertCallbackExt (tFNSLQSOMADMAlert pCallback)
- ULONG SetOMADMStateCallback (tFNOMADMState pCallback)
- ULONG SLQSSetServingSystemCallback (tFNServingSystem pCallback)
- ULONG SLQSSetBandPreferenceCbk (tFNBandPreference pCallback)
- ULONG SetUSSDReleaseCallback (tFNUSSDRelease pCallback)
- ULONG SetUSSDNotificationCallback (tFNUSSDNotification pCallback)
- ULONG SLQSSetSignalStrengthsCallback (tFNSLQSSignalStrengths pCallback, struct SLQSSignalStrengthsIndReq *pSLQSSignalStrengthsIndReq)
- ULONG iSLQSSetSignalStrengthsCallback (tFNSLQSSignalStrengths pCallback)
- ULONG SLQSVoiceSetSUPSNotificationCallback (tFNSUPSNotification pCallback)
- ULONG SLQSSetSDKTerminatedCallback (tFNSDKTerminated pCallback)
- ULONG SLQSVoiceSetAllCallStatusCallBack (tFNAllCallStatus pCallback)
- ULONG SLQSSetTransLayerInfoCallback (tFNtransLayerInfo pCallback)
- ULONG SLQSSetTransNWRegInfoCallback (tFNtransNWRegInfo pCallback)
- ULONG SLQSSetSysSelectionPrefCallBack (tFNSysSelectionPref pCallback)
- ULONG SLQSUIMSetRefreshCallBack (tFNUIMRefresh pCallback)
- ULONG SLQSUIMSetStatusChangeCallBack (tFNUIMStatusChangeInfo pCallback)
- ULONG SLQSVoiceSetPrivacyChangeCallBack (tFNPrivacyChange pCallback)
- ULONG SLQSVoiceSetDTMFEventCallBack (tFNDTMFEvent pCallback)
- ULONG SLQSVoiceSetSUPSCallBack (tFNSUPSInfo pCallback)
- ULONG SLQSNasSysInfoCallBack (tFNSysInfo pCallback)
- ULONG SLQSNasNetworkTimeCallBack (tFNNetworkTime pCallback)
- ULONG SLQSWmsMemoryFullCallBack (tFNMemoryFull pCallback)
- ULONG SLQSVoiceSetOTASPStatusCallBack (tFNOTASPStatus pCallback)
- ULONG SLQSVoiceInfoRecCallback (tFNInfoRec pCallback)
- ULONG SLQSWmsMessageWaitingCallBack (tFNMessageWaiting pCallback)
- ULONG SLQSSetQosEventCallback (BYTE instance, tFNSLQSQOSEvent pCallback)
- ULONG SLQSSetQosStatusCallback (BYTE instance, tFNQosStatus pCallback)
- ULONG SLQSSetQosNWStatusCallback (tFNQosNWStatus pCallback)
- ULONG SLQSSetQosPriEventCallback (tFNQosPriEvent pCallback)
- ULONG SLQSNasSigInfoCallBack (tFNSigInfo pCallback, sigInfo *pSigInfo)
- ULONG SLQSSetModemTempCallback (tFNModemTempInfo pCallback)
- ULONG SLQSSetPacketSrvStatusCallback (tFNPacketSrvState pCallback)
- ULONG SLQSSetSwiHDRPersCallback (tFNHDRPersonaity pCallback)
- ULONG SLQSSetSIPConfigCallback (tFNImsSIPConfig pCallback)
- ULONG SLQSSetRegMgrConfigCallback (tFNImsRegMgrConfig pCallback)
- ULONG SLQSSetIMSSMSConfigCallback (tFNImsSMSConfig pCallback)
- ULONG SLQSSetIMSUserConfigCallback (tFNImsUserConfig pCallback)
- ULONG SLQSSetIMSVoIPConfigCallback (tFNImsVoIPConfig pCallback)
- ULONG SetUSSDNoWaitIndicationCallback (tFNUSSDNoWaitIndication pCallback)
- ULONG SLQSSetDUNCallInfoCallback (BYTE StatsPeriod, tFNDUNCallInfo pCallback)
- ULONG iSLQSSetDUNCallInfoCallback (tFNDUNCallInfo pCallback)
- ULONG SLQSSetDataSystemStatusCallback (tFNDataSysStatus pCallback)
- ULONG SLQSWmsAsyncRawSendCallBack (tFNAsyncRawSend pCallback)
- ULONG SLQSNasSwiOTAMessageCallback (NasSwiIndReg *req, tFNASwiOTAMsg pCallback)
- ULONG SetGPSCallback (tFNNewGPS pCallback)
- ULONG SetRMTransferStatisticsCallback (tFNNewRMTransferStatistics pCallback)
- ULONG SLQSSetDHCPv4ClientLeaseStatusCallback (BYTE instance, tFNDHCPv4ClientLeaseStatus pCallback)
- ULONG SetLocCradleMountCallback (tFNSetCradleMount pCallback)
- ULONG SetLocEventTimeSyncCallback (tFNSetEventTimeSync pCallback)

- ULONG SetLocInjectTimeCallback (tFNInjectTimeStatus pCallback)
- ULONG SetLocSensorStreamingCallback (tFNSensorStreaming pCallback)
- ULONG SetLocInjectSensorDataCallback (tFNInjectSensorData pCallback)
- ULONG SetLocEventPositionCallback (tFNEventPosition pCallback)
- ULONG SetLocOpModeCallback (tFNOpMode pCallback)
- ULONG SLQSSetIMSARegStatusCallback (tFNImsaRegStatus pCallback)
- ULONG SLQSSetIMSASvcStatusCallback (tFNImsaSvcStatus pCallback)
- ULONG SLQSSetIMSARatStatusCallback (tFNImsaRatStatus pCallback)
- ULONG SLQSSetIMSAPdpStatusCallback (tFNImsaPdpStatus pCallback)
- ULONG SLQSNasSigInfo2CallBack (tFNSigInfo pCallback, setSignalStrengthInfo *pSigInfo2)
- ULONG SetLocGnssSvInfoCallback (tFNGnssSvInfo pCallback)
- ULONG SetLocDeleteAssistDataCallback (tFNDelAssistData pCallback)
- ULONG SetNasLTECphyCaIndCallback (tFNASwiLTECphyCaInfo pCallback)
- ULONG SetRankIndicatorCallback (tFNRankIndicator pCallback)
- ULONG SLQSSetLocInjectUTCTimeCallback (tFNInjectUTCTime pCallback)
- ULONG SLQSSetLocInjectPositionCallback (tFNInjectPosition pCallback)
- ULONG SetUimSlotStatusChangeCallback (tFNCbkUimSlotStatusChangeInd pCallback)
- ULONG SetLocEngineStateCallback (tFNSetEngineState pCallback)
- ULONG SLQSSetSwiGetResetInfoCallback (tFNResetInfo pCallback)
- ULONG SLQSTmdMitigationLvlRptCallback (TmdMitigationLvlIndReq *req, tFNMitiLvlRpt pCallback)
- ULONG SetLocSetExtPowerConfigCallback (tFNSetExtPowerConfig pCallback)
- ULONG SetLocBestAvailPosCallback (tFNBestAvailPos pCallback)
- ULONG SLQSNasTimerCallback (tFNNasTimer pCallback)
- ULONG SLQSNasNetworkRejectCallback (tFNNetworkReject pCallback)
- ULONG SLQSSwiAVMSEventReportCallback (tFNSwiAvmEventReport pCallback)
- ULONG SetLocEventMaskNMEACallback (tFNEventNmea pCallback)
- ULONG SetLocGetServerCallback (LocGetServerReq *req, tFNGetServer pCallback)

9.12.1 Detailed Description

Callback Service API function prototypes.

9.12.2 Macro Definition Documentation

9.12.2.1 #define CBK_DISABLE_EVENT 0x00

9.12.2.2 #define CBK_ENABLE_EVENT 0x01

9.12.2.3 #define CBK_NOCHANGE 0xFF

9.12.2.4 #define DEREGISTER_EVENT 0x00

9.12.2.5 #define DEREGISTER_SRV 0x00

9.12.2.6 #define DHCP_MAX_NUM_OPTIONS 30

9.12.2.7 #define DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */

9.12.2.8 #define EVENT_MASK_CARD 0x00000001

9.12.2.9 #define EVENT_MASK_DEREGISTER_ALL 0x00000000

9.12.2.10 #define EVENT_MASK_PHY_SLOT_STATUS 0x00000010

9.12.2.11 #define FIRST_INSTANCE 0x00

9.12.2.12 #define INVALID_INSTACNE 0x08

9.12.2.13 #define IPV4 4

9.12.2.14 #define IPV4V6 7

9.12.2.15 #define IPV6 6

9.12.2.16 #define LOC_EVENT_MASK_ENG_STATE 0x00000080

9.12.2.17 #define LOC_EVENT_MASK_GNSS_SV_INFO 0x00000002

9.12.2.18 #define LOC_EVENT_MASK_INJECT_TIME 0x00000010

9.12.2.19 #define LOC_EVENT_MASK_NMEA 0x00000004

9.12.2.20 #define LOC_EVENT_MASK_SENSOR_STREAM 0x00000400

9.12.2.21 #define LOC_EVENT_MASK_TIME_SYNC 0x00000800

9.12.2.22 #define LOC_EVENT_POSITION_REPORT 0x00000001

9.12.2.23 #define MAX_MITIGATION_DEV_ID_LEN 255

9.12.2.24 #define MAX_NMEA_STR_LEN 201

9.12.2.25 #define MAX_NO_OF_APPLICATIONS 10

9.12.2.26 #define MAX_NO_OF_CALLS 20

9.12.2.27 #define MAX_NO_OF_FILES 255

9.12.2.28 #define MAX_NO_OF_SLOTS 5

9.12.2.29 #define MAX_NO_OF_UUSINFO 20

9.12.2.30 #define MAX_PATH_LENGTH 255

9.12.2.31 #define MAX_RADIO_INTERFACE_LIST 255

9.12.2.32 #define MAX_URL_ADDR_STR_LEN 256

9.12.2.33 #define MAXUSSDLENGTH 182

9.12.2.34 #define NAS_SRV 0x02

9.12.2.35 #define NUM_OF_SET 0xFF

9.12.2.36 #define PDS_SRV 0x04

9.12.2.37 #define QMI_ETWS_MAX_PAYLOAD_LENGTH 1254 /* Qualcomm defined max */

9.12.2.38 #define QMI_MAX_VOICE_NUMBER_LENGTH 81

9.12.2.39 #define QMI_WMS_MAX_PAYLOAD_LENGTH 256

9.12.2.40 #define REGISTER_EVENT 0x01

9.12.2.41 #define REGISTER_SRV 0x01

9.12.2.42 #define SECOND_INSTANCE 0x01

9.12.2.43 #define SIGSTRENGTH_THRESHOLD_ARR_SZ 5

9.12.2.44 #define THIRD_INSTANCE 0x02

9.12.2.45 #define USSD_DCS_8BIT 0x02 /∗ 8-bit coding scheme ∗/

9.12.2.46 #define USSD_DCS_ASCII 0x01 /∗ ASCII coding scheme ∗/

9.12.2.47 #define USSD_DCS_UCS2 0x03 /∗ UCS2 coding scheme ∗/

9.12.2.48 #define VOICE_SRV 0x08

9.12.2.49 #define WDS_SRV 0x01

9.12.3 Typedef Documentation

9.12.3.1 typedef struct accelAcceptReady_s accelAcceptReady

This structure contains Accelerometer Accept Ready Info

Parameters

injectEnable	• GNSS location engine is ready to accept data from sensor. • 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

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.

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

mobileCountryCode	• 16 bit representation of MCC value range : 0 -999
mobileNetworkCode	• 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
pMTMessageInfo	• pointer to the SMSMTMessageInfo structure NULL, if this event is not present in the smsEventType parameter
pTransferRouteMTMessageInfo	• pointer to the SMSTransferRouteMTMessageInfo structure . NULL, if this event is not present in the smsEventType parameter
pMessageModeInfo	• pointer to the SMSMessageModeInfo structure NULL, if this event is not present in the smsEventType parameter
pEtwsMessageInfo	• 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
pSMSCAddressInfo	• 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(∗ tFNEventNmea)(CHAR ∗pNMEAData)

NMEA Event Notification callback.

Parameters

pNMEAData	• NMEA string • Type - NULLterminated string • Maximum string length (including NULL terminator) - 201

9.12.3.45 typedef void(∗ tFNEventPosition)(QmiCbkLocPositionReportInd ∗pLocPositionReport)

Position report callback function.

Parameters

pLocPosition-Report[OUT]	• Events related to LOC, see QmiCbkLocPositionReportInd for details

9.12.3.46 typedef void(∗ tFNFwDldCompletion)(ULONG fwdld_completion_status)

Firmware Download Completion callback function prototype

Parameters

error_code	• error code returned from firmware download operation, the possible return values are listed below: – 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

9.12.3.47 typedef void(∗ tFNGetServer)(QmiCbkLocGetServerInd ∗pGetLocServerInfo)

Get LOC Server Indication callback.

Parameters

pGetLocServer-Info	• See QmiCbkLocGetServerInd for more information.

9.12.3.48 typedef void(∗ tFNGnssSvInfo)(gnssSvInfoNotification ∗pGnssSvInfoNotification)

GNSS SVN Information Notification callback.

Parameters

pGnssSvInfo-Notification	• See gnssSvInfoNotification for more information.

9.12.3.49 typedef void(∗ tFNHDRPersonaity)(HDRPersonalityInd ∗pHDRPers)

HDR Personality indication callback.

Parameters

pHDRPers	• See HDRPersonalityInd for more information.

Note

Technology Supported: CDMA

9.12.3.50 typedef void(∗ tFNImsaPdpStatus)(imsaPdpStatusInfo ∗pImsaPdpStatusInfo)

IMSA PDP status indication callback.

Parameters

pImsaPdp-StatusInfo	• See imsaPdpStatusInfo for more information.

9.12.3.51 typedef void(∗ tFNImsaRatStatus)(imsaRatStatusInfo ∗pImsaRatStatusInfo)

IMSA RAT handover status indication callback.

Parameters

pImsaRatStatus-Info	• See imsaRatStatusInfo for more information.

9.12.3.52 typedef void(∗ tFNImsaRegStatus)(imsaRegStatusInfo ∗pImsaRegStatusInfo)

IMSA Registration Status indication callback.

Parameters

pImsaReg-StatusInfo	• See imsaRegStatusInfo for more information.

9.12.3.53 typedef void(∗ tFNImsaSvcStatus)(imsaSvcStatusInfo ∗pImsaSvcStatusInfo)

IMSA Service Status indication callback.

Parameters

pImsaSvcStatus-Info	• See imsaSvcStatusInfo for more information.

9.12.3.54 typedef void(* tFNImsRegMgrConfig)(imsRegMgrConfigInfo *pImsRegMgrConfigInfo)

IMS Reg Mgr Config indication callback.

Parameters

pImsRegMgrConfigInfo	• See imsRegMgrConfigInfo for more information.

9.12.3.55 typedef void(* tFNImsSIPConfig)(imsSIPConfigInfo *pImsSIPConfigInfo)

IMS SIP Config indication callback.

Parameters

pImsSIPConfigInfo	• See imsSIPConfigInfo for more information.

9.12.3.56 typedef void(* tFNImsSMSConfig)(imsSMSConfigInfo *pImsSMSConfigInfo)

IMS SMS Config indication callback.

Parameters

pImsSMSConfigInfo	• See imsSMSConfigInfo for more information.

9.12.3.57 typedef void(* tFNImsUserConfig)(imsUserConfigInfo *pImsUserConfigInfo)

IMS User Config indication callback.

Parameters

pImsUserConfigInfo	• See imsUserConfigInfo for more information.

9.12.3.58 typedef void(* tFNImsVoIPConfig)(imsVoIPConfigInfo *pImsVoIPConfigInfo)

IMS VoIP Config indication callback.

Parameters

pImsVoIPConfigInfo	• See imsVoIPConfigInfo for more information.

9.12.3.59 typedef void(∗ tFNInfoRec)(voiceInfoRec ∗pVoiceInfoRec)

Voice Information Record callback.

Parameters

pVoiceInfoRec	• See voiceInfoRec for more information.

Note

Technology Supported: CDMA
Device Supported: MC7750

9.12.3.60 typedef void(∗ tFNInjectPosition)(QmiCbkLocInjectPositionInd ∗pInjectPositionNotification)

Inject Position Notification callback.

Parameters

pInjectPosition-Notification	• See QmiCbkLocInjectPositionInd for more information.

9.12.3.61 typedef void(∗ tFNInjectSensorData)(QmiCbkLocInjectSensorDataInd ∗pLocInjectSensorData)

LOC Inject Sensor Data indication callback.

Parameters

pLocInject-SensorData	• See QmiCbkLocInjectSensorDataInd for more information.

9.12.3.62 typedef void(∗ tFNInjectTimeStatus)(QmiCbkLocInjectTimeInd ∗pLocInjectTime)

LOC Inject time status indication callback.

Parameters

pLocInjectTime	• See QmiCbkLocInjectTimeInd for more information.

9.12.3.63 typedef void(∗ tFNInjectUTCTime)(QmiCbkLocInjectUTCTimeInd ∗pInjectUTCTimeNotification)

Inject UTC Time Notification callback.

Parameters

pInjectUTCTime-Notification	• See QmiCbkLocInjectUTCTimeInd for more information.

9.12.3.64 typedef void(* tFNLUReject)(ULONG serviceDomain, ULONG rejectCause)

LU reject callback function.

Parameters

serviceDomain	• Service domain – 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

9.12.3.65 typedef void(∗ tFNMemoryFull)(SMSMemoryInfo ∗pSMSMemoryFullInfo)

SMS Memory related callback function.

Parameters

pSMSMemoryFullInfo[OUT]	• pointer to SMSMemoryInfo. • see SMSMemoryInfo for details.

9.12.3.66 typedef void(∗ tFNMessageWaiting)(msgWaitingInfo ∗pSMSMessageWaitingInfo)

SMS Memory related callback function.

Parameters

pSMSMessageWaitingInfo[OUT]	• pointer to msgWaitingInfo. • see msgWaitingInfo for details.

9.12.3.67 typedef void(∗ tFNMitiLvlRpt)(QmiCbkTmdMitiLvlRptInd ∗pSetLocCradleMount)

Set Loc cradle mount Indication callback.

Parameters

pSetLocCradleMount	• See QmiCbkTmdMitiLvlRptInd for more information.

9.12.3.68 typedef void(∗ tFNMobileIPStatus)(ULONG mipStatus)

Mobile IP status callback function.

Parameters

mipStatus	• Mobile IP Status – 0 - success – All others error codes as defined in RFC 2002 See qaGobiApiTableCallEndReasons.h for mobile IP error codes

9.12.3.69 typedef void(∗ tFNModemTempInfo)(modemTempNotification ∗pModemTempNotification)

Modem Temperature Information callback.

Parameters

pModemTemp-Notification	• See modemTempNotification for more information.

9.12.3.70 typedef void(∗ tFNNasTimer)(nasTimers ∗timers)

Network timer Indication callback.

Parameters

timers	• See nasTimers for more information.

9.12.3.71 typedef void(∗ tFNNet)(ULONG q_depth, BYTE isThrottle, BYTE instanceId)

Transmit Queue Length Change callback function prototype

Parameters

q_depth	• transmit queue length
isThrottle	• 0: unthrottle • 1: throttle
instanceId	• qmi instance id

Note

Does not require communication with the device

9.12.3.72 typedef void(∗ tFNNetworkReject)(nasNetworkReject ∗pNetworkReject)

Network Reject Cause callback.

Parameters

pNasNetwork-Time	• See nasNetworkTime for more information.

9.12.3.73 typedef void(∗ tFNNetworkTime)(nasNetworkTime ∗pNasNetworkTime)

Network Time indication callback.

Parameters

pNasNetwork-Time	• See nasNetworkTime for more information.

9.12.3.74 typedef void(∗ tFNNewGPS)(double dLongitude, double dLatitude, BYTE session_status, ULONG pos_src)

Set Current Location Data.

Parameters

dLongitude[IN]	• Current Longitude Value
dLatitude[IN]	• Current Latitude Value
session_status[I-N]	• Session Status – 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

9.12.3.75 typedef void(∗ tFNNewNMEA)(LPCSTR pNMEA)

New NMEA sentence callback function.

Parameters

pNMEA	• NULL-terminated string containing the position data in NMEA sentence format

9.12.3.76 typedef void(∗ tFNNewRMTransferStatistics)(QmiCbkWdsStatisticsIndState ∗pMsg)

PDS session state callback function.

Parameters

enabledStatus	• GPS enabled status – 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

9.12.3.77 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

9.12.3.78 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

9.12.3.79 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

9.12.3.80 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

9.12.3.81 typedef void(∗ tFNPacketSrvState)(packetSrvStatus ∗pPacketSrvStatus)

Packet Service state callback function.

Parameters

pPacketSrv-Status	• See packetSrvStatus for more details

9.12.3.82 typedef void(∗ tFNPDSState)(ULONG enabledStatus, ULONG trackingStatus)

PDS session state callback function.

Parameters

enabledStatus	• GPS enabled status – 0 - Disable – 1 - Enable
trackingStatus	• GPS tracking status – 0 - Unknown – 1 - Inactive – 2 - Active

9.12.3.83 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

9.12.3.84 typedef void(∗ tFNPrivacyChange)(voicePrivacyInfo ∗pVoicePrivacyInfo)

Preferred voice privacy indication callback.

Parameters

pVoicePrivacy-Info	• See voicePrivacyInfo for more information.

Note

Technology Supported: CDMA

9.12.3.85 typedef void(∗ 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

9.12.3.86 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

9.12.3.87 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

9.12.3.88 typedef void(∗ tFNRankIndicator)(RankIndicatorInd ∗pRankIndicatorInd)

Rank Indicator callback function.

Parameters

pRankIndicator-Ind[OUT]	• Events related to NAS, see RankIndicatorInd for details.

9.12.3.89 typedef void(∗ tFNResetInfo)(ResetInfoNotification ∗pResetInfoNotification)

Get Reset Info Indication callback.

Parameters

pResetInfo-Notification	• See ResetInfoNotification for more information.

9.12.3.90 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

9.12.3.91 typedef void(∗ tFNRoamingIndicator)(ULONG roaming)

Roaming indicator callback function.

Parameters

roaming	• Roaming Indication – 0 - Roaming – 1 - Home – 2 - Roaming partner – >2 - Operator defined values

9.12.3.92 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

9.12.3.93 typedef void(∗ tFNSensorStreaming)(QmiCbkLocSensorStreamingInd ∗pLocSensorStream)

LOC Sensor Streaming indication callback.

Parameters

pLocSensor-Stream	• See QmiCbkLocSensorStreamingInd for more information.

9.12.3.94 typedef void(∗ tFNServingSystem)(struct ServingSystemInfo ∗pServingSystem,struct RoamingInfo ∗pRoamingInfo)

Serving System callback function

Parameters

pServingSystem	• ServingSystemInfo structure

9.12.3.95 typedef void(∗ tFNSetCradleMount)(QmiCbkLocCradleMountInd ∗pSetLocCradleMount)

Set Cradle Mount indication callback.

Parameters

pSetLocCradle-Mount	• See QmiCbkLocCradleMountInd for more information.

9.12.3.96 typedef void(∗ tFNSetEngineState)(QmiCbkLocEngineStateInd ∗pSetLocEngineState)

Set Engine State Indication callback.

Parameters

pSetLocEngine-State	• See QmiCbkLocEngineStateInd for more information.

9.12.3.97 typedef void(∗ tFNSetEventTimeSync)(QmiCbkLocEventTimeSyncInd ∗pSetLocEventTimeSync)

LOC event time sync indication callback.

Parameters

pSetLocEvent-TimeSync	• See QmiCbkLocEventTimeSyncInd for more information.

9.12.3.98 typedef void(∗ tFNSetExtPowerConfig)(QmiCbkLocSetExtPowerConfigInd ∗pSetExtConfigIndStatus)

Set Ext Power Config Indication callback.

Parameters

pSetExtConfig-IndStatus	• See QmiCbkLocSetExtPowerConfigInd for more information.

9.12.3.99 typedef void(∗ tFNSigInfo)(nasSigInfo ∗pNasSigInfo)

Signal Strength Information indication callback.

Parameters

pNasSigInfo	• See nasSigInfo for more information.

9.12.3.100 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

9.12.3.101 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

9.12.3.102 typedef void(∗ tFNSLQSQOSEvent)(BYTE instance, QosFlowInfo ∗pFlowInfo)

QOS Event callback function.

Parameters

instance	• QMI instance
pFlowInfo	• See QosFlowInfo for more information

9.12.3.103 typedef void(* tFNSLQSSessionState)(slqsSessionStateInfo *pSessionStateInfo)

Session state callback function.

Parameters

pSessionStateInfo	• See slqsSessionStateInfo for more details

9.12.3.104 typedef void(* tFNSLQSSignalStrengths)(struct SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo)

Received Signal Strength Information callback function.

Parameters

sSLQSSignalStrengthsInfo	• See SLQSSignalStrengthsInformation for more information.

9.12.3.105 typedef void(* tFNSLQSWDSEvent)(slqsWdsEventInfo *pWdsEventInfo)

WDS Event callback function.

Parameters

pWdsEventInfo	• See slqsWdsEventInfo for more details

9.12.3.106 typedef void(* tFNSMSEvents)(SMSEventInfo *pSMSEventInfo)

SMS event related callback function.

Parameters

pSMSEventInfo[-OUT]	• Events related to SMS, see SMSEventInfo for details

9.12.3.107 typedef void(* tFNSUPSInfo)(voiceSUPSInfo *pVoiceSUPSInfo)

Preferred SUPS indication callback.

Parameters

pVoiceSUPSInfo	• See voiceSUPSInfo for more information.

Note

Technology Supported: GSM

9.12.3.108 typedef void(∗ tFNSUPSNotification)(**voiceSUPSNotification** ∗pVoiceSUPSNotification)

Supplementary service notification callback.

Parameters

pVoiceSUPS-Notification	• See voiceSUPSNotification for more information.

9.12.3.109 typedef void(∗ tFNSwiAvmEventReport)(**SwiAvmsEventInd** ∗pEvent)

AVMS Event Report indication callback.

Parameters

pEvent	• See SwiAvmsEventInd for more information.

9.12.3.110 typedef void(∗ tFNSysInfo)(**nasSysInfo** ∗pNasSysInfo)

System Information indication callback.

Parameters

pNasSysInfo	• See nasSysInfo for more information.

9.12.3.111 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

9.12.3.112 typedef void(∗ tFNtransLayerInfo)(**transLayerNotification** ∗pTransLayerNotification)

Transport Layer Information callback.

Parameters

transLayer-Notification	• See transLayerNotification for more information.

9.12.3.113 typedef void(∗ tFNtransNWRegInfo)(transNWRegInfoNotification ∗pTransNWRegInfoNotification)

Transport Network Registration Information callback.

Parameters

pTransNWReg-InfoNotification	• See transNWRegInfoNotification for more information.

9.12.3.114 typedef void(∗ tFNUIMRefresh)(UIMRefreshEvent ∗pUIMRefreshEvent)

UIM Refresh Callback function

Parameters

pUIMRefresh-Event	• Pointer to Refresh Event structure. • See UIMRefreshEvent for more information

9.12.3.115 typedef void(∗ tFNUIMStatusChangeInfo)(UIMStatusChangeInfo ∗pUIMStatusChangeInfo)

UIM Status Change Callback function

Parameters

pUIMStatus-ChangeInfo	• Pointer to UIM status change structure. • See UIMStatusChangeInfo for more information

9.12.3.116 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

9.12.3.117 typedef void(* tFNUSSDNoWaitIndication)(USSDNoWaitIndicationInfo *pNetworkInfo)

USSD No Wait indication callback.

Parameters

pNetworkInfo	• See USSDNoWaitIndicationInfo for more information.

9.12.3.118 typedef void(* tFNUSSDRelease)(void)

USSD releasecallback function prototype

Note

Technology Supported: UMTS

9.12.3.119 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

9.12.3.120 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 SetLocEventMaskNMEACallback (tFNEventNmea *pCallback*)

Enables/disables NMEA Event callback function

Parameters

pCallback[IN]	• Callback function pointer (0 - Disable) • See tFNEventNmea for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

9.12.5.21 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

9.12.5.22 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

9.12.5.23 ULONG SetLocGetServerCallback (LocGetServerReq * *req*, tFNGetServer *pCallback*)

Enables/disables Get location server callback function

Parameters

req[IN]	• See LocGetServerReq for more information.
pCallback[IN]	• Callback function pointer (0 - Disable) • See tFNGetServer for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

9.12.5.24 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

9.12.5.25 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

9.12.5.26 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

9.12.5.27 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)

9.12.5.28 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

9.12.5.29 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

9.12.5.30 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

9.12.5.31 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

9.12.5.32 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

9.12.5.33 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

9.12.5.34 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

9.12.5.35 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

9.12.5.36 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

9.12.5.37 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

9.12.5.38 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

9.12.5.39 ULONG SetRankIndicatorCallback (tFNRankIndicator *pCallback*)

9.12.5.40 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

9.12.5.41 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

9.12.5.42 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

9.12.5.43 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.

9.12.5.44 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

9.12.5.45 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

9.12.5.46 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

9.12.5.47 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

9.12.5.48 ULONG SetUSSDNoWaitIndicationCallback (tFNUSSDNoWaitIndication *pCallback*)

SetUSSDNoWaitIndicationCallback

Parameters

pNetworkInfo	• Data from the network. • See USSDNoWaitIndicationInfo for more details.

Note

Technology Supported: UMTS
Device Supported: MC83x5

9.12.5.49 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

9.12.5.50 ULONG SLQSNasNetworkRejectCallback (tFNNetworkReject *pCallback*)

9.12.5.51 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.

9.12.5.52 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

9.12.5.53 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

9.12.5.54 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

9.12.5.55 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.

9.12.5.56 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

9.12.5.57 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

9.12.5.58 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

9.12.5.59 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

9.12.5.60 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

9.12.5.61 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

9.12.5.62 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

9.12.5.63 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

9.12.5.64 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

9.12.5.65 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

9.12.5.66 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

9.12.5.67 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

9.12.5.68 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)

9.12.5.69 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)

9.12.5.70 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

9.12.5.71 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

9.12.5.72 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)

9.12.5.73 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

9.12.5.74 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

9.12.5.75 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

9.12.5.76 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

9.12.5.77 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
```

9.12.5.78 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

9.12.5.79 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

9.12.5.80 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.

9.12.5.81 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

9.12.5.82 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

9.12.5.83 ULONG SLQSSetSwiGetResetInfoCallback (tFNResetInfo *pCallback*)

Reset Info callback.

Parameters

pCallback	• See tFNResetInfo for more information.

9.12.5.84 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

9.12.5.85 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

9.12.5.86 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

9.12.5.87 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

9.12.5.88 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.

9.12.5.89 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.

9.12.5.90 ULONG SLQSSwiAVMSEventReportCallback (tFNSwiAvmEventReport *pCallback*)

9.12.5.91 ULONG SLQSTmdMitigationLvlRptCallback (TmdMitigationLvlIndReq * *req,* tFNMitiLvlRpt *pCallback*)

Thermal Mitigation callback.

Parameters

req	• See TmdMitigationLvlIndReq for more information.
pCallback	• See tFNMitiLvlRpt for more information.

9.12.5.92 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.

9.12.5.93 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

9.12.5.94 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

9.12.5.95 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

9.12.5.96 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.

9.12.5.97 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.

9.12.5.98 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.

9.12.5.99 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.

9.12.5.100 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

9.12.5.101 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

9.12.5.102 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

9.12.5.103 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 QCWWAN2kEnumerateDevices() 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/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

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[OUT]	• Pointer to SLQS Device Mode of type eDevState

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.13.3.9 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 – 1 - Activated – 2 - Blocked
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[O-UT]	-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
- struct LocGetServerReq

Macros

- #define MAX_SENSOR_DATA_LEN 64
- #define MAX_TEMP_DATA_LEN 64

Functions

- ULONG SLQSLOCEventRegister (LOCEventRegisterReqResp ∗pLOCEventRegisterReqResp)
- ULONG SLQSLOCSetExtPowerState (LOCExtPowerStateReqResp ∗pLOCExtPowerStateReqResp)
- ULONG SLQSLOCStart (LOCStartReq ∗pLOCStartReq)
- ULONG SLQSLOCStop (LOCStopReq ∗pLOCStopReq)
- ULONG SLQSLOCSetOpMode (ULONG mode)
- ULONG SLQSLOCDelAssData (LocDelAssDataReq request)
- ULONG SwiLocGetAutoStart (SwiLocGetAutoStartResp ∗resp)
- ULONG SwiLocSetAutoStart (SwiLocSetAutoStartReq ∗req)
- ULONG SLQSLOCInjectUTCTime (ULONGLONG timeMsec, ULONG timeUncMsec)
- ULONG SLQSLOCInjectPosition (LocInjectPositionReq ∗pLocInjectPositionReq)
- ULONG SLQSLOCSetCradleMountConfig (LocSetCradleMountReq ∗pLocSetCradleMountReq)
- ULONG SLQSLOCInjectSensorData (LocInjectSensorDataReq ∗pLocInjectSensorDataReq)
- ULONG SLQSLOCGetBestAvailPos (ULONG xid)
- ULONG SLQSLOCGetServer (LocGetServerReq ∗pLocGetServerReq)

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 SLQSLOCGetServer (LocGetServerReq * *pLocGetServerReq*)

Gets the A-GPS server address from the engine. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through SetLocGetServerCallback callback If successful, the callback also contains the A-GPS server address

Parameters

pLocGetServerReq[IN]	• See LocGetServerReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.5 ULONG SLQSLOCInjectPosition (LocInjectPositionReq * *pLocInjectPositionReq*)

Injects a position to the location engine.

Parameters

pLocInjectPositionReq[IN]	• See LocInjectPositionReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.6 ULONG SLQSLOCInjectSensorData (LocInjectSensorDataReq * *pLocInjectSensorDataReq*)

Control point to to inject sensor data into the GNSS location engine.

Parameters

pLocInject-SensorData-Req[IN]	• See LocInjectSensorDataReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.7 ULONG SLQSLOCInjectUTCTime (ULONGLONG *timeMsec,* ULONG *timeUncMsec*)

Injects UTC time in the location engine.

Parameters

timeMsec[IN]	• The UTC time since Jan. 1, 1970
timeUncMsec[I-N]	• The time Uncertainty

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.8 ULONG SLQSLOCSetCradleMountConfig (LocSetCradleMountReq * *pLocSetCradleMountReq*)

Control point to set the current cradle mount configuration.

Parameters

pLocSetCradle-MountReq[IN]	• See LocSetCradleMountReq for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.9 ULONG SLQSLOCSetExtPowerState (LOCExtPowerStateReqResp * *pLOCExtPowerStateReqResp*)

Used by the control point to set the current external power configuration.

Parameters

pLOCExtPower-StateReqResp[I-N]	• See LOCExtPowerStateReqResp for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 5 seconds

9.18.3.10 ULONG SLQSLOCSetOpMode (ULONG *mode*)

Used by the control point to tells the engine to use the specified operation mode while making the position fixes

Parameters

mode[IN]	• Valid values: – 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

9.18.3.11 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

9.18.3.12 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

9.18.3.13 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

9.18.3.14 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
pPCSDigit-Instances[IN/O-UT]	• 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[O-UT]	• 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[IN]	• 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[-OUT]	• See QmiSwiAvmsEventReportConfigResp for more information.
pNotif[OUT]	• See QmiSwiAvmsEventReportNotificationResp for more information.
pPackageID[OU-T]	• See QmiSwiAvmsEventReportPackageIDResp for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.28.4.2 ULONG SLQSAVMSGetSettings (SLQSAVMSSettings * *settings*)

9.28.4.3 ULONG SLQSAVMSSelection (BYTE *selection*, ULONG * *pDeferTime*, BYTE * *pClientPerfOperationFlag*, BYTE * *pPackageID*, WORD * *pRejectReason*)

Sends the specified AVMS selection for the current network initiated session.

Parameters

selection[IN]	• AVMS User Selection. – 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
- 227 - INTERNAL_PDN_INACTIVITY_TIMER_EXPIRED, this error code is returned when PDN inactivity timer expired due to no data transmission in a configurable duration of time
- 228 - MAX_V4_CONNECTIONS, this error code is returned when IPv4 data call bring up is rejected because the UE already maintains the allotted maximum number of IPv4 data connections
- 229 - MAX_V6_CONNECTIONS, this error code is returned when IPv6 data call bring up is rejected because the UE already maintains the allotted maximum number of IPv6 data connections
- 230 - APN_MISMATCH, this error code is returned when New PDN bring up is rejected during interface selection because the UE has already allotted the available interfaces for other PDNs
- 231 - IP_VERSION_MISMATCH, this error code is returned when New call bring up is rejected because the existing data call IP type does not match the requested IP type
- 232 - DUN_CALL_DISALLOWED, this error code is returned when DUN call bring up is rejected because the UE is in eHRPD RAT
- 233 - INVALID_PROFILE, this error code is returned when Call bring up was requested with an invalid profile
- 234 - INTERNAL_EPC_NONEPC_TRANSITION, this error code is returned when Data call is rejected or brought down because the UE is in transition between EPC and non-EPC RAT
- 235 - INVALID_PROFILE_ID, this error code is returned when Call bring up was requested with an invalid profile ID
- 236 - INTERNAL_CALL_ALREADY_PRESENT, this error code is returned when a data call with the same policy is already connected or in the process of bring up on another RmNet instance
- 237 - IFACE_IN_USE, this error code is returned when the current IFACE is in use
- 238 - IP_PDP_MISMATCH, this error code is returned when a PPP call is attempted on a PDP profile
- 239 - APN_DISALLOWED_ON_ROAMING, this error code is returned when PDN connection to the APN is disallowed on the roaming network. The DS generates DS_SYS_EVENT_3GPP_ROAMING_DISALLOWED_INFO to inform clients of the APNs that must deregister and disconnect (if the PDN is up) while the UE is on the roaming network. Clients must deregister or disconnect if the APN name corresponding to the client is listed in this system event. Clients can register again if the APN name corresponding to the client is not listed in this system event.

- 240 - APN_PARAM_CHANGE, this error code is returned when Failure to reestablish the PDN with the changed parameters; when APN-related parameters are changed, the PDN associated with the parameters must be brought down and reestablished with the changed parameters
- 241 - IFACE_IN_USE_CFG_MATCH, this error code is returned when the IFACE is already in use with a matching configuration
- 242 - NULL_APN_DISALLOWED, this error code is returned when a PDN is attempted to be brought up with a NULL APN and when a NULL APN is not supported
- 243 - THERMAL_MITIGATION, this error code is returned when the thermal level increases and causes calls to be torn down when normal mode of operation is not allowed
- 244 - SUBS_ID_MISMATCH, this error code is returned when a new call bring up is rejected due to a mismatch between the subs_id in the profile and the subs_id in the ACL policy information
- 245 - DATA_SETTINGS_DISABLED, this error code is returned when the PDN connection to a given APN is disallowed because data is disabled from the device user interface (UI) settings
- 246 - DATA_ROAMING_SETTINGS_DISABLED, this error code is returned when the PDN Connection to a given APN is disallowed because data roaming is disabled from the device UI settings and the UE is roaming
- 247 - APN_FORMAT_INVALID, this error code is returned when the APN specified in the policy does not follow the format specified in the 3GPP Specification
- 248 - DDS_CALL_ABORT, this error code is returned when a DDS switch occurs; the data call is brought down with the DDS abort reason
- 249 - VALIDATION_FAILURE, this error code is returned when a data call is brought down due to an internal validation failure

9.32.2.8 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
- 535 - USER CALL ORIGINATE DURING VOICE CALL, this error code is returned when data call is brought down because traffic channel got released by CM(Call Manager) in favor of voice call or SMS when concurrent voice and data are not supported
- 1000 - CONFERENCE FAILED
- 1001 - INCOMING REJECTED
- 1002 - NO GATEWAY SERVICE
- 1003 - NO GPRS CONTEXT
- 1004 - ILLEGAL MS, This cause is sent to the MS when the network refuses service to the MS either because an identity of the MS is not acceptable to the network or because the MS does not pass the authentication check, i.e. the SRES received from the MS is different from that generated by the network
- 1005 - ILLEGAL ME, This cause is sent to the MS if the ME used is not acceptable to the network, e.g. blacklisted
- 1006 - GPRS SERVICES AND NON GPRS SERVICES NOT ALLOWED
- 1007 - GPRS SERVICES NOT ALLOWED

- 1008 - MS IDENTITY CANNOT BE DERIVED BY THE NETWORK
- 1009 - IMPLICITLY DETACHED, this error code is sent to the MS either if the network has implicitly detached the MS, e.g. some while after the Mobile reachable timer has expired, or if the GMM context data related to the subscription does not exist in the SGSN e.g. because of a SGSN restart.
- 1010 - PLMN NOT ALLOWED, this error code is sent to the MS if it requests location updating in a PLMN where the MS, by subscription or due to operator determined barring is not allowed to operate
- 1011 - LOCAL AREA NOT ALLOWED
- 1012 - GPRS SERVICES NOT ALLOWED IN THIS PLMN
- 1013 - PDP DUPLICATE
- 1014 - USER EQUIPMENT RADIO ACCESS TECHNOLOGY CHANGE
- 1015 - CONGESTION
- 1016 - NO PDP CONEXT ACTIVATED
- 1017 - ACCESS CLASS DSAC REJECTION
- 1018 - PDP ACTIVATE MAX RETRY FAILED
- 1019 - RAB FAILURE
- 1020 - EPS SERVICE NOT ALLOWED
- 1021 - TRACKING AREA NOT ALLOWED
- 1022 - ROAMING NOT ALLOWED IN THIS TRACKING AREA
- 1023 - NO SUITABLE CELLS IN TRACKING AREA
- 1024 - NOT AUTHORIZED FOR THIS CLOSED SUBSCRIBER GROUP
- 1025 - ESM UNKNOWN EPS BEARER CONTEXT
- 1026 - DRB RELEASED AT RRC
- 1027 - NAS SIG CONN RELEASED
- 1028 - EPS MOBILITY MANAGEMENT DETACHED
- 1029 - EPS MOBILITY MANAGEMENT ATTACH FAILED
- 1030 - EPS MOBILITY MANAGEMENT ATTACH STARTED
- 1031 - LTE NAS SERVICE REQ FAILED
- 1032 - ESM(EPS Session Management) ACTIVE DEDICATED BEARER REACTIVATED BY NW
- 1033 - ESM(EPS Session Management) LOWER LAYER FAILURE
- 1034 - ESM(EPS Session Management) SYNC UP WITH NW
- 1035 - ESM(EPS Session Management) NW ACTIVATED DED BEARER WITH ID OF DEF BEARER
- 1036 - ESM(EPS Session Management) BAD OTA MESSAGE
- 1037 - ESM DS REJECTED THE CALL
- 1038 - ESM(EPS Session Management) CONTEXT TRANSFERRED DUE TO IRAT
- 1039 - DS EXPLICIT DEACT
- 1040 - ESM(EPS Session Management) LOCAL CAUSE NONE
- 1041 - LTE NAS SERVICE REQ FAILED NO THROTTLE

- 1042 - ACL FAILURE, This error code should rarely triggered and reported to the application
- 1043 - LTE NAS SERVICE REQ FAILED DS DISALLOW
- 1044 - EMM(EPS Mobility Management) T3417 EXPIRED
- 1045 - EMM(EPS Mobility Management) T3417 EXT EXPIRED
- 1046 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE TXN - Light Radio Resource Controller Uplink data confirmation failure
- 1047 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE HO
- 1048 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CONN REL
- 1049 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE RLF
- 1050 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CTRL NOT CONN
- 1051 - LRRC(LTE Radio Resource Control) CONN EST FAILURE
- 1052 - LRRC(LTE Radio Resource Control) CONN EST FAILURE ABORTED
- 1053 - LRRC(LTE Radio Resource Control) CONN EST FAILURE ACCESS BARRED
- 1054 - LRRC(LTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRRC(LTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRRC(LTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRRC(LTE Radio Resource Control) CONN EST FAILURE LINK FAILURE
- 1058 - LRRC(LTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRRC(LTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRRC(LTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRRC(LTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRRC(LTE Radio Resource Control) CONN REL RLF
- 1063 - LRRC(LTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRC(LTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRC(LTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRC(LTE Radio Resource Control) CONN REL SIB READ ERROR
- 1067 - DETACH WITH REATTACH LTE NW DETACH
- 1068 - DETACH WITH OUT REATTACH LTE NW DETACH
- 1069 - ESM(EPS Session Management) PROC TIME OUT
- 1070 - MESSAGE EXCEED MAX L2 LIMIT
- 1500 - CD GEN OR BUSY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of general or network busy
- 1501 - CD BILL OR AUTH, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of billing failure or authentication failure
- 1502 - CHG HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is a change to HDR system due to redirection or PRL not preferred

- 1503 - EXIT HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device exited HDR due to redirection or PRL not preferred
- 1504 - HDR NO SESSION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have a HDR session
- 1505 - HDR ORIG DURING GPS FIX, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since it is ending an HDR call origination in favor of a GPS fix
- 1506 - HDR CS TIMEOUT, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since connection setup on HDR system timed out
- 1507 - HDR RELEASED BY CM, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when it wants to release a HDR call so a 1X call can continue
- 1508 - COLLOC ACQ FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when device failed to acquire co-located HDR for origination
- 1509 - OTASP COMMIT IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since an OTASP commit is in progress
- 1510 - NO HYBR HDR SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device has no Hybrid HDR service
- 1511 - HDR NO LOCK GRANTED, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module could not get the RF lock
- 1512 - HOLD OTHER IN PROG, this error code is returned when data call is brought down by CM(Call Manager) because DBM or SMS is in progress
- 1513 - HDR FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module released the call due to fade
- 1514 - HDR ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to HDR system Access Failure
- 2000 - CLIENT END, this error code is returned when client ends the data call
- 2001 - NO SRV, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device has no service
- 2002 - FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device lost the system due to fade
- 2003 - REL NORMAL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with no reason
- 2004 - ACC IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access attempt already in progress
- 2005 - ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access Failure
- 2006 - REDIR OR HANDOFF, this error code is returned when data call is brought down because device is in the process of redirecting/handing off to a different target system
- 2500 - OFFLINE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device went offline
- 2501 - EMERGENCY MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device is operating in Emergency mode
- 2502 - PHONE IN USE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device is in use (e.g voice call)

- 2503 - INVALID MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the device's operational mode is different from the mode requested in the traffic channel bring up
- 2504 - INVALID SIM STATE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the SIM was marked by network as invalid for circuit and/or packet service domain
- 2505 - NO COLLOC HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is no collocated HDR
- 2506 - CALL CONTROL REJECTED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since Call control module rejected the request

9.32.2.9 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 qaGobiApiTableSwiOMADMSessionStatus.h File Reference

Session Status table.

9.42.1 Detailed Description

Session Status table.

9.42.2 OMA DM Session Status (Session Status - Meaning - Usage)

- 0x0000 - Successful - Session succeeded
- 0x0001 - Break - Session succeeded
- 0x0002 - Large Object Handled - Session succeeded
- 0x0003 - No status - Session succeeded
- 0x0004 - No more commands - Session succeeded
- 0x0005 - User cancel - Session cancelled
- 0x0100-0x01FF - General errors
- 0x0200-0x02FF - Syncml errors
- 0x0300-0x03FF - Authentication errors
- 0x0400-0x04FF - Protocol errors
- 0x0500-0x05FF - Tree errors (DM Only)
- 0x0600-0x06FF - Not applicable
- 0x0700-0x07FF - Trigger errors
- 0x0800-0x08FF - FUMO errors
- 0x0900-0x09FF - Communication errors
- 0x0A00-0x0AFF - Parsing errors
- 0x0B00-0x0CFF - Not applicable
- 0x7F00-0x7F12 - Insignia errors
- 0x7F13 - Illegal text - Text received contains illegal characters
- 0x7F14 - Download failure - Failed to download FOTA image

- 0x7F15 - Empty session - Session ran successfully, but no information was updated
- 0x7F16 - Factory reset successful - Factory reset succeeded
- 0x7F17 - Factory reset fail - Factory reset failed

Copyright: © 2017 Sierra Wireless, Inc. all rights reserved

9.43 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

9.43.1 Detailed Description

Update Complete Status table.

9.43.2 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

9.44 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

9.44.1 Detailed Description

Voice Service Call and supplementary services end reasons.

9.44.2 Voice Call and supplementary services end reason codes (Code - Reason)

9.44.2.1 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

9.44.2.2 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

9.44.2.3 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

9.44.2.4 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

9.44.2.5 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

9.44.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

9.44.2.7 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

9.44.2.8 reject causes

- 237 - Service option is not supported

9.44.2.9 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

9.45 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)

9.45.1 Detailed Description

Thermal Mitigation Device API function prototypes.

9.45.2 Macro Definition Documentation

9.45.2.1 #define MAX_MITIGATION_DEV_ID_LEN 255

9.45.2.2 #define MAX_MITIGATION_DEV_LIST_LEN 255

9.45.3 Function Documentation

9.45.3.1 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

9.45.3.2 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

9.45.3.3 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

9.45.3.4 ULONG SLQSTmdRegNotMitigationLvl (TmdRegNotMitigationLvlReq * *pTmdRegNotMitigationLvlReq*)

Registers for notification of mitigation device level changes.

Parameters

pTmdRegNot-MitigationLvl-Req[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

9.46 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)

9.46.1 Detailed Description

Uim Service API function prototypes.

9.46.2 Macro Definition Documentation

9.46.2.1 #define MAX_ACTIVE_PERS_FEATURES 12

9.46.2.2 #define MAX_CONTENT_LENGTH 1024

9.46.2.3 #define MAX_DESCRIPTION_LENGTH 255

9.46.2.4 #define MAX_ICCID_LENGTH 255

9.46.2.5 #define MAX_NO_OF_APPLICATIONS 10

9.46.2.6 #define MAX_NO_OF_SLOTS 5

9.46.2.7 #define MAX_PATH_LENGTH 255

9.46.2.8 #define MAX_PUK_LENGTH 8

9.46.2.9 #define MAX_SLOTS_STATUS 255

9.46.3 Function Documentation

9.46.3.1 ULONG SLQSUIMAuthenticate (UIMAuthenticateReq * *pUIMAuthenticateReq*, UIMAuthenticateResp * *pUIMAuthenticateResp*)

This API executes the authentication algorithm on the card.

Parameters

pUIMAuthenticateReq[IN]	• See UIMAuthenticateReq for more information.
pUIMAuthenticateResp[OUT]	• See UIMAuthenticateResp for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 30 Secs

```
This API executes a security command on the card that depends on
the card type.
The response contains the status code received from the card
(SW1 and SW2) when the card responded to the read request.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMAuthenticateCallback
```

9.46.3.2 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.
```

9.46.3.3 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.
```

9.46.3.4 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.
```

9.46.3.5 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.
```

9.46.3.6 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

9.46.3.7 ULONG SLQSUIMGetFileAttributes (UIMGetFileAttributesReq * *pUIMGetFileAttributesReq*, UIMGetFileAttributesResp * *pUIMGetFileAttributesResp*)

This API retrieves the file attributes for any Elementary File(EF) or Dedicated File(DF) in the card and provides access by the path.

Parameters

pUIMGetFileAttributesReq[IN]	• See UIMGetFileAttributesReq for more information.
pUIMGetFileAttributesResp[OUT]	• See UIMGetFileAttributesResp for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 30 Secs

```
This API retrieves the file attributes for any Elementary File(EF)
or Dedicated File(DF) in the card and provides access by the path.
The response contains the status code received from the card
(SW1 and SW2) when the card responded to the select request.
The client can pass a token in the request to receive the result
in a subsequent SLQSUIMGetFileAttributesCallback.
```

9.46.3.8 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

9.46.3.9 ULONG SLQSUIMPowerDown (UIMPowerDownReq * *pUIMPowerDownReq*)

This API powers down the SIM card.

Parameters

pUIMPowerDownReq[IN]	• See UIMPowerDownReq for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 30 Secs

```
This function powers down the card.
This is usually performed when the phone is switched off or when
it is set to Airplane mode.
```

9.46.3.10 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.
```

9.46.3.11 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.
```

9.46.3.12 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
```

9.46.3.13 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 )
```

9.46.3.14 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.
```

9.46.3.15 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.
```

9.46.3.16 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.
```

9.46.3.17 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.
```

9.46.3.18 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.
```

9.46.3.19 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.
```

9.46.3.20 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.
```

9.47 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)

9.47.1 Detailed Description

Voice Service API function prototypes.

9.47.2 Macro Definition Documentation

9.47.2.1 #define MAX_CALL_NO_LEN 81

9.47.2.2 #define MAX_DESCRIPTION_LENGTH 255

9.47.2.3 #define MAX_NO_OF_CALLS 20

9.47.2.4 #define MAXUSSDLENGTH 182

9.47.2.5 #define PASSWORD_LENGTH 4

9.47.3 Enumeration Type Documentation

9.47.3.1 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

9.47.4 Function Documentation

9.47.4.1 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

9.47.4.2 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

9.47.4.3 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

9.47.4.4 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

9.47.4.5 ULONG SLQSVoiceALSSelectLine (voiceALSSelectLineInfo * *pVoiceALSSelectLineInfo*)

This API allows the user to select the preferred line.

Parameters

pVoiceALSSelectLineInfo[IN]	• See voiceALSSelectLineInfo for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 30 Secs
This API allows the user to select the preferred line, and the status is updated on the card. The API is supported only for specific SIM/USIMs that support alternate line service. This command is applicable only in 3GPP devices. A No Effect error is returned if the update on the card fails.

9.47.4.6 ULONG SLQSVoiceALSSetLineSwitching (voiceALSSetLineSwitchInfo * *pVoiceALSSetLineSwitchInfo*)

This API sets the line switch setting on the card.

Parameters

pVoiceALSSetLineSwitchInfo[IN]	• See voiceALSSetLineSwitchInfo for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 30 Secs
This API sets a line to be switchable or unswitchable, and the switch status is updated on the card. The API is supported only for specific SIM/USIMs that support alternate line service. This command is applicable only in 3GPP devices. A No Effect error is returned if the update on the card fails.

9.47.4.7 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.

9.47.4.8 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.

9.47.4.9 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.

9.47.4.10 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. SLQSVoiceSetAllCallStatusCallback() callback can be subscribed to learn if the call was successful.

9.47.4.11 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.

9.47.4.12 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.

9.47.4.13 ULONG SLQSVoiceGetCallBarring (voiceGetCallBarringReq * *pVoiceGetCallBarringReq*, voiceGetCallBarringResp * *pVoiceGetCallBarringResp*)

Queries the status of Call Barring Supplementary Service (applicable only for 3GPP).

Parameters

pVoiceGetCallBarringReq[IN]	• Pointer to structure of voiceGetCallBarringReq – See voiceGetCallBarringReq for more information
pVoiceGetCallBarringResp[OUT]	• 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.

9.47.4.14 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.

9.47.4.15 ULONG SLQSVoiceGetCallInfo (voiceCallInfoReq * *pGetCallInfoReq,* voiceCallInfoResp * *pGetCallInfoResp*)

This API queries the information associated with a call and gives information about a particular call whose call Id is sent in as request.

Parameters

pGetCallInfoReq[IN]	• See voiceCallInfoReq for more information.
pGetCallInfoResp[OUT]	• See voiceCallInfoResp for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 30 Secs
If no voice call is in progress or an invalid call_id is sent in the request, an error is returned as the response.
This API requires a firmware with atleast voice 2.0 support.

9.47.4.16 ULONG SLQSVoiceGetCallWaiting (voiceGetCallWaitInfo * *pVoiceGetCallWaitInfo*)

Queries the status of Call Waiting Supplementary Service (applicable only for 3GPP).

Parameters

pVoiceGetCallWaitInfo[IN/OUT]	• Pointer to structure of voiceGetCallWaitInfo – 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.

9.47.4.17 ULONG SLQSVoiceGetCLIP (voiceGetCLIPResp * *pVoiceGetCLIPResp*)

Queries the status of the Calling Line Identification Presentation (CLIP) supplementary service (applicable only for 3GPP).

Parameters

pVoiceGetCLIPResp[OUT]	• Pointer to structure of voiceGetCLIPResp – 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.

9.47.4.18 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.

9.47.4.19 ULONG SLQSVoiceGetCNAP (voiceGetCNAPResp * *pVoiceGetCNAPResp*)

Queries the status of the Calling Name Presentation(CNAP) supplementary service (applicable only for 3GPP).

Parameters

pVoiceGetCNA-PResp[OUT]	• Pointer to structure of voiceGetCNAPResp – 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.

9.47.4.20 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.

9.47.4.21 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.

9.47.4.22 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.

9.47.4.23 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.
```

9.47.4.24 ULONG SLQSVoiceManageCalls (voiceManageCallsReq * *pVoiceManageCallsReq*, voiceManageCallsResp * *pVoiceManageCallsResp*)

Manages the calls by using the supplementary service applicable during the call. In cases of successful API completion if the state of any call is changed, it is indicated using CallBack SLQSVoiceSetAllCallStatusCallBack. If there are other calls while an incoming voice call (waiting call) is received, this API is used to answer the call. This API is applicable only in "3GPP devices".

Parameters

pVoiceManage-CallsReq[IN]	• Request structure of to manage calls.
pVoiceManage-CallsResp[OUT]	• Response Structure to manage Calls

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 10 Secs
Applicable only for "3GPP"

9.47.4.25 ULONG SLQSVoiceOrigUSSDNoWait (voiceOrigUSSDNoWaitInfo * *pVoiceOrigUSSDNoWaitInfo*)

This API initiates a USSD operation such that the response for this request is returned immediately and the data is returned via an indication.

Parameters

pVoiceOrigUSS-DNoWaitInfo[IN]	• See voiceOrigUSSDNoWaitInfo for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS
Timeout: 30 Secs
This API starts a new USSD operation. The response to the request is sent immediately. The response result is sent to the client via the SLQSVoiceOrigUSSDNoWaitCallback. This command is applicable only in 3GPP devices.

9.47.4.26 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.

9.47.4.27 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

pSetCallBarring-PwdResp[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.

9.47.4.28 ULONG SLQSVoiceSetConfig (voiceSetConfigReq * *pVoiceSetConfigReq*, voiceSetConfigResp * *pVoiceSetConfigResp*)

This message sets various configuration parameters that control the modem behavior related to circuit-switched services.

Parameters

pVoiceSet-ConfigReq[IN]	• Pointer to structure of voiceSetConfigReq – See voiceSetConfigReq for more information
pVoiceSet-ConfigResp[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.

9.47.4.29 ULONG SLQSVoiceSetPreferredPrivacy (voiceSetPrefPrivacy * *pSetPrefPrivacy*)

This API sets the voice privacy preference. Applicable only for 3GPP2 devices.

Parameters

pSetPref-Privacy[IN]	• See voiceSetPrefPrivacy for more information.

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: CDMA
Device Supported: SL9090
Timeout: 10 Secs

9.47.4.30 ULONG SLQSVoiceSetSUPSService (voiceSetSUPSServiceReq * *pVoiceSetSUPSServiceReq,* voiceSetSUPSServiceResp * *pVoiceSetSUPSServiceResp*)

This API manages call-independent supplementary services, e.g., activation of call forwarding (to forward incoming calls to a third party), activation of call barring (to request the network to block some of the call attempts), and activation of call waiting (to be notified of an incoming call even when the user is engaged in an active or held call).

Parameters

pVoiceSetSUPS-ServiceReq	[IN] • Pointer to structure of voiceSetSUPSServiceReq – See voiceSetSUPSServiceReq for more information
pVoiceSetSUPS-ServiceResp	[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

9.47.4.31 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.

9.47.4.32 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.

9.48 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)

9.48.1 Detailed Description

Wireless Data Service API function prototypes.

9.48.2 Macro Definition Documentation

9.48.2.1 #define IPV6_ADDRESS_ARRAY_SIZE 8

9.48.3 Typedef Documentation

9.48.3.1 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

9.48.3.2 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.

9.48.3.3 typedef union WdsProfileParam QmiProfileInfo

This union WdsProfileParam consist of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

Parameters

SlqsProfile3GPP	• 3GPP profile
SlqsProfile3GPP2	• 3GPP2 profile

9.48.3.4 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[OUT]	• current data bearer technology value. – NULL if the parameter is not required
pLastCallData-Bearer-Technology[OUT]	• last call data bearer technology value. – NULL if the parameter is not required

9.48.3.5 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

9.48.3.6 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

9.48.4 Enumeration Type Documentation

9.48.4.1 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

9.48.5 Function Documentation

9.48.5.1 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

9.48.5.2 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

9.48.5.3 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

9.48.5.4 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

9.48.5.5 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) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
pIPAddress[OUT]	• Preferred IPv4 address to be assigned to device
pPrimaryDNS[OUT]	• Primary DNS Ipv4 address preference
pSecondaryDNS[OUT]	• Secondary DNS Ipv4 address preference
pAuthentication[OUT]	• Bitmap that indicates authentication algorithm preference – 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

9.48.5.6 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) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
pIPAddressv4[OUT]	• Preferred IPv4 addr to be assigned to device
pPrimaryDNSv4[OUT]	• Primary DNS Ipv4 address preference

pSecondaryDNSv4[OUT]	• Secondary DNS Ipv4 address preference
pIPAddressv6[OUT]	• Preferred IPv6 addr to be assigned to device Space for storing 8 element array for the IPv6 addresses is allocated by the application. The IP Address will be retrieved in the big endian format. For example User buffer contents: [<U0>..<U7>] IPv6 address: 1234:2A01:....:5678 U0 corresponds to 1234 U1 corresponds to 2A01 – ----------— – -— – ----------— – -— U7 corresponds to 5678
pPrimaryDNSv6[OUT]	• Primary DNS Ipv6 address preference
pSecondaryDNSv6[OUT]	• Secondary DNS Ipv6 address preference
pAuthentication[OUT]	• Bitmap that indicates authentication algorithm preference – 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

9.48.5.7 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
```

9.48.5.8 ULONG GetDormancyState (ULONG * *pDormancyState*, BYTE *instance*)

Returns the dormancy state of the current packet data session when connected.

Parameters

pDormancyState[OUT]	• Dormancy state of current packet data session – 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

9.48.5.9 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.

9.48.5.10 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

9.48.5.11 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

9.48.5.12 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

9.48.5.13 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

9.48.5.14 ULONG GetPacketStatus (ULONG * *pTXPacketSuccesses*, ULONG * *pRXPacketSuccesses*, ULONG * *pTXPacketErrors*, ULONG * *pRXPacketErrors*, ULONG * *pTXPacketOverflows*, ULONG * *pRXPacketOverflows*, BYTE *instance*)

Returns the packet data transfer statistics since the start of the current packet data.

Parameters

pTXPacket-Successes[OUT]	• Number of packets transmitted without error
pRXPacket-Successes[OUT]	• Number of packets received without error
pTXPacket-Errors[OUT]	• Number of outgoing packets with framing errors

pRXPacket-Errors[OUT]	• Number of incoming packets with framing errors
pTXPacket-Overflows[OUT]	• Number of packets dropped because Tx buffer overflowed
pRXPacket-Overflows[OUT]	• Number of packets dropped because Rx buffer overflowed
instance	• PDP instance

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 2 seconds

9.48.5.15 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

9.48.5.16 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

9.48.5.17 ULONG iGetByteTotals (ULONG * *pv4sessionId,* ULONG * *pv6sessionId,* struct WdsByteTotalsElmnts * *pByteTotalsElmnt*)

9.48.5.18 ULONG iGetConnectionRate (ULONG * *pv4sessionId,* ULONG * *pv6sessionId,* struct WdsConnectionRateElmnts * *pConnectionRateElmnt*)

9.48.5.19 ULONG iGetPacketStatistics (ULONG * *pV4sessionId,* ULONG * *pV6sessionId,* struct WdsPktStatisticsReq * *pStatMask,* struct WdsPktStatisticsElmnts * *pPktStatisticsElmnt*)

9.48.5.20 ULONG iSLQSMISetIPFamilyPreference (BYTE *IPFamilyPreference,* BYTE *instance*)

9.48.5.21 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
```

9.48.5.22 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

9.48.5.23 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

9.48.5.24 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) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
pIPAddress[IN]	• Preferred IPv4 addr to be assigned to device (optional)
pPrimaryDNS[IN]	• Primary DNS Ipv4 address preference (optional)
pSecondaryDNS[IN]	• Secondary DNS Ipv4 address preference (optional)
pAuthentication[IN]	• Bitmap that indicates authentication algorithm preference (optional) – 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.

9.48.5.25 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) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
pIPAddressv4[IN]	• Preferred IPv4 address to be assigned to device (optional)
pPrimaryDNSv4[IN]	• Primary DNS Ipv4 address preference (optional)
pSecondaryDNSv4[IN]	• Secondary DNS Ipv4 address preference (optional)
pIPAddressv6[IN]	• Preferred IPv6 address to be assigned to device (optional)
pPrimaryDNSv6[IN]	• Primary DNS Ipv6 address preference (optional)

pSecondaryDNSv6[IN]	• Secondary DNS Ipv6 address preference (optional)
pAuthentication[IN]	• Bitmap that indicates authentication algorithm preference (optional) – 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

9.48.5.26 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) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
pIPAddressv4[IN]	• Preferred IPv4 address to be assigned to device (optional)
pPrimaryDNSv4[IN]	• Primary DNS Ipv4 address preference (optional)
pSecondaryDNSv4[IN]	• Secondary DNS Ipv4 address preference (optional)
pIPAddressv6[IN]	• Preferred IPv6 addr to be assigned to device (optional)
pPrimaryDNSv6[IN]	• Primary DNS Ipv6 address preference (optional)
pSecondaryDNSv6[IN]	• Secondary DNS Ipv6 address preference (optional)
pAuthentication[IN]	• Bitmap that indicates authentication algorithm preference (optional) – 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

9.48.5.27 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
```

9.48.5.28 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

9.48.5.29 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

9.48.5.30 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

9.48.5.31 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

9.48.5.32 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

9.48.5.33 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

9.48.5.34 ULONG SLQSGet3GPPConfigItem (slqs3GPPConfigItem * *pSLQS3GPPConfigItem*)

Reads the 3gpp configuration item.

Parameters

pSLQS3GPP-ConfigItem[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

9.48.5.35 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
```

9.48.5.36 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

9.48.5.37 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
```

9.48.5.38 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 SLQSSetCustFeatures

Timeout: 2 seconds

9.48.5.39 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

qmiWDSDataBearers[OUT]	• Indicates the current and the last call data bearer technology. Should not be NULL, on input.
instance	• PDP instance

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Technology Supported: UMTS/CDMA
Device Supported: MC7750,GOBI,MC7700
Timeout: 2 seconds

9.48.5.40 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

pDataBearerTech	[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
```

9.48.5.41 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

9.48.5.42 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[O-UT]	– 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
```

9.48.5.43 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) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
pIPAddress[OUT]	• Preferred IPv4 address to be assigned to device
pPrimaryDNS[OUT]	• Primary DNS Ipv4 address preference
pSecondaryDNS[OUT]	• Secondary DNS Ipv4 address preference
pAuthentication[OUT]	• Bitmap that indicates authentication algorithm preference – 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

9.48.5.44 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

9.48.5.45 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

9.48.5.46 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

9.48.5.47 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

9.48.5.48 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
```

9.48.5.49 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

9.48.5.50 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) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
pIPAddress[IN]	• Preferred IPv4 address to be assigned to device (optional)
pPrimaryDNS[IN]	• Primary DNS Ipv4 address preference (optional)

pSecondaryDNS[IN]	• Secondary DNS Ipv4 address preference (optional)
pAuthentication[IN]	• Bitmap that indicates authentication algorithm preference (optional) – 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.

9.48.5.51 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
```

9.48.5.52 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
```

9.48.5.53 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
```

9.48.5.54 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
```

9.48.5.55 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

9.48.5.56 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

9.48.5.57 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

9.48.5.58 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

9.48.5.59 ULONG SLQSWdsSwiPDPRuntimeSettings (swiPDPRuntimeSettingsReq * *pPDPRuntimeSettingsReq*, swiPDPRuntimeSettingsResp * *pPDPRuntimeSettingsResp*)

This API requests the device to return the active PDP context associated with a context id.

Parameters

pPDPRuntime-SettingsReq[IN]	• See swiPDPRuntimeSettingsReq for more information
pPDPRuntime-SettingsResp[O-UT]	• 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

9.48.5.60 BOOL WDS_IsGobiDevice ( )

9.49 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)

9.49.1 Enumeration Type Documentation

9.49.1.1 enum eQMI_NAS_GET_RF_INFO_RESP

Enumerator

eTLV_RF_BAND_INFO

9.49.2 Function Documentation

9.49.2.1 enum eQCWWANError PkQmiNasGetRFBandInfo (WORD * *pMlength,* BYTE * *pBuffer*)

9.49.2.2 enum eQCWWANError UpkQmiNasGetRFBandInfo (BYTE * *pMdmResp,* struct QmiNasGetRFBandInfoResp * *pApiResp*)

9.50 qaNasPerformNetworkScan.h File Reference

Data Structures

- struct QmiNas3GppNetworkInfo
- struct QmiNasPerformNetworkScanResp

Macros

- #define QMI_NAS_NETSTATUS_MASK 0x03
- #define QMI_NAS_MAX_INSTANCES 20
- #define INDEX_ZERO 0
- #define ROAMING_INDEX 2
- #define FORBIDDEN_INDEX 4
- #define PREFERRED_INDEX 6
- #define MAX_DESCRIPTION_LENGTH 255

Enumerations

- enum eQMI_NAS_PERFORM_NETWORK_SCAN_RESP { eTLV_3GPP_NETWORK_INFO = 0x10 }

Functions

- enum eQCWWANError PkQmiNasPerformNetworkScan (WORD *pMlength, BYTE *pParamField)
- enum eQCWWANError UpkQmiNasPerformNetworkScan (BYTE *pMdmResp, struct QmiNasPerformNetworkScanResp *pAipResp)

9.50.1 Macro Definition Documentation

9.50.1.1 #define FORBIDDEN_INDEX 4

9.50.1.2 #define INDEX_ZERO 0

9.50.1.3 #define MAX_DESCRIPTION_LENGTH 255

9.50.1.4 #define PREFERRED_INDEX 6

9.50.1.5 #define QMI_NAS_MAX_INSTANCES 20

9.50.1.6 #define QMI_NAS_NETSTATUS_MASK 0x03

9.50.1.7 #define ROAMING_INDEX 2

9.50.2 Enumeration Type Documentation

9.50.2.1 enum eQMI_NAS_PERFORM_NETWORK_SCAN_RESP

Enumerator

eTLV_3GPP_NETWORK_INFO

9.50.3 Function Documentation

9.50.3.1 enum eQCWWANError PkQmiNasPerformNetworkScan (WORD * *pMlength*, BYTE * *pParamField*)

9.50.3.2 enum eQCWWANError UpkQmiNasPerformNetworkScan (BYTE * *pMdmResp*, struct QmiNasPerformNetworkScanResp * *pAipResp*)

9.51 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 }

9.51.1 Enumeration Type Documentation

9.51.1.1 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

9.51.1.2 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

9.52 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)

9.52.1 Macro Definition Documentation

9.52.1.1 #define LITEQMI_MAX_QOS_FILTERS 25

9.52.1.2 #define LITEQMI_MAX_QOS_FLOW_PER_APN_STATS 10

9.52.1.3 #define LITEQMI_MAX_QOS_FLOWS 8

9.52.2 Function Documentation

9.52.2.1 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• 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

9.52.2.2 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes

in,out	*pLen*	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
in	*reqParam*	• See pack_qos_SLQSQosSwiReadApnExtraParams_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- Timeout: 2 seconds
  - PDN Specific: Yes

9.52.2.3 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
in	*reqParam*	• See pack_qos_SLQSQosSwiReadDataStats_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- Timeout: 2 seconds
  - PDN Specific: Yes

9.52.2.4 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
in	*reqParam*	• See pack_qos_SLQSSetQosEventCallback_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- Timeout: 2 seconds
  - PDN Specific: Yes

9.52.2.5 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• See unpack_qos_SLQSQosGetNetworkStatus_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.52.2.6 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• See unpack_qos_SLQSQosSwiReadApnExtraParams_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.52.2.7 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• See unpack_qos_SLQSQosSwiReadDataStats_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.52.2.8 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.52.2.9 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• See unpack_qos_SLQSSetQosEventCallback_ind_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_NETWORK_STATUS_IND indication to identify this event from QOS service read function

9.52.2.10 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• See unpack_qos_SLQSSetQosNWStatusCallback_ind_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- Technology Supported: CDMA
- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_NETWORK_STATUS_IND indication to identify this event from QOS service read function

9.52.2.11 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• See unpack_qos_SLQSSetQosPriEventCallback_ind_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_PRIMARY_QOS_EVENT_IND indication to identify this event from QOS service read function
- This is only generated when the primary flow is modified by the host

9.52.2.12 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• See unpack_qos_SLQSSetQosStatusCallback_ind_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_FLOW_STATUS_IND indication to identify this event from QOS service read function

9.53 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)

9.53.1 Function Documentation

9.53.1.1 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	*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

9.53.1.2 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.53.1.3 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

9.53.1.4 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

9.54 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)

9.54.1 Function Documentation

9.54.1.1 int pack_sar_SLQSGetRfSarState (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen*)

Gets the specified RF SAR state pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.1.2 int pack_sar_SLQSSetRfSarState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sar_SLQSSetRfSarState_t * *reqArg*)

Sets the specified RF SAR state pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.54.1.3 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

9.54.1.4 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

9.55 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)

9.55.1 Macro Definition Documentation

9.55.1.1 #define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255

9.55.1.2 #define MAX_MS_TRANSFER_ROUTE_MSG 256

9.55.1.3 #define MAX_MSC_ADDRESS_SIZE 256

9.55.1.4 #define MAX_MSE_TWS_MSG 1254

9.55.1.5 #define MAX_SMS_LIST_SIZE 255

9.55.1.6 #define MAX_SMS_MESSAGE_SIZE 2048

9.55.1.7 #define SMS_CONFIG_LEN 0x05

9.55.1.8 #define SMS_MAX_SMS_ROUTES 0x0A

9.55.1.9 #define SMS_NUM_OF_SET 0xFF

9.55.1.10 #define SMSC_TYPE_LEN 0x03

9.55.2 Typedef Documentation

9.55.2.1 typedef struct sMSCAddress sMSCAddressInfo

Parameters

length	• Number of sets of following element
data	• SMSC address

9.55.2.2 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

9.55.2.3 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

9.55.2.4 typedef struct sMSMessageMode sMSMessageModeInfo

Parameters

messageMode	Message Mode

9.55.2.5 typedef struct sMSMTMessage sMSMTMessageInfo

Parameters

storageType	memory storage 0x00-UIM 0x01-NV
messageIndex	MT Message index

9.55.2.6 typedef struct sMSOnIMS sMSOnIMSInfo

Parameters

smsOnIMS	SMS on IMS

9.55.2.7 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

9.55.3 Enumeration Type Documentation

9.55.3.1 enum eqmiCbkSetStatus

Enumerator

LITEQMI_QMI_CBK_PARAM_RESET

LITEQMI_QMI_CBK_PARAM_SET

LITEQMI_QMI_CBK_PARAM_NOCHANGE

9.55.4 Function Documentation

9.55.4.1 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

9.55.4.2 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

9.55.4.3 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

9.55.4.4 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

9.55.4.5 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

9.55.4.6 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

9.55.4.7 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

9.55.4.8 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

9.55.4.9 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

9.55.4.10 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

9.55.4.11 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

9.55.4.12 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

9.55.4.13 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

9.55.4.14 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

9.55.4.15 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

9.55.4.16 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

9.55.4.17 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

9.55.4.18 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

9.55.4.19 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

9.55.4.20 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

9.55.4.21 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

9.55.4.22 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

9.55.4.23 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

9.55.4.24 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.25 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.26 int unpack_sms_SendSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SendSMS_t * *pOutput*)

send sms unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.27 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	*respLen*	qmi response length
out	*Output*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.28 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.29 int unpack_sms_SetSMSCAddress (uint8_t * *pResp*, uint16_t *respLen*)

Set the SMS center address unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.30 int unpack_sms_SLQSDeleteSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSDeleteSMS_t * *pOutput*)

delete sms unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.31 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.32 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.33 int unpack_sms_SLQSGetSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSMS_t * *pOutput*)

get sms unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.34 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.35 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.36 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.37 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.38 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.39 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	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.40 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	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.41 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	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.42 int unpack_sms_SLQSSetSmsBroadcastConfig (uint8_t * *pResp*, uint16_t *respLen*)

Sets the information about the SMS BroadcastConfiguration unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.43 int unpack_sms_SLQSSetSmsStorage (uint8_t * *pResp*, uint16_t *respLen*)

Sets the SMS Storage on the device unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.44 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.45 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.46 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	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.47 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.55.4.48 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.56 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

9.56.1 Detailed Description

SWI data types.

9.56.2 Macro Definition Documentation

9.56.2.1 #define QMI_NO_LTE_FW_SUPPORT 0

9.56.2.2 #define QMI_TLV_PLACEHOLDER 0x8F

9.56.2.3 #define SWI_API

9.56.2.4 #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.

9.56.3 Typedef Documentation

9.56.3.1 typedef int BOOL

9.56.3.2 typedef unsigned char BYTE

9.56.3.3 typedef char CHAR

9.56.3.4 typedef float FLOAT

9.56.3.5 typedef signed int INT32

9.56.3.6 typedef signed char INT8

9.56.3.7 typedef const char* LPCSTR

9.56.3.8 typedef signed short SHORT

9.56.3.9 typedef unsigned int ULONG

9.56.3.10 typedef unsigned long long ULONGLONG

9.56.3.11 typedef unsigned short USHORT

9.56.3.12 typedef unsigned short WORD

9.57 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)

9.57.1 Function Documentation

9.57.1.1 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

9.57.1.2 int pack_swiloc_SwiLocSetAutoStart (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiloc_SwiLocSetAutoStart_t * *reqArg*)

Set Auto Start pack

Parameters

in	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.57.1.3 int unpack_swiloc_SwiLocGetAutoStart (uint8_t * *pResp*, uint16_t *respLen*, unpack_swiloc_SwiLocGetAutoStart_t * *pOutput*)

Get Auto Start unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.57.1.4 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

9.58 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)

9.58.1 Macro Definition Documentation

9.58.1.1 #define LITEQMI_MAX_SWIOMA_STR_LEN 255

9.58.2 Function Documentation

9.58.2.1 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• 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

9.58.2.2 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• On input, size of pReqBuf • On output, number of bytes actually packed
in	*reqParam*	• 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

9.58.2.3 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes

in,out	*pLen*	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
in	*reqParam*	• 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

9.58.2.4 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• On input, size of pReqBuf • On output, number of bytes actually packed

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.58.2.5 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
in	*reqParam*	• 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

9.58.2.6 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
in	*reqParam*	• 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

9.58.2.7 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

out	*pCtx*	• See pack_qmi_t for more information
in,out	*pReqBuf*	• Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in,out	*pLen*	• On input, size of pReqBuf • On ouptut, number of bytes actually packed
in	*reqParam*	• 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

9.58.2.8 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

- Please use eQMI_SWIOMA_EVENT_IND indication to identify this event from SWIOMA service read function

9.58.2.9 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• 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

9.58.2.10 int unpack_swioma_SLQSOMADMCancelSession (uint8_t * *pResp*, uint16_t *respLen*)

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

Parameters

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.11 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• 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

9.58.2.12 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem

out	*pOutput*	• 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

9.58.2.13 int unpack_swioma_SLQSOMADMSendSelection (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack OMA-DM send selection command This maps to SLQSOMADMSendSelection2

Parameters

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.14 int unpack_swioma_SLQSOMADMSetSettings (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

Parameters

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.58.2.15 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

in	*pResp*	• Response from modem
in	*respLen*	• Length of pResp from modem
out	*pOutput*	• 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

9.59 SWIWWANCMAPI.h File Reference

9.60 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)

9.60.1 Macro Definition Documentation

9.60.1.1 #define MAX_DESCRIPTION_LENGTH 255

9.60.1.2 #define MAX_ICCID_LENGTH 255

9.60.1.3 #define MAX_NO_OF_APPLICATIONS 10

9.60.1.4 #define MAX_NO_OF_SLOTS 5

9.60.1.5 #define MAX_SLOTS_STATUS 255

9.60.1.6 #define UIM_MAX_ACTIVE_PERS_FEATURES 12

9.60.1.7 #define UIM_MAX_CONTENT_LENGTH 1024

9.60.1.8 #define UIM_MAX_DESCRIPTION_LENGTH 255

9.60.1.9 #define UIM_MAX_NO_OF_APPLICATIONS 10

9.60.1.10 #define UIM_MAX_NO_OF_SLOTS 5

9.60.1.11 #define UIM_UINT8_MAX_STRING_SZ 255

9.60.2 Function Documentation

9.60.2.1 int pack_uim_ChangePin (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_ChangePin_t * *reqArg*)

Change Pin pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.2 int pack_uim_GetCardStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Card Status pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.3 int pack_uim_ReadTransparent (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_ReadTransparent_t * *reqArg*)

SLQS ReadTransparent pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.4 int pack_uim_SetPinProtection (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SetPinProtection_t * *reqArg*)

Set Pin Protection pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.5 int pack_uim_SLQSUIMAuthenticate (**pack_qmi_t** * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, **pack_uim_SLQSUIMAuthenticate_t** * *reqArg*)

Send a security command to the card pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.6 int pack_uim_SLQSUIMDepersonalization (**pack_qmi_t** * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, **pack_uim_SLQSUIMDepersonalization_t** * *reqArg*)

De-activates or unblocks the personalization on the phone pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.7 int pack_uim_SLQSUIMEventRegister (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMEventRegister_t * *reqArg*)

UIM Status Change callback enable pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.8 int pack_uim_SLQSUIMGetConfiguration (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMGetConfiguration_t * *reqArg*)

Gets the modem configuration for the UIM module pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.9 int pack_uim_SLQSUIMGetFileAttributes (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMGetFileAttributes_t * *reqArg*)

Retrieves the file attributes for any EF(Elementary File) or DF(Dedicated File) in the card and provides access by the path pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.10 int pack_uim_SLQSUIMGetSlotsStatus (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

9.60.2.11 int pack_uim_SLQSUIMPowerDown (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMPowerDown_t * *reqArg*)

Powers down the card pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.12 int pack_uim_SLQSUIMPowerUp (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMPowerUp_t * *reqArg*)

Powers up the card pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.13 int pack_uim_SLQSUIMRefreshComplete (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshComplete_t * *reqArg*)

Enables the terminal response to be sent to the card When the client finished Refresh procedure pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.14 int pack_uim_SLQSUIMRefreshGetLastEvent (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshGetLastEvent_t * *reqArg*)

Retrieve the last refresh event pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.15 int pack_uim_SLQSUIMRefreshOK (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshOK_t * *reqArg*)

Enables the client to indicate whether it is OK to start the Refresh procedure pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.16 int pack_uim_SLQSUIMRefreshRegister (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshRegister_t * *reqArg*)

Registers for file change notifications triggered by the card pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.17 int pack_uim_SLQSUIMReset (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Resets the issuing control points state kept by the service pack.

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.18 int pack_uim_SLQSUIMSwitchSlot (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMSwitchSlot_t * *reqArg*)

switch slot pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.19 int pack_uim_UnblockPin (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_UnblockPin_t * *reqArg*)

Unblock Pin pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.20 int pack_uim_VerifyPin (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_VerifyPin_t * *reqArg*)

Verify Pin Status pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.21 int unpack_uim_ChangePin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_ChangePin_t * *pOutput*)

Change Pin unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.22 int unpack_uim_GetCardStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_GetCardStatus_t * *pOutput*)

Get Card Status unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.23 int unpack_uim_ReadTransparent (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_ReadTransparent_t * *pOutput*)

SLQS ReadTransparent unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.24 int unpack_uim_SetPinProtection (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SetPinProtection_t * *pOutput*)

Set Pin Protection unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.25 int unpack_uim_SetUimSlotStatusChangeCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SetUimSlotStatusChangeCallback_ind_t * *pOutput*)

UIM Slot Status Change indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

use pack_uim_SLQSUIMEventRegister to subscribe

9.60.2.26 int unpack_uim_SLQSUIMAuthenticate (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMAuthenticate_t * *pOutput*)

Unpack the status code received from the card when card responded to the read request

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.27 int unpack_uim_SLQSUIMDepersonalization (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMDepersonalization_t * *pOutput*)

De-activates or unblocks the personalization on the phone unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.28 int unpack_uim_SLQSUIMEventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMEvent-Register_t * *pOutput*)

UIM Status Change callback enable unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.29 int unpack_uim_SLQSUIMGetConfiguration (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMGetConfiguration_t * *pOutput*)

Gets the modem configuration for the UIM module unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.30 int unpack_uim_SLQSUIMGetFileAttributes (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMGetFileAttributes_t * *pOutput*)

Retrieves the file attributes for any EF(Elementary File) or DF(Dedicated File) in the card and provides access by the path unpack.

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.31 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

9.60.2.32 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

9.60.2.33 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

9.60.2.34 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

9.60.2.35 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	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.36 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

9.60.2.37 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

9.60.2.38 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	*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_uim_SLQSUIMSetStatusChangeCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t * *pOutput*)

UIM Status Change indication unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

use pack_uim_SLQSUIMEventRegister to subscribe

9.60.2.40 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

9.60.2.41 int unpack_uim_UnblockPin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_UnblockPin_t * *pOutput*)

Unblock Pin unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.60.2.42 int unpack_uim_VerifyPin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_VerifyPin_t * *pOutput*)

Verify Pin unpack

Parameters

in	*pResp*	qmi response
in	*respLen*	qmi response length
out	*pOutput*	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 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

Enumerations

- enum liteServiceClassInformation {
  LITE_VOICE_SUPS_SRV_CLASS_NONE = 0x00,
  LITE_VOICE_SUPS_SRV_CLASS_VOICE = 0x01,
  LITE_VOICE_SUPS_SRV_CLASS_DATA = 0x02,
  LITE_VOICE_SUPS_SRV_CLASS_FAX = 0x04,
  LITE_VOICE_SUPS_SRV_CLASS_SMS = 0x08,
  LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC = 0x10,
  LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC = 0x20,
  LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS = 0x40,
  LITE_VOICE_SUPS_SRV_CLASS_PADACCESS = 0x80 }

Functions

- int pack_voice_OriginateUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_OriginateUSSD_t *reqArg)
- int unpack_voice_OriginateUSSD (uint8_t *pResp, uint16_t respLen)
- int pack_voice_AnswerUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_AnswerUSSD_t *reqArg)
- int unpack_voice_AnswerUSSD (uint8_t *pResp, uint16_t respLen)
- int pack_voice_CancelUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_CancelUSSD (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceDialCall (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceDialCall_t *pReq)
- int unpack_voice_SLQSVoiceDialCall (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceDialCall_t *pOutput)
- int pack_voice_SLQSVoiceEndCall (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceEndCall_t *pReq)
- int unpack_voice_SLQSVoiceEndCall (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceEndCall_t *pOutput)
- int pack_voice_SLQSVoiceSetSUPSService (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetSUPSService_t *pReq)
- int unpack_voice_SLQSVoiceSetSUPSService (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetSUPSService_t *pOutput)
- int pack_voice_SLQSVoiceSetConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetConfig_t *pReq)
- int unpack_voice_SLQSVoiceSetConfig (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetConfig_t *pOutput)
- int pack_voice_SLQSVoiceAnswerCall (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceAnswerCall_t *pReq)
- int unpack_voice_SLQSVoiceAnswerCall (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceAnswerCall_t *pOutput)
- int pack_voice_SLQSVoiceGetCLIR (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCLIR (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCLIR_t *pOutput)
- int pack_voice_SLQSVoiceGetCLIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCLIP (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCLIP_t *pOutput)
- int pack_voice_SLQSVoiceGetCallWaiting (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallWaiting_t *pReq)
- int unpack_voice_SLQSVoiceGetCallWaiting (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallWaiting_t *pOutput)
- int pack_voice_SLQSVoiceGetCallBarring (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallBarring_t *pReq)
- int unpack_voice_SLQSVoiceGetCallBarring (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallBarring_t *pOutput)
- int pack_voice_SLQSVoiceGetCallForwardingStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallForwardingStatus_t *pReq)
- int unpack_voice_SLQSVoiceGetCallForwardingStatus (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallForwardingStatus_t *pOutput)
- int pack_voice_SLQSVoiceSetCallBarringPassword (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetCallBarringPassword_t *pReq)
- int unpack_voice_SLQSVoiceSetCallBarringPassword (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetCallBarringPassword_t *pOutput)
- int pack_voice_SLQSVoiceGetCallInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallInfo_t *pReq)
- int unpack_voice_SLQSVoiceGetCallInfo (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallInfo_t *pOutput)

- int pack_voice_SLQSVoiceGetAllCallInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetAllCallInfo (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetAllCallInfo_t *pOutput)
- int pack_voice_SLQSVoiceManageCalls (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceManageCalls_t *pReq)
- int unpack_voice_SLQSVoiceManageCalls (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceManageCalls_t *pOutput)
- int pack_voice_SLQSVoiceBurstDTMF (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceBurstDTMF_t *pReq)
- int unpack_voice_SLQSVoiceBurstDTMF (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceBurstDTMF_t *pOutput)
- int pack_voice_SLQSVoiceStartContDTMF (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceStartContDTMF_t *pReq)
- int unpack_voice_SLQSVoiceStartContDTMF (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceStartContDTMF_t *pOutput)
- int pack_voice_SLQSVoiceStopContDTMF (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceStopContDTMF_t *pReq)
- int unpack_voice_SLQSVoiceStopContDTMF (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceStopContDTMF_t *pOutput)
- int pack_voice_SLQSVoiceSendFlash (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSendFlash_t *pReq)
- int unpack_voice_SLQSVoiceSendFlash (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSendFlash_t *pOutput)
- int pack_voice_SLQSVoiceSetPreferredPrivacy (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetPreferredPrivacy_t *reqArg)
- int unpack_voice_SLQSVoiceSetPreferredPrivacy (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceIndicationRegister_t *reqArg)
- int unpack_voice_SLQSVoiceIndicationRegister (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceGetConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetConfig_t *reqArg)
- int unpack_voice_SLQSVoiceGetConfig (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetConfig_t *pOutput)
- int pack_voice_SLQSVoiceOrigUSSDNoWait (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceOrigUSSDNoWait_t *reqArg)
- int unpack_voice_SLQSVoiceOrigUSSDNoWait (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceBindSubscription (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceBindSubscription_t *reqArg)
- int unpack_voice_SLQSVoiceBindSubscription (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceALSSetLineSwitching (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceALSSetLineSwitching_t *reqArg)
- int unpack_voice_SLQSVoiceALSSetLineSwitching (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceALSSelectLine (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceALSSelectLine_t *reqArg)
- int unpack_voice_SLQSVoiceALSSelectLine (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceGetCOLP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCOLP (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCOLP_t *pOutput)
- int pack_voice_SLQSVoiceGetCOLR (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCOLR (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCOLR_t *pOutput)
- int pack_voice_SLQSVoiceGetCNAP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCNAP (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCNAP_t *pOutput)
- int pack_voice_SLQSOriginateUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSOriginateUSSD_t *reqArg)
- int unpack_voice_SLQSOriginateUSSD (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSOriginateUSSD_t *pOutput)

9.61.1 Macro Definition Documentation

9.61.1.1 #define BARRING_PASSWORD_LENGTH 4

9.61.1.2 #define MAX_VOICE_CALL_NO_LEN 81

9.61.1.3 #define MAX_VOICE_DESCRIPTION_LENGTH 255

9.61.1.4 #define MAXVOICEUSSDLENGTH 182

9.61.1.5 #define VOICE_MAX_NO_OF_CALLS 20

9.61.2 Enumeration Type Documentation

9.61.2.1 enum liteServiceClassInformation

Service Class information

Enumerator

LITE_VOICE_SUPS_SRV_CLASS_NONE

LITE_VOICE_SUPS_SRV_CLASS_VOICE

LITE_VOICE_SUPS_SRV_CLASS_DATA

LITE_VOICE_SUPS_SRV_CLASS_FAX

LITE_VOICE_SUPS_SRV_CLASS_SMS

LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC

LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC

LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS

LITE_VOICE_SUPS_SRV_CLASS_PADACCESS

9.61.3 Function Documentation

9.61.3.1 int pack_voice_AnswerUSSD (pack_qmi_t * *pCtx,* uint8_t * *pReqBuf,* uint16_t * *pLen,* pack_voice_AnswerUSSD_t * *reqArg*)

Responds to a USSD request from the network pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.2 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

9.61.3.3 int pack_voice_OriginateUSSD (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_OriginateUSSD_t * *reqArg*)

Initiates a USSD operation pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.4 int pack_voice_SLQSOriginateUSSD (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSOriginateUSSD_t * *reqArg*)

Initiates a USSD session pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.5 int pack_voice_SLQSVoiceALSSelectLine (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceALSSelectLine_t * *reqArg*)

allows the user to select the preferred line pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.6 int pack_voice_SLQSVoiceALSSetLineSwitching (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceALSSetLineSwitching_t * *reqArg*)

sets the line switch setting on the card pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.7 int pack_voice_SLQSVoiceAnswerCall (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceAnswerCall_t * *pReq*)

Answers an incoming voice call pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.8 int pack_voice_SLQSVoiceBindSubscription (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceBindSubscription_t * *reqArg*)

binds a subscription type to a specific voice client ID pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.9 int pack_voice_SLQSVoiceBurstDTMF (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceBurstDTMF_t * *pReq*)

Sends a burst Dual-Tone Multi frequency (DTMF) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.10 int pack_voice_SLQSVoiceDialCall (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceDialCall_t * *pReq*)

Originates a voice call (MO call) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.11 int pack_voice_SLQSVoiceEndCall (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceEndCall_t * *pReq*)

Ends a voice call pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.12 int pack_voice_SLQSVoiceGetAllCallInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

information associated with all the calls originating or terminating pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.13 int pack_voice_SLQSVoiceGetCallBarring (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallBarring_t * pReq)

Status of Call Barring Supplementary Service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.14 int pack_voice_SLQSVoiceGetCallForwardingStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallForwardingStatus_t * pReq)

Status of Call Forwarding Supplementary Service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.15 int pack_voice_SLQSVoiceGetCallInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallInfo_t * pReq)

information associated with a call pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.16 int pack_voice_SLQSVoiceGetCallWaiting (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetCallWaiting_t * *pReq*)

Status of Call Waiting Supplementary Service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.17 int pack_voice_SLQSVoiceGetCLIP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Calling Line Identification Presentation (CLIP) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.18 int pack_voice_SLQSVoiceGetCLIR (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Calling Line Identification Restriction (CLIR) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.19 int pack_voice_SLQSVoiceGetCNAP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Calling Name Presentation(CNAP) pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.20 int pack_voice_SLQSVoiceGetCOLP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Connected Line Identification Presentation pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.21 int pack_voice_SLQSVoiceGetCOLR (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

status of the Connected Line Identification Restriction pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.22 int pack_voice_SLQSVoiceGetConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetConfig_t * *reqArg*)

retrieves various configuration parameters that control the modem behavior pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.23 int pack_voice_SLQSVoiceIndicationRegister (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceIndicationRegister_t * *reqArg*)

Sets the registration state for different QMI_VOICE indications pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.24 int pack_voice_SLQSVoiceManageCalls (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceManageCalls_t * *pReq*)

Manages the calls by using the supplementary service pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.25 int pack_voice_SLQSVoiceOrigUSSDNoWait (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceOrigUSSDNoWait_t * *reqArg*)

initiates a USSD operation No Wait pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.26 int pack_voice_SLQSVoiceSendFlash (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSendFlash_t * *pReq*)

sends a simple flash message pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.27 int pack_voice_SLQSVoiceSetCallBarringPassword (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetCallBarringPassword_t * *pReq*)

Sets a Call Barring Password pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.28 int pack_voice_SLQSVoiceSetConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetConfig_t * *pReq*)

configuration parameters that control the modem behavior pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.29 int pack_voice_SLQSVoiceSetPreferredPrivacy (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetPreferredPrivacy_t * *reqArg*)

sets the voice privacy preference pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.30 int pack_voice_SLQSVoiceSetSUPSService (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetSUPSService_t * *pReq*)

call-independent supplementary services pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.31 int pack_voice_SLQSVoiceStartContDTMF (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceStartContDTMF_t * *pReq*)

Starts a continuous DTMF pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.32 int pack_voice_SLQSVoiceStopContDTMF (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceStopContDTMF_t * *pReq*)

Stops a continuous DTMF pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.61.3.33 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

9.61.3.34 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

9.61.3.35 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

9.61.3.36 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

9.61.3.37 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

9.61.3.38 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

9.61.3.39 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

9.61.3.40 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

9.61.3.41 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

9.61.3.42 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

9.61.3.43 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

9.61.3.44 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

9.61.3.45 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

9.61.3.46 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

9.61.3.47 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

9.61.3.48 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

9.61.3.49 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

9.61.3.50 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

9.61.3.51 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

9.61.3.52 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

9.61.3.53 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

9.61.3.54 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

9.61.3.55 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

9.61.3.56 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

9.61.3.57 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

9.61.3.58 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

9.61.3.59 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

9.61.3.60 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

9.61.3.61 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

9.61.3.62 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

9.61.3.63 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

9.61.3.64 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

9.62 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

Enumerations

- enum liteQmiDataBearerMasks {
  QMI_LITE_WDS_CURRENT_CALL_DB_MASK = 0x01,
  QMI_LITE_WDS_LAST_CALL_DB_MASK = 0x02 }

Functions

- int pack_wds_SLQSStartDataSession (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_wds_SLQSStartDataSession_t ∗reqArg)
- int unpack_wds_SLQSStartDataSession (uint8_t ∗pResp, uint16_t respLen, unpack_wds_SLQSStartDataSession_t ∗pOutput)
- int unpack_wds_SLQSSetPacketSrvStatusCallback (uint8_t ∗pResp, uint16_t respLen, unpack_wds_SLQSSetPacketSrvStatusCallback_t ∗pOutput)
- int pack_wds_SLQSStopDataSession (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_wds_SLQSStopDataSession_t ∗reqArg)
- int unpack_wds_SLQSStopDataSession (uint8_t ∗pResp, uint16_t respLen)

- int unpack_wds_SLQSGetRuntimeSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetRuntimeSettings_t *pOutput)
- int unpack_wds_SLQSSetWdsEventCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetWdsEventCallback_ind_t *pOutput)
- int unpack_wds_SLQSSetWdsEventCallback (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSSetWdsEventCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetWdsEventCallback_t *reqArg)
- int pack_wds_SLQSGetRuntimeSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetRuntimeSettings_t *reqArg)
- int pack_wds_SLQSCreateProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSCreateProfile_t *reqArg)
- int unpack_wds_SLQSCreateProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSCreateProfile_t *pOutput)
- int pack_wds_SLQSModifyProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSModifyProfile_t *reqArg)
- int unpack_wds_SLQSModifyProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSModifyProfile_t *pOutput)
- int pack_wds_SLQSGetProfileSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetProfileSettings_t *reqArg)
- int unpack_wds_SLQSGetProfileSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetProfileSettings_t *pOutput)
- int pack_wds_GetSessionState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetSessionState (uint8_t *pResp, uint16_t respLen, unpack_wds_GetSessionState_t *pOutput)
- int pack_wds_GetDefaultProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfile_t *reqParam)
- int unpack_wds_GetDefaultProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDefaultProfile_t *pOutput)
- int pack_wds_GetConnectionRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetConnectionRate (uint8_t *pResp, uint16_t respLen, unpack_wds_GetConnectionRate_t *pOutput)
- int pack_wds_GetPacketStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatus_t *reqParam)
- int unpack_wds_GetPacketStatus (uint8_t *pResp, uint16_t respLen, unpack_wds_GetPacketStatus_t *pOutput)
- int pack_wds_GetSessionDuration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetSessionDuration_t *reqParam)
- int unpack_wds_GetSessionDuration (uint8_t *pResp, uint16_t respLen, unpack_wds_GetSessionDuration_t *pOutput)
- int pack_wds_GetDormancyState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDormancyState_t *reqParam)
- int unpack_wds_GetDormancyState (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDormancyState_t *pOutput)
- int pack_wds_SLQSDeleteProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSDeleteProfile_t *reqParam)
- int unpack_wds_SLQSDeleteProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSDeleteProfile_t *pOutput)
- int pack_wds_SetDefaultProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfile_t *reqParam)
- int unpack_wds_SetDefaultProfile (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSGet3GPPConfigItem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSGet3GPPConfigItem (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGet3GPPConfigItem_t *pOutput)
- int pack_wds_SLQSSet3GPPConfigItem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSet3GPPConfigItem_t *reqParam)
- int unpack_wds_SLQSSet3GPPConfigItem (uint8_t *pResp, uint16_t respLen)

- int pack_wds_GetMobileIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIP_t *pReqParam)
- int unpack_wds_GetMobileIP (uint8_t *pResp, uint16_t respLen, unpack_wds_GetMobileIP_t *pOutput)
- int pack_wds_GetMobileIPProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIPProfile_t *reqParam)
- int unpack_wds_GetMobileIPProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_GetMobileIPProfile_t *pOutput)
- int pack_wds_SLQSGetCurrDataSystemStat (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetCurrDataSystemStat_t *pReqParam)
- int unpack_wds_SLQSGetCurrDataSystemStat (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrDataSystemStat_t *pOutput)
- int pack_wds_GetLastMobileIPError (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetLastMobileIPError_t *pReqParam)
- int unpack_wds_GetLastMobileIPError (uint8_t *pResp, uint16_t respLen, unpack_wds_GetLastMobileIPError_t *pOutput)
- int pack_wds_RMSetTransferStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_RMSetTransferStatistics_t *reqParam)
- int unpack_wds_RMSetTransferStatistics (uint8_t *pResp, uint16_t respLen, unpack_wds_RMSetTransferStatistics_t *pOutput)
- int pack_wds_SetMobileIPProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPProfile_t *reqParam)
- int unpack_wds_SetMobileIPProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SetMobileIPProfile_t *pOutput)
- int pack_wds_SLQSWdsSwiPDPRuntimeSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t *reqParam)
- int unpack_wds_SLQSWdsSwiPDPRuntimeSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t *pOutput)
- int pack_wds_SLQSGetDUNCallInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDUNCallInfo_t *reqParam)
- int unpack_wds_SLQSGetDUNCallInfo (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDUNCallInfo_t *pOutput)
- int pack_wds_SLQSGetDataBearerTechnology (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDataBearerTechnology_t *pReqParam)
- int unpack_wds_SLQSGetDataBearerTechnology (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDataBearerTechnology_t *pOutput)
- int pack_wds_SLQSSetIPFamilyPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetIPFamilyPreference_t *pReqParam)
- int unpack_wds_SLQSSetIPFamilyPreference (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetIPFamilyPreference_t *pOutput)
- int pack_wds_SetDefaultProfileNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfileNum_t *pReqParam)
- int unpack_wds_SetDefaultProfileNum (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetDefaultProfileNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfileNum_t *pReqParam)
- int unpack_wds_GetDefaultProfileNum (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDefaultProfileNum_t *pOutput)
- int pack_wds_SLQSSGetDHCPv4ClientConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSGetDHCPv4ClientConfig_t *pReq)
- int unpack_wds_SLQSSGetDHCPv4ClientConfig (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSGetDHCPv4ClientConfig_t *pOutput)
- int pack_wds_GetPacketStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatistics_t *pReq)
- int unpack_wds_GetPacketStatistics (uint8_t *pResp, uint16_t respLen, unpack_wds_GetPacketStatistics_t *pOutput)
- int pack_wds_GetByteTotals (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetByteTotals (uint8_t *pResp, uint16_t respLen, unpack_wds_GetByteTotals_t *pOutput)

- int pack_wds_SLQSGetCurrentChannelRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSGetCurrentChannelRate (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrentChannelRate_t *pOutput)
- int pack_wds_SLQSSGetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSSGetLoopback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSGetLoopback_t *pOutput)
- int pack_wds_SLQSSSetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSSetLoopback_t *reqArg)
- int unpack_wds_SLQSSSetLoopback (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_RMTransferStatistics_ind (uint8_t *pResp, uint16_t respLen, unpack_RMTransferStatistics_ind_t *pOutput)
- int pack_wds_DHCPv4ClientLeaseChange (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_DHCPv4ClientLeaseChange_t *reqArg)
- int unpack_wds_DHCPv4ClientLeaseChange (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_DHCPv4ClientLease_ind (uint8_t *pResp, uint16_t respLen, unpack_wds_DHCPv4ClientLease_ind_t *pOutput)
- int pack_wds_SetMobileIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIP_t *reqArg)
- int unpack_wds_SetMobileIP (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SetMobileIPParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPParameters_t *reqArg)
- int unpack_wds_SetMobileIPParameters (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SetAutoconnect (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetAutoconnect_t *reqArg)
- int unpack_wds_SetAutoconnect (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetAutoconnect (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetAutoconnect (uint8_t *pResp, uint16_t respLen, unpack_wds_GetAutoconnect_t *pOutput)
- int pack_wds_SLQSWdsSetEventReport (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSetEventReport_t *reqArg)
- int unpack_wds_SLQSWdsSetEventReport (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSWdsGoDormant (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSWdsGoDormant (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSWdsGoActive (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSWdsGoActive (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSResetPacketStatics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSResetPacketStatics (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSSSetDHCPv4ClientConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSSetDHCPv4ClientConfig_t *reqArg)
- int unpack_wds_SLQSSSetDHCPv4ClientConfig (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetDataBearerTechnology (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetDataBearerTechnology (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDataBearerTechnology_t *pOutput)

9.62.1 Macro Definition Documentation

9.62.1.1 #define BYT_STAT_STAT_MASK 0X000000C0

9.62.1.2 #define IPV6_ADDRESS_ARRAY_SIZE 8

9.62.1.3 #define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24

9.62.1.4 #define PACK_WDS_IPV4 4

9.62.1.5 #define PACK_WDS_IPV6 6

9.62.1.6 #define WDS_DHCP_MAX_NUM_OPTIONS 30

9.62.1.7 #define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */

9.62.2 Typedef Documentation

9.62.2.1 typedef union unpackWdsProfileParam UnpackQmiProfileInfo

This union WdsProfileParam consists of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

Parameters

SlqsProfile3GPP	3GPP profile
SlqsProfile3GP-P2	3GPP2 profile

9.62.3 Enumeration Type Documentation

9.62.3.1 enum liteQmiDataBearerMasks

Bit mask values to indicate the presence of data bearer information for the current and last data calls

Enumerator

QMI_LITE_WDS_CURRENT_CALL_DB_MASK

QMI_LITE_WDS_LAST_CALL_DB_MASK

9.62.4 Function Documentation

9.62.4.1 int pack_wds_DHCPv4ClientLeaseChange (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_DHCPv4ClientLeaseChange_t * *reqArg*)

DHCPv4 lease state changes pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.2 int pack_wds_GetAutoconnect (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Gets auto connect data session setting pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.3 int pack_wds_GetByteTotals (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get Rx/Tx byte counts since the start of the last packet data session pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.4 int pack_wds_GetConnectionRate (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get connection rate pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: No

9.62.4.5 int pack_wds_GetDataBearerTechnology (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get current data bearer technology pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.6 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	*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

9.62.4.7 int pack_wds_GetDefaultProfileNum (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDefaultProfileNum_t * pReqParam)

get default profile number pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.8 int pack_wds_GetDormancyState (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDormancyState_t * reqParam)

get dormancy state pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: No

9.62.4.9 int pack_wds_GetLastMobileIPError (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetLastMobileIPError_t * pReqParam)

get current data system pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: No

9.62.4.10 int pack_wds_GetMobileIP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetMobileIP_t * *pReqParam*)

get mobile ip mode pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: No

9.62.4.11 int pack_wds_GetMobileIPProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetMobileIPProfile_t * *reqParam*)

get mobile ip profile pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.12 int pack_wds_GetPacketStatistics (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetPacketStatistics_t * *pReq*)

gets current packet transfer counter values pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.13 int pack_wds_GetPacketStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetPacketStatus_t * *reqParam*)

get packet status pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.14 int pack_wds_GetSessionDuration (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetSessionDuration_t * reqParam)

get session duration pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: No

9.62.4.15 int pack_wds_GetSessionState (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

get session state pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.16 int pack_wds_RMSetTransferStatistics (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_RMSetTransferStatistics_t * reqParam)

rm set transfer statistics pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Note

PDN Specific: No

9.62.4.17 int pack_wds_SetAutoconnect (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetAutoconnect_t * *reqArg*)

Auto connect data session parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.18 int pack_wds_SetDefaultProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetDefaultProfile_t * *reqParam*)

set default profile pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.19 int pack_wds_SetDefaultProfileNum (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetDefaultProfileNum_t * *pReqParam*)

set default profile number pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.20 int pack_wds_SetMobileIP (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetMobileIP_t * *reqArg*)

Sets the current mobile IP setting pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.21 int pack_wds_SetMobileIPParameters (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetMobileIPParameters_t * *reqArg*)

Sets the specified mobile IP parameters pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.22 int pack_wds_SetMobileIPProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetMobileIPProfile_t * *reqParam*)

set mobile ip profile pack

Parameters

in,out	*pCtx*	qmi request context
out	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*reqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.23 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.24 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	*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

9.62.4.25 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	*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

9.62.4.26 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.27 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	*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

9.62.4.28 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.29 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	*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

9.62.4.30 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.31 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
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.32 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.33 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

9.62.4.34 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	*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

9.62.4.35 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReqParam*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.36 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
in	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: No

9.62.4.37 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	*pReqBuf*	qmi request buffer
out	*pLen*	qmi request length
in	*pReq*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.38 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	*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

9.62.4.39 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

see qmerrno.h for eQCWWAN_xxx error values

9.62.4.40 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.41 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.42 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.43 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

9.62.4.44 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

9.62.4.45 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	*reqArg*	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.62.4.46 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	*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

9.62.4.47 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

9.62.4.48 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

9.62.4.49 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

9.62.4.50 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

9.62.4.51 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

9.62.4.52 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

9.62.4.53 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

9.62.4.54 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

9.62.4.55 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

9.62.4.56 int unpack_wds_GetLastMobileIPError (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetLastMobileIPError_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

9.62.4.57 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

9.62.4.58 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

9.62.4.59 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

9.62.4.60 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

9.62.4.61 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

9.62.4.62 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

9.62.4.63 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
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.64 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

9.62.4.65 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

9.62.4.66 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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.67 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

9.62.4.68 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

9.62.4.69 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

9.62.4.70 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
out	*pOutput*	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.71 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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.72 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

9.62.4.73 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

9.62.4.74 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

9.62.4.75 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

9.62.4.76 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

9.62.4.77 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

9.62.4.78 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

9.62.4.79 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

9.62.4.80 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
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

9.62.4.81 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

9.62.4.82 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

9.62.4.83 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

9.62.4.84 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

9.62.4.85 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

9.62.4.86 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

9.62.4.87 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

9.62.4.88 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

9.62.4.89 int unpack_wds_SLQSSSetDHCPv4ClientConfig (uint8_t * *pResp*, uint16_t *respLen*)

Gets the DHCP Client V4 Configuration unpack.

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.90 int unpack_wds_SLQSSSetLoopback (uint8_t * *pResp*, uint16_t *respLen*)

Enable/disable Data Loopback Mode and set the value of loopback multiplier unpack

Parameters

in	*pResp*	qmi response from modem
in	*respLen*	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See qmerrno.h for eQCWWAN_xxx error values

9.62.4.91 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

9.62.4.92 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

9.62.4.93 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

9.62.4.94 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

9.62.4.95 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

9.62.4.96 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, 1242
_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, 1235
AAAState
unpack_wds_GetMobileIPProfile_t, 1235
ABSOLUTE_VALIDITY
qaGobiApiSms.h, 1795
ACT_CODE_MAX_SIZE
dms.h, 1410
ALS
getAllCallInformation, 253
voice_getAllCallInformation, 1293
AMSSString
unpack_dms_GetFirmwareRevision_t, 1036
unpack_dms_GetFirmwareRevisions_t, 1037
APNName
unpack_wds_SLQSGetRuntimeSettings_t, 1249
AVMS (SWIAVMS), 59
absoluteValidity
cdmaMsgDecodingParams, 166
accelAcceptReady
qaGobiApiCbk.h, 1559
accelAcceptReady_s, 113
batchPerSec, 113
injectEnable, 113
samplesPerBatch, 113
accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 617
accelTempAcceptReady
qaGobiApiCbk.h, 1559
accelTempAcceptReady_s, 113
batchPerSec, 114
injectEnable, 114
samplesPerBatch, 114
acceleroData
pack_loc_SLQSLOCInjectSensorData_t, 617
acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 617
AccessMac
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763
accolc
pack_nas_SetACCOLC_t, 622
accuracy
pack_pds_SetPDSDefaults_t, 649
ackIndicator
SMSTransferRouteMTMessage, 914
sMSTransferRouteMTMessage, 915
acqOrdeLen
acqOrderPref, 115
nas_acqOrderPref, 415
acqOrderPref, 114
acqOrdeLen, 115
pAcqOrder, 115
acroamsetting

pack_wds_SetAutoconnect_t, 708
slqsautoconnect, 871
acsetting
pack_wds_SetAutoconnect_t, 708
slqsautoconnect, 871
actCode
pack_dms_ActivateAutomatic_t, 590
ActPilotPNElement, 115
ActSetPilotPN, 115
ActSetPilotPNStrength, 115
ActSetCnt
nas_NetworkStat1x, 476
NetworkStat1x, 574
ActSetPilotPN
ActPilotPNElement, 115
nas_ActPilotPNElement, 415
ActSetPilotPNStrength
ActPilotPNElement, 115
nas_ActPilotPNElement, 416
action
pack_nas_InitiateDomainAttach_t, 622
slqsautoconnect, 871
ssdatasession_params, 920
ActivateAutomatic
qaGobiApiDms.h, 1653
activated_ind
_qaQmi3GPP2BroadcastCfgInfo, 71
_qaQmi3GPPBroadcastCfgInfo, 71
sms_qaQmi3GPP2BroadcastCfgInfo, 894
sms_qaQmi3GPPBroadcastCfgInfo, 894
activationState
pack_dms_SetActivationStatusCallback_t, 592
activationStatus
dms_ActivationStatusTlv, 225
ActivationStatusTlv
unpack_dms_SetEventReport_ind_t, 1049
activeBandClass
nas_RFInfoTlv, 495
RFBandInfoElements, 813
activeChannel
nas_RFInfoTlv, 495
RFBandInfoElements, 813
activeInd
messageWaitingInfoContent, 410
sms_messageWaitingInfoContent, 892
ActiveStatus
CLIPResp, 179
CLIRResp, 180
CNAPResp, 182
COLPResp, 183
COLRResp, 183
voice_CLIPResp, 1285
voice_CLIRResp, 1286
voice_CNAPResp, 1286
voice_COLPResp, 1287
voice_COLRResp, 1288
ActiveTechPref
unpack_nas_GetNetworkPreference_t, 1097
AddCDMASysInfo, 115
geoSysIdx, 116
regPrd, 116
AddSysInfo, 116
cellBroadcastCap, 116
geoSysIdx, 116
addr
IPv4Addr, 328
IPv6Addr, 329
unpack_qos_IPv4Addr_t, 1155
unpack_qos_IPv6Addr_t, 1156
address
IPv4Info, 329
IPv6Info, 331
loc_IPv4Info, 367
loc_IPv6Info, 367
loc_urlAddr, 374
unpack_wds_GetMobileIPProfile_t, 1235
urlAddr, 1259
addressSize
unpack_sms_GetSMSCAddress_t, 1177
aid
uim_refreshevent, 983
uim_sessionInformation, 985
uim_UIMSessionInformation, 989
UIMRefreshEvent, 1003
UIMSessionInformation, 1006
aidLength
appStats, 124
appStatus, 127
uim_appStatus, 968
uim_refreshevent, 983
uim_sessionInformation, 985
uim_UIMSessionInformation, 989
UIMRefreshEvent, 1003
UIMSessionInformation, 1006
aidVal
appStats, 124
appStatus, 127
uim_appStatus, 968
aidingIndicatorMask
loc_sensorDataUsage, 372
sensorDataUsage_s, 832
t_sensor, 944
airTimer, 116
airTimerValue, 117
namID, 117
airTimerValue
airTimer, 117
voice_airTimer, 1267
alertPitch
signalInfo, 866
alertingPattern
arrAlertingPattern, 128
voice_arrAlertingPattern, 1270
AlertingType
arrAlertingType, 129
voice_arrAlertingType, 1271

alertmsg
omaDmConfigTlv, 580
omaDmConfigTlvExt, 583
unpack_omaDmConfigTlv_t, 1140
alertmsglength
omaDmConfigTlv, 580
omaDmConfigTlvExt, 583
unpack_omaDmConfigTlv_t, 1140
allCallsAlphaIDInfo, 117
AlphaIDInfo, 117
callID, 117
allCallsAlphaIDInfoArr
arrAlphaID, 129
voice_arrAlphaID, 1271
allCallsDiagInfo, 117
callID, 118
DiagInfo, 118
AllCallsUUSInfo
arrUUSInfo, 135
voice_arrUUSInfo, 1277
allCallsUUSInfo, 118
callID, 118
uusInfo, 118
alphaDcs
alphaIDInfo, 119
voice_alphaIDInfo, 1269
AlphaID
CatAlPhaIdentifierTlv, 158
AlphaIDInfo
allCallsAlphaIDInfo, 117
voice_allCallsAlphaIDInfo, 1267
alphaIDInfo, 118
alphaDcs, 119
alphaLen, 119
alphaText, 119
alphaIDLen
SMSAsyncRawSend_s, 901
AlphaIDLength
CatAlPhaIdentifierTlv, 158
alphaLen
alphaIDInfo, 119
voice_alphaIDInfo, 1269
alphaText
alphaIDInfo, 119
voice_alphaIDInfo, 1269
alphabet
wcdmaMsgEncodingParams, 1360
altSrcInfo_t, 120
coverage, 121
linkage, 121
source, 121
Altitude
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
altitudeAssumed
unpack_loc_GnssSvInfo_Ind_t, 1081
altitudeSrcInfo, 119
coverage, 120
linkage, 120
pack_loc_SLQSLOCInjectPosition_t, 615
source, 120
altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 615
altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 615
ambr_dl
sApnExtraParams, 825
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1161
ambr_dl_ext
sApnExtraParams, 825
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1161
ambr_dl_ext2
sApnExtraParams, 825
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1161
ambr_ul
sApnExtraParams, 825
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1161
ambr_ul_ext
sApnExtraParams, 825
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1161
ambr_ul_ext2
sApnExtraParams, 825
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1161
amssSize
unpack_dms_GetFirmwareRevision_t, 1036
unpack_dms_GetFirmwareRevisions_t, 1037
AnswerUSSD
qaGobiApiVoice.h, 1903
apdoxypages.c, 1399
apnId
pack_qos_SLQSQosSwiReadApnExtraParams_t, 658
pack_qos_SLQSQosSwiReadDataStats_t, 659
sApnExtraParams, 825
sQosStat, 917
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1161
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
apnName
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1257
apnname
unpack_wds_GetDefaultProfile_t, 1232
apnsize
unpack_wds_GetDefaultProfile_t, 1232
appNameLength
loc_LocApplicationInfo, 368
LocApplicationInfo, 375
appProviderLength
loc_LocApplicationInfo, 368

LocApplicationInfo, 375
appState
appStats, 124
appStatus, 127
uim_appStatus, 968
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, 868
slotInfo, 870
uim_slotInfo, 988
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, 968
appVersionLength
loc_LocApplicationInfo, 368
LocApplicationInfo, 375
appVersionValid
loc_LocApplicationInfo, 368
LocApplicationInfo, 375
Application
unpack_nas_GetCDMANetworkParameters_t, 1093
appversion_str
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
arfcn
GERANInfo, 251
gsmCellInfo, 291
nas_GERANInfo, 440
nas_gsmCellInfo, 441
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, 1340
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, 809
uim_refreshevent, 983
uim_registerRefresh, 984
UIMRefreshEvent, 1003
AtCmdPort
DcsUsbPortNames, 217
Audio Service (AUDIO), 53
auth

unpack_wds_GetDefaultProfile_t, 1232
authData
pack_uim_SLQSUIMAuthenticate_t, 678
UIMAuthenticateReq, 991
AuthProt
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763
authenticateResult, 135
content, 136
contentLen, 136
Authentication
unpack_wds_SLQSGetRuntimeSettings_t, 1250
authentication
pack_wds_SetDefaultProfile_t, 710
authenticationData, 136
context, 136
data, 137
dataLen, 137
Autosdm
unpack_swioma_SLQSOMADMGetSettings_t, 1192
avgPeriod
LTESigRptCfg, 401
LTESigRptConfig, 402
nas_LTESigRptCfg, 465
nas_LTESigRptConfig, 466
azimuth
loc_satelliteInfo, 371
satelliteInfo, 827

bAPNLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bAltitudeAssumed
gnssSvInfoNotification, 284
bAuto
pack_pds_SetPortAutomaticTracking_t, 650
pack_pds_SetServiceAutomaticTracking_t, 650
bBinaryType
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
bEnable
pack_dms_UIMSetPINProtection_t, 603
pack_nas_SLQSSetSignalStrengthsCallback_t, 640
bEnabled
pack_pds_SetXTRAAutomaticDownload_t, 651
bForceDownload
pack_fms_SetImagesPreference_t, 606
bICCID
slot_t, 867
UIMSlotStatus, 1008
bICCIDLength
slot_t, 867
UIMSlotStatus, 1008
bLogicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 683
slot_t, 867
UIMSlotStatus, 1008
UIMSwitchSlotReq, 1010
bManSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bModelSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bNameSize
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
bNotification
_QmiSwiAvmsEventReportNotificationResp, 77
bNumberOfPhySlots
UIMSlotStatusChangeInfo, 1009
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 1196
BOOL
SwiDataTypes.h, 2007
BPTlv
NASQmiCbkNasSystemSelPrefInd, 560
bPWDLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bPackageID
_QmiSwiAvmsEventReportPackageIDResp, 78
bPlasmaIDSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bRegStatus
_QmiSwiAvmsEventReportRegStatusResp, 78
bResetStatistics
rmTrasnferStaticsReq, 814
swiRMTrasnferStaticsReq, 942
bSWVerSize
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
bSerity
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
bState
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
bType
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
_QmiSwiAvmsEventReportSessionTypeResp, 79
BUILD_ID_LEN
qaGobiApiFms.h, 1686
BUILD_ID_MAX_LEN
qaGobiApiFms.h, 1686
bUnameLength
_QmiSwiAvmsSetSettingsAPNInfo, 80
_SLQSAVMSSettingsAPNInfo, 86
bUserInputRequest
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74

_QmiSwiAvmsEventReportConnectionRequest-Resp, 75
bVersionSize
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
BYT_STAT_STAT_MASK
wds.h, 2069
BYTE
SwiDataTypes.h, 2007
band
LTEInfo, 393
nas_LTEInfo, 457
band1900
gsmCellInfo, 291
nas_gsmCellInfo, 442
band_pref
NASBandPreferenceTlv, 529
BandCapability
unpack_dms_GetBandCapability_t, 1025
bandCapability
BandCapabilityResp, 140
unpack_dms_SLQSGetBandCapability_t, 1056
BandCapabilityResp, 137
bandCapability, 140
pLteBandCapability, 140
pTdsBandCapability, 140
bandPref
pack_nas_SLQSSetBandPreference_t, 639
bandwidth
LTEInfo, 393
nas_LTEInfo, 457
baseId
CDMAInfo, 164
CDMASysInfo, 173
nas_CDMAInfo, 420
nas_CDMASysInfo, 424
baseLat
CDMAInfo, 164
CDMASysInfo, 173
nas_CDMAInfo, 420
nas_CDMASysInfo, 424
baseLong
CDMAInfo, 164
CDMASysInfo, 173
nas_CDMAInfo, 420
nas_CDMASysInfo, 424
BasestationID
qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
BasestationLatitude
qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
BasestationLongitude
qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
batchPerSec
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 296
gyroTempAcceptReady_s, 297
BdsSV, 140
id, 140
mask, 140
BdsSVInfo, 140
len, 141
pSV, 141
BearerID
unpack_qos_QosFlowInfo_t, 1159
bearerID
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1252
bearerId
sQosFlowStat, 916
unpack_QosFlowStat_t, 1175
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1257
BinaryUpdateSessionInfo
_SwiAvmsEventInd, 99
bootSize
unpack_dms_GetFirmwareRevisions_t, 1037
BootString
unpack_dms_GetFirmwareRevisions_t, 1037
bootversion_str
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
Broadcast
unpack_nas_GetCDMANetworkParameters_t, 1093
broadcastActivate
pack_sms_SLQSSetSmsBroadcastActivation_t, 667
BroadcastConfig, 141
fromServiceId, 142
selected, 142
toServiceId, 142
broadcastConfig
_qaQmi3GPPBroadcastCfgInfo, 71
sms_qaQmi3GPPBroadcastCfgInfo, 894
bsInfoValid
CDMASysInfo, 173
nas_CDMASysInfo, 424
bsPRev
CDMASysInfo, 173
nas_CDMASysInfo, 424
bsPRevValid
CDMASysInfo, 173
nas_CDMASysInfo, 424
bsic
GERANInfo, 251
nas_GERANInfo, 440
bsicId
gsmCellInfo, 291
nas_gsmCellInfo, 442
bucketSz
tokenBucket, 958

unpack_qos_tokenBucket_t, 1173
buildID
CurrImageInfo, 200
FMSImageIdElement, 247
image_info_t, 308
ImageIdElement, 309
buildIDLen
CurrImageInfo, 200
image_info_t, 308
buildIDLength
FMSImageIdElement, 247
ImageIdElement, 310
buildId
FMSImageElement, 246
ImageElement, 309
buildIdLength
FMSImageElement, 247
ImageElement, 309
BurstDTMFInfo
pack_voice_SLQSVoiceBurstDTMF_t, 689
voiceBurstDTMFInfo, 1305
burstDTMFInfo, 142
digitCnt, 142
pCallID, 142
pDigitBuff, 142
ByteLoopbackMode
unpack_wds_SLQSSGetLoopback_t, 1255
WDSGetLoopbackData, 1390
ByteLoopbackMultiplier
unpack_wds_SLQSSGetLoopback_t, 1255
WDSGetLoopbackData, 1390
ByteTotalsElmntsV4
WdsByteTotals, 1381
ByteTotalsElmntsV6
WdsByteTotals, 1381

CATEventDataType, 159
eventMask, 159
pErrorMask, 159
CATSendEnvelopeCommand
qaGobiApiCat.h, 1548
CATSendTerminalResponse
qaGobiApiCat.h, 1548
CBK_DISABLE_EVENT
qaGobiApiCbk.h, 1557
CBK_ENABLE_EVENT
qaGobiApiCbk.h, 1557
CBK_NOCHANGE
qaGobiApiCbk.h, 1557
CCETlv
QmiCbkCatEventStatusReportInd, 769
CDMA_P_Rev
qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
CDMABroadcastConfig, 161
_qaQmi3GPP2BroadcastCfgInfo, 71
language, 162
selected, 162
serviceCategory, 162
sms_qaQmi3GPP2BroadcastCfgInfo, 894
CDMAChannel, 162
priChA, 162
priChB, 162
secChA, 162
secChB, 163
CDMAECIOThresh, 163
CDMAECIOThreshListLen, 163
pCDMAECIOThreshList, 163
CDMAECIOThreshListLen
CDMAECIOThresh, 163
nas_CDMAECIOThresh, 419
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, 421
CDMASSInfo, 169
ecio, 170
rssi, 170
unpack_nas_SLQSNasGetSigInfo_t, 1127
CDMASysInfo, 170
baseId, 173
baseLat, 173
baseLong, 173
bsInfoValid, 173
bsPRev, 173
bsPRevValid, 173
ccsSupported, 173
ccsSupportedValid, 173
cdmaSysIdValid, 174
isSysPrlMatch, 174
isSysPrlMatchValid, 174
MCC, 174
MNC, 174
networkID, 174
networkIdValid, 174
pRevInUse, 174
pRevInUseValid, 174
packetZone, 174
packetZoneValid, 174
sysInfoCDMA, 174
systemID, 174
CDMASysInfoExt, 174
imsi_11_12, 174
MCC, 174
CDMASystemInfoExt
qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
CHAR

SwiDataTypes.h, 2007
CK_MAX_SIZE
dms.h, 1410
CLIPResp, 178
ActiveStatus, 179
ProvisionStatus, 179
CLIRResp, 179
ActiveStatus, 180
ProvisionStatus, 180
CNAPResp, 181
ActiveStatus, 182
ProvisionStatus, 182
COLPResp, 182
ActiveStatus, 183
ProvisionStatus, 183
COLRResp, 183
ActiveStatus, 183
ProvisionStatus, 183
CONFIG_LEN
qaGobiApiSms.h, 1795
CQIValueCW0
LteCQIParm, 389
unpack_nas_SLQSSwiGetLteCQI_t, 1135
CQIValueCW1
LteCQIParm, 389
unpack_nas_SLQSSwiGetLteCQI_t, 1135
CSDomain
unpack_nas_GetServingNetwork_t, 1100
CSGID, 193
id, 194
mcc, 194
mnc, 194
mncPcsDigits, 194
rat, 194
CUGIndex
CUGInfo, 195
voice_CUGInfo, 1290
CUGInfo, 194
CUGIndex, 195
SuppOA, 195
SuppPrefCUG, 195
CallBacK registration (CBK), 41
CallBarStatus
qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
callBarStatus, 143
csBarStatus, 144
psBarStatus, 144
CallBarringSysInfo, 142
csBarStatus, 143
psBarStatus, 143
callDuration
unpack_wds_GetSessionDuration_t, 1239
CallEndReason
DUNCallInfoInd, 233
callEndReason
arrCallEndReason, 131
unpack_wds_SLQSGetDUNCallInfo_t, 1247
voice_arrCallEndReason, 1272
CallFWExtInfo
getCallFWExtInfo, 260
voice_getCallFWExtInfo, 1294
callFWExtInfo, 148
noReplyTimer, 150
numLen, 150
numPlan, 150
numType, 150
number, 150
PI, 150
SI, 150
SvcClass, 150
SvcStatus, 150
CallFWInfo
getCallFWInfo, 261
voice_getCallFWInfo, 1295
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, 232
getAllCallRmtPtyName, 254
getAllCallRmtPtyNum, 254
pack_voice_SLQSVoiceGetCallInfo_t, 693
pack_voice_SLQSVoiceStopContDTMF_t, 704
peerNumberInfo, 736
unpack_voice_SLQSVoiceStopContDTMF_t, 1227
voice_allCallsAlphaIDInfo, 1267
voice_allCallsDiagInfo, 1268
voice_allCallsUUSInfo, 1268
voice_arrAlertingPattern, 1270
voice_arrAlertingType, 1271
voice_arrCallEndReason, 1272
voice_arrSvcOption, 1276
voice_callInfo, 1283
voice_getAllCallRmtPtyName, 1293
voice_getAllCallRmtPtyNum, 1294
voice_peerNumberInfo, 1297
voiceCallInfoReq, 1305
voiceInfoRec, 1334
voiceOTASPStatusInfo, 1337
voicePrivacyInfo, 1338
voiceStopContDTMFinfo, 1350

voiceSUPSNotification, 1354
callInfo, 151
callID, 152
callState, 152
callType, 153
direction, 153
mode, 153
callNumber
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
callState
callInfo, 152
voice_callInfo, 1283
callType
callInfo, 153
voice_callInfo, 1284
calledPartyInfo, 144
numLen, 145
numPlan, 145
numType, 145
number, 145
PI, 146
SI, 146
CalledPartyNum
arrCalledPartyNum, 130
voice_arrCalledPartyNum, 1272
calledPartySubAdd, 146
extBit, 146
oddEvenInd, 146
subAddr, 146
subAddrLen, 146
subAddrType, 146
callerID
callerIDInfo, 147
connectNumInfo, 188
voice_connectNumInfo, 1289
callerIDInfo, 147
callerID, 147
callerIDLen, 147
PI, 147
callerIDLen
callerIDInfo, 147
connectNumInfo, 188
voice_connectNumInfo, 1289
callerName
remotePartyName, 810
voice_remotePartyName, 1299
CallInfo
getAllCallInformation, 253
voice_getAllCallInformation, 1293
callingPartyInfo, 153
numLen, 154
numPlan, 154
numType, 154
number, 154
PI, 154
SI, 154
CancelUSSD
qaGobiApiVoice.h, 1903
Card Application Toolkit (CAT), 44
cardResult, 154
sw1, 155
sw2, 155
cardState
slotInf, 868
slotInfo, 870
uim_slotInfo, 988
cardStatus, 155
index1xPri, 156
index1xSec, 156
indexGwPri, 156
indexGwSec, 156
numSlot, 156
SlotInfo, 156
Carrier
fwinfo_s, 250
carrier
CurrentImgList, 198
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1059
carrier_str
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
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, 197
CatCommonEventTlv, 158
CatEvent, 158
EventID, 158
EventLength, 158
TlvPresent, 158
CatEndPS
currentCatEvent, 197
CatEndProactiveSessionTlv, 158
EndProactiveSession, 159
CatEvIDData
currentCatEvent, 197
CatEvent
CatCommonEventTlv, 158
CatEventIDDataTlv, 159
Data, 159
DataLength, 159
ReferenceID, 159
CatEventListTlv, 159
SetupEventList, 160
CatEventLst

currentCatEvent, 197
CatRefresh
currentCatEvent, 197
CatRefreshTlv, 160
RefreshMode, 160
RefreshStage, 160
causeCode
SMSAsyncRawSend_s, 901
ccSUPSType, 160
reason, 161
svcType, 161
ccsSupported
CDMASysInfo, 173
nas_CDMASysInfo, 424
ccsSupportedValid
CDMASysInfo, 173
nas_CDMASysInfo, 424
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, 174
nas_CDMASysInfo, 424
cell_resel_priority
infoInterFreq, 327
nas_infoInterFreq, 452
cellBroadcastCap
AddSysInfo, 116
nas_AddSysInfo, 417
CellDb, 175
mask, 175
CellID
qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
cellID
GERANInfo, 252
nas_GERANInfo, 440
nas_UMTSInfo, 518
UMTSInfo, 1013
cellId
GSMSysInfo, 295
LTESysInfo, 407
nas_GSMSysInfo, 445
nas_LTESysInfo, 470
nas_WCDMASysInfo, 527
WCDMASysInfo, 1363
cellIdValid
gsmCellInfo, 291
GSMSysInfo, 295
LTESysInfo, 407
nas_gsmCellInfo, 442
nas_GSMSysInfo, 445
nas_LTESysInfo, 470
nas_WCDMASysInfo, 527
WCDMASysInfo, 1364
cellInterFreqParams
infoInterFreq, 327
nas_infoInterFreq, 452
cellIsTDD
nas_umtsLTENbrCell, 520
umtsLTENbrCell, 1015
CellParams
LTEInfoIntrafreq, 395
nas_LTEInfoIntrafreq, 460
cellParams, 175
pci, 176
rsrp, 176
rsrq, 176
rssi, 176
srxlev, 176
cellReselPriority
lteGsmCellInfo, 391
LTEInfoIntrafreq, 395
lteWcdmaCellInfo, 409
nas_lteGsmCellInfo, 455
nas_LTEInfoIntrafreq, 460
nas_lteWcdmaCellInfo, 471
cells_len
infoInterFreq, 327
lteGsmCellInfo, 391
nas_infoInterFreq, 452
nas_lteGsmCellInfo, 455
cellsLen
LTEInfoIntrafreq, 395
lteWcdmaCellInfo, 409

nas_LTEInfoIntrafreq, 460
nas_lteWcdmaCellInfo, 471
chaddr
wds_DHCPv4HWConfig, 1370
WdsDHCPv4HWConfig, 1386
wdsDhcpv4HwConfig, 1386
chaddrLen
wds_DHCPv4HWConfig, 1370
WdsDHCPv4HWConfig, 1386
wdsDhcpv4HwConfig, 1386
changePIN
pack_uim_ChangePin_t, 675
UIMChangePinReq, 992
changeUIMPIN, 176
oldPINLen, 177
oldPINVal, 177
pinID, 177
pinLen, 177
pinVal, 177
ChannelRate, 178
CurrChanRxRate, 178
CurrChanTxRate, 178
DUNCallInfoInd, 233
MaxChanRxRate, 178
MaxChanTxRate, 178
channelRate, 177
CurrChanRxRate, 177
CurrChanTxRate, 177
unpack_wds_SLQSGetDUNCallInfo_t, 1247
Chipset
DeviceConfigDetail, 221
nas_DeviceConfigDetail, 435
ckLen
depersonalizationInformation, 219
uim_depersonalizationInformation, 974
ckVal
depersonalizationInformation, 219
uim_depersonalizationInformation, 974
clear
pack_dms_SLQSSwiGetCrashInfo_t, 598
ClkInfo, 180
mask, 181
codingScheme
nas_PLMNNetworkNameData, 489
PLMNNetworkNameData, 746
remotePartyName, 810
voice_remotePartyName, 1300
CommInfo, 183
imsRegState, 185
modemMode, 185
psState, 185
systemMode, 185
temperature, 185
common.h
eCAT, 1402
eCTL, 1402
eDMS, 1402
eIMSA, 1402
eIND, 1403
eLOC, 1402
eLOG_DEBUG, 1401
eLOG_FATAL, 1402
eLOG_INFO, 1401
eLOG_WARN, 1401
eNAS, 1402
ePDS, 1402
eQOS, 1402
eREQ, 1403
eRMS, 1402
eRSP, 1403
eSAR, 1402
eSMS, 1402
eSWILOC, 1402
eSWIM2MCMD, 1402
eSWIOMA, 1402
eSWIOMAEXT, 1402
eTIMEOUT_10_S, 1402
eTIMEOUT_20_S, 1402
eTIMEOUT_2_S, 1402
eTIMEOUT_300_S, 1402
eTIMEOUT_30_S, 1402
eTIMEOUT_5_S, 1402
eTIMEOUT_60_S, 1402
eTIMEOUT_8_S, 1402
eTIMEOUT_DEFAULT, 1402
eTMD, 1402
eUIM, 1402
eVOICE, 1402
eWDS, 1402
common.h, 1399
eLOG_LEVEL, 1401
eQMI_SVC, 1402
eTimeout, 1402
fill_pack_ctx, 1403
fill_sdu_hdr, 1403
get_version, 1403
glog, 1404
gloglvl, 1404
helper_get_resp_ctx, 1403
helper_get_xid, 1403
helper_set_log_func, 1403
helper_set_log_lvl, 1403
liteqmi_GetVersion, 1403
liteqmi_helper_decode7bitAsciiEncString, 1403
liteqmi_log, 1404
logger, 1401
MINREQBKLEN, 1401
MSGID_AND_LEN, 1401
MSGID_DONT_CARE, 1401
msgtype, 1402
SDU_HDR_LEN, 1401
UNUSEDPARAM, 1401
unpack_result_code_only, 1404
commonInfo
swiModemStatusResp, 929
unpack_nas_SLQSNasSwiModemStatus_t, 1131

ConcSvcInfo
unpack_nas_SLQSGetServingSystem_t, 1113
concSvcInfo
qaQmiServingSystemParam, 768
confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 619
Config
_SwiAvmsEventInd, 99
conn_status
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1252
ConnRateElmntsV4
WdsConnectionRate, 1383
ConnRateElmntsV6
WdsConnectionRate, 1383
connStatus
_packetSrvStatus, 70
connectNumInfo, 187
callerID, 188
callerIDLen, 188
numPlan, 188
numPresInd, 188
numType, 188
screeningInd, 188
ConnectedPartyNum
arrConnectPartyNum, 132
voice_arrConnectPartyNum, 1273
ConnectionRequest
_SwiAvmsEventInd, 99
ConnectionStatus, 185
MDMCallDuration, 186
MDMConnStatus, 186
connectionStatus, 186
MDMCallDuration, 186
MDMConnStatus, 186
unpack_wds_GetSessionState_t, 1240
unpack_wds_SLQSGetDUNCallInfo_t, 1247
connetionState
imsaPdpStatusInfo, 313
content
authenticateResult, 136
readResult, 806
uim_authenticateResult, 969
uim_readResult, 981
contentLen
authenticateResult, 136
readResult, 806
uim_authenticateResult, 969
uim_readResult, 981
context
authenticationData, 136
uim_authenticationData, 970
contextId
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 728
swiPDPRuntimeSettingsReq, 931
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1257
contextType
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 728
swiPDPRuntimeSettingsReq, 931
ControlMac
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763
Count1
RankIndicatorInd, 805
Count2
RankIndicatorInd, 805
countryInitials
nas_PLMNNetworkNameData, 489
PLMNNetworkNameData, 746
coverage
altitudeSrcInfo, 120
altSrcInfo_t, 121
cpich_ecno
nas_wcdmaCellInfo, 522
wcdmaCellInfo, 1354
cpich_rscp
nas_wcdmaCellInfo, 522
wcdmaCellInfo, 1354
cradleMountConfigStatus
QmiCbkLocCradleMountInd, 776
crashAction
pack_dms_SetCrashAction_t, 592
crashData
CrashInfo, 189
crashInformation, 191
crashId
CrashInfo, 189
crashInformation, 191
CrashInfo, 188
crashData, 189
crashId, 189
crashStrLen, 189
gcDumpStrLen, 189
numCrashes, 189
pCrashString, 189
pGCDumpString, 189
crashInfo
crashInfoParams, 190
crashInfoParam
unpack_dms_SLQSSwiGetCrashInfo_t, 1057
CrashInfoParams, 190
pCrashInfo, 190
pDevCrashStatus, 190
crashInfoParams, 189
crashInfo, 190
crashStatus, 190
crashInformation, 190
crashData, 191
crashId, 191
crashString, 191
crashStrlen, 191
gcdumpString, 191
gcdumpStrlen, 191

numCrashes, 191
crashStatus
crashInfoParams, 190
crashStrLen
CrashInfo, 189
crashString
crashInformation, 191
crashStrlen
crashInformation, 191
CreateProfileIn, 191
curProfile, 192
pProfileID, 192
pProfileType, 192
CreateProfileOut, 192
pExtErrorCode, 193
pProfileIndex, 193
pProfileType, 193
csAttachState
nas_servSystem, 504
NASServingSystemInfo, 562
ServingSystemInfo, 835
servSystem, 837
csBarStatus
CallBarringSysInfo, 143
callBarStatus, 144
nas_CallBarringSysInfo, 417
nas_callBarStatus, 418
CsgId, 193
csgId, 193
TlvPresent, 193
csgId
CsgId, 193
cur_carr_name
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
cur_carr_rev
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
curAMRConfig, 195
gsmAmrStat, 196
wcdmaAmrStat, 196
curDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 1245
curProfile
_GetProfileSettingOut, 65
CreateProfileIn, 192
ModifyProfileIn, 413
pack_wds_SLQSModifyProfile_t, 720
UnPackGetProfileSettingOut, 1258
CurrChanRxRate
ChannelRate, 178
channelRate, 177
dunchannelRate, 234
CurrChanTxRate
ChannelRate, 178
channelRate, 177
dunchannelRate, 234
currDBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
CurrDataSysStat, 196
pCurrNetworkInfo, 197
pNetworkInfoLen, 197
pPrefNetwork, 197
CurrImageInfo, 199
buildID, 200
buildIDLen, 200
imageType, 200
uniqueID, 200
currNWInfo
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
CurrNetworkInfo, 200
NetworkType, 201
RATMask, 202
SOMask, 202
currNetworkInfo, 202
NetworkType, 203
RATMask, 203
SOMask, 203
unpack_wds_SLQSGetCurrDataSystemStat_t, 1243
current_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1244
WDSSWICurrentChannelRates, 1398
current_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1244
WDSSWICurrentChannelRates, 1398
currentCatEvent, 197
CatAlphaIdtfr, 197
CatEndPS, 197
CatEvIDData, 197
CatEventLst, 197
CatRefresh, 197
currentChannelRXRate
unpack_wds_GetConnectionRate_t, 1229
currentChannelTXRate
unpack_wds_GetConnectionRate_t, 1229
currentDataBearer
pack_wds_SLQSSetWdsEventCallback_t, 722
CurrentImgList, 197
carrier, 198
fwvers, 198
numEntries, 198
pCurrImgInfo, 198
pkgver, 198
priver, 198
currentMitigationLvl
QmiCbkTmdMitiLvlRptInd, 791
currentNetwork
dataBearerTechnology, 213
qmiWDSDataBearerTechnology, 796
CurrentPLMN

qaQmiServingSystemParam, 768
unpack_nas_SLQSGetServingSystem_t, 1113
currentPLMN, 198
MCC, 199
MNC, 199
netDescr, 199
netDescrLength, 199
cust_attr
custSettingInfo, 208
DMScustSettingInfo, 227
cust_id
custSettingInfo, 208
DMScustSettingInfo, 227
DMSgetCustomInput, 228
getCustomInput, 262
pack_dms_GetCustFeaturesV2_t, 591
pack_dms_SetCustFeaturesV2_t, 595
setCustomSettingV2, 844
cust_value
custSettingInfo, 208
DMScustSettingInfo, 227
pack_dms_SetCustFeaturesV2_t, 595
setCustomSettingV2, 844
custFeaturesInfo, 203
GpsEnable, 205
pDHCPRelayEnabled, 205
pDisableIMSI, 205
pGPSLPM, 205
pGPSSel, 205
pIPFamSupport, 205
pIsVoiceEnabled, 205
pRMAutoConnect, 205
pSMSSupport, 205
qaGobiApiDms.h, 1646
custFeaturesSetting, 206
pDHCPRelayEnabled, 207
pGPSEnable, 207
pGPSLPM, 207
pGPSSel, 207
pIsVoiceEnabled, 207
qaGobiApiDms.h, 1648
custSetting
custSettingList, 208
DMScustSettingList, 227
custSettingInfo, 207
cust_attr, 208
cust_id, 208
cust_value, 208
id_length, 208
value_length, 208
custSettingList, 208
custSetting, 208
list_type, 208
num_instances, 209
CustomSCP
unpack_nas_GetCDMANetworkParameters_t, 1093
CwtMute
GetM2MAudioProfileResp, 275
GetM2MAVMuteResp, 277

DEVICE_STATE_BOOT
qaGobiApiCbk.h, 1598
DEVICE_STATE_DISCONNECTED
qaGobiApiCbk.h, 1598
DEVICE_STATE_READY
qaGobiApiCbk.h, 1598
DEVICE_STATE_SERIAL_ONLY
qaGobiApiCbk.h, 1598
dBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
dBTechExtRatValue
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
dBTechExtSoMask
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
dBTechnology
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
dBtechExtAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
dBtechnologyExt
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
DEFAULTBYTEVALUE
qaGobiApiPds.h, 1769
DEFAULTLONGVALUE
qaGobiApiPds.h, 1769
DEFAULTWORDVALUE
qaGobiApiPds.h, 1769
DEREGISTER_EVENT
qaGobiApiCbk.h, 1557
DEREGISTER_SRV
qaGobiApiCbk.h, 1557
DEVICE_OFFLINE
qaGobiApiFms.h, 1686
DEVICE_RESET
qaGobiApiFms.h, 1686
DEVICE_SHUTDOWN
qaGobiApiFms.h, 1686
DHCPOption, 222
optCode, 222
optValLen, 222
pOptVal, 222
DHCPOptionList, 222
numOpt, 223
pOptions, 223
DHCPRelayEnabled
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
DHCPv4LeaseOptTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1228
DHCPv4LeaseStateTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1228

dLatitude
unpack_pds_SetEventReport_Ind_t, 1148
dLongitude
unpack_pds_SetEventReport_Ind_t, 1148
DMS_IMGDETAILS_LEN
dms.h, 1410
DMS_PM_FACTORY
dms.h, 1410
DMS_PM_LOW
dms.h, 1411
DMS_PM_OFFLINE
dms.h, 1411
DMS_PM_ONLINE
dms.h, 1411
DMS_PM_RESET
dms.h, 1411
DMS_PM_SHUT_DOWN
dms.h, 1411
DMScustSettingInfo, 226
cust_attr, 227
cust_id, 227
cust_value, 227
id_length, 227
value_length, 227
DMScustSettingList, 227
custSetting, 227
list_type, 227
num_instances, 227
DMSgetCustomFeatureV2, 227
pCustSettingInfo, 228
pCustSettingList, 228
pGetCustomInput, 228
DMSgetCustomInput, 228
cust_id, 228
list_type, 228
DRCCover
DRCParams, 231
nas_DRCParams, 436
DRCParams, 231
DRCCover, 231
DRCValue, 231
DRCValue
DRCParams, 231
nas_DRCParams, 436
DTMFEvent
DTMFInfo, 232
DTMFInfo, 231
callID, 232
DTMFEvent, 232
digitBuff, 232
digitCnt, 232
DTMFInformation
voiceDTMFEventInfo, 1312
DTMFInterdigitInterval
DTMFLengths, 233
voice_DTMFLengths, 1292
DTMFLengths, 232
DTMFInterdigitInterval, 233
DTMFPulseWidth, 233
DTMFPulseWidth
DTMFLengths, 233
voice_DTMFLengths, 1292
DTMFdigit
pack_voice_SLQSVoiceStartContDTMF_t, 704
voiceContDTMFinfo, 1311
DTMInd
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1113
DUNCallInfoInd, 233
CallEndReason, 233
ChannelRate, 233
DataBearerTech, 233
DormancyStatus, 233
MdmConnStatus, 233
RXOKBytesCount, 233
TXOKBytesCount, 233
Data
CatEventIDDataTlv, 159
data
authenticationData, 137
SMSCAddress, 902
sMSCAddress, 902
SMSEtwsMessage, 904
sMSEtwsMessage, 903
SMSTransferRouteMTMessage, 914
sMSTransferRouteMTMessage, 915
SwiOTAMsg_s, 930
uim_authenticationData, 970
data_buf
NASOTAMessageTlv, 551
data_len
NASOTAMessageTlv, 551
SwiOTAMsg_s, 930
dataBearer
pack_wds_SLQSSetWdsEventCallback_t, 723
dataBearerMask
dataBearers, 209
unpack_wds_SLQSGetDataBearerTechnology_t, 1245
DataBearerTech, 209
DUNCallInfoInd, 233
ratValue, 211
soMask, 211
techType, 211
dataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1247
DataBearerTechExt, 211
pBearerTech, 212
pLastBearerTech, 212
dataBearerTechExt
pack_wds_SLQSSetWdsEventCallback_t, 723
dataBearerTechnology, 212
currentNetwork, 213
ratMask, 213
soMask, 213
dataBearers, 209

dataBearerMask, 209
pCurDataBearerTechnology, 209
pLastCallDataBearerTechnology, 209
dataCapabilities
dataSrvCapabilities, 214
nas_dataSrvCapabilities, 431
dataCapabilitiesLen
dataSrvCapabilities, 214
nas_dataSrvCapabilities, 431
DataCaps
unpack_nas_GetServingNetwork_t, 1100
unpack_nas_GetServingNetworkCapabilities_t, 1101
dataCaps
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1103
DataCapsLen
unpack_nas_GetServingNetwork_t, 1100
unpack_nas_GetServingNetworkCapabilities_t, 1101
dataCapsSize
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1103
dataLen
authenticationData, 137
uim_authenticationData, 970
DataLength
CatEventIDDataTlv, 159
DataRate
unpack_qos_swiQosFlow_t, 1172
dataRate, 213
dataRateMax, 214
guaranteedRate, 214
dataRateMax
dataRate, 214
unpack_qos_dataRate_t, 1155
dataServiceCaCapability
unpack_dms_GetDeviceCapabilities_t, 1031
DataServiceCapability
unpack_dms_GetDeviceCap_t, 1030
DataSessionStatus
_SwiAvmsEventInd, 99
DataSrvCapabilities
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1113
dataSrvCapabilities, 214
dataCapabilities, 214
dataCapabilitiesLen, 214
DataStatusDetail, 215
IPAddress, 216
LastErrCode, 216
dataSysStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1253
dataSystemStatus
pack_wds_SLQSSetWdsEventCallback_t, 723
DataULongLongTlv, 217
TlvPresent, 217
ullData, 217
DataULongTlv, 217
TlvPresent, 217
ulData, 217
Date
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
wcdmaLongMsgDecodingParams, 1357
wcdmaMsgDecodingParams, 1359
DateLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
day
nas_timeInfo, 516
nas_UniversalTime, 522
timeInfo, 954
UniversalTime, 1023
dayLtSavingAdj
nas_timeInfo, 516
timeInfo, 954
dayOfWeek
nas_timeInfo, 516
nas_UniversalTime, 522
timeInfo, 954
UniversalTime, 1023
daylightSavings
nas_qaQmi3Gpp2TimeZone, 491
qaQmi3Gpp2TimeZone, 764
DcsUsbPortNames, 217
AtCmdPort, 217
DmPort, 217
NmeaPort, 217
defaultPDNEnabled
unpack_wds_SLQSGet3GPPConfigItem_t, 1242
DefaultRoamInd
unpack_nas_SLQSGetServingSystem_t, 1113
defaultRoamInd
qaQmiServingSystemParam, 769
delAssistDataStatus, 217
status, 218
delayClass
GPRSQoS, 285
GPRSRequestedQoS, 286
LibPackGPRSRequestedQoS, 333
wds_GPRSQoS, 1373
DeleteStoredImage
qaGobiApiFms.h, 1689
deliveryErrSDU
LibPackUMTSQoS, 360
UMTSMinQoS, 1017
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
depersonalizationInformation, 218
ckLen, 219
ckVal, 219
feature, 219
operation, 219
depersonilisationInfo

pack_uim_SLQSUIMDepersonalization_t, 678
UIMDepersonalizationReq, 993
Description
SlqsNas3GppNetworkInfo, 874
description
omaDmFotaTlv, 584
omaDmFotaTlvExt, 587
unpack_omaDmFotaTlv_t, 1142
descriptionlength
omaDmFotaTlv, 584
omaDmFotaTlvExt, 587
unpack_omaDmFotaTlv_t, 1142
Desription
nas_QmiNas3GppNetworkInfo, 492
destPortRangeEnd
LibPackTFTIDParams, 357
TFTIDParams, 952
destPortRangeStart
LibPackTFTIDParams, 357
TFTIDParams, 952
detailSvcInfo, 219
hdrHybrid, 220
hdrSrvStatus, 220
isSysForbidden, 220
srvCapability, 220
srvStatus, 221
DetailedSvcInfo
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1113
dev
qmifwinfo_s, 792
DevCrashState
unpack_dms_GetCrashAction_t, 1026
Device
SetM2MAudioAVCFGReq, 850
Device Connectivity Service (DCS), 37
Device Management Service (DMS), 39
device_state_enum
qaGobiApiCbk.h, 1598
DeviceConfigDetail, 221
Chipset, 221
HWVersion, 221
QLIC, 222
Technology, 222
deviceID
_MitigationDevInfo, 67
deviceIdLen
_MitigationDevInfo, 67
DiagInfo
allCallsDiagInfo, 118
arrDiagInfo, 132
voice_allCallsDiagInfo, 1268
voice_arrDiagInfo, 1274
diagInfo, 223
diagInfoLen, 223
diagnosticInfo, 223
diagInfoLen
diagInfo, 223
voice_diagInfo, 1291
diagnosticInfo
diagInfo, 223
voice_diagInfo, 1291
digitBuff
DTMFInfo, 232
digitCnt
burstDTMFInfo, 142
DTMFInfo, 232
voice_burstDTMFInfo, 1277
dirNum, 223
dirNum, 224
dirNumLen, 224
dirNum, 224
nas_dirNum, 435
dirNumLen
dirNum, 224
nas_dirNum, 435
direction
callInfo, 153
voice_callInfo, 1284
DisableIMSI
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
dispType
extDispRecInfo, 240
displayCondition
nas_serviceProviderName, 503
serviceProviderName, 834
dl_bw_value
nas_PhyCaAggPcellInfo, 481
nas_PhyCaAggScellDlBw, 482
nas_PhyCaAggScellInfo, 486
NASPhyCaAggPcellInfo, 551
NASPhyCaAggScellDlBw, 552
NASPhyCaAggScellInfo, 554
PhyCaAggPcellInfo, 738
PhyCaAggScellDlBw, 738
PhyCaAggScellInfo, 742
DmPort
DcsUsbPortNames, 217
dms.h, 1404
ACT_CODE_MAX_SIZE, 1410
CK_MAX_SIZE, 1410
DMS_IMGDETAILS_LEN, 1410
DMS_PM_FACTORY, 1410
DMS_PM_LOW, 1411
DMS_PM_OFFLINE, 1411
DMS_PM_ONLINE, 1411
DMS_PM_RESET, 1411
DMS_PM_SHUT_DOWN, 1411
ERI_DATA_MAX_SIZE, 1411
MAX_BUILD_ID_LEN, 1411
MEID_MAX_SIZE, 1411
pack_dms_ActivateAutomatic, 1412
pack_dms_GetActivationState, 1412
pack_dms_GetBandCapability, 1412
pack_dms_GetCrashAction, 1413

pack_dms_GetCustFeature, 1413
pack_dms_GetCustFeaturesV2, 1413
pack_dms_GetDeviceCap, 1414
pack_dms_GetDeviceCapabilities, 1414
pack_dms_GetDeviceHardwareRev, 1414
pack_dms_GetDeviceMfr, 1415
pack_dms_GetDeviceSerialNumbers, 1415
pack_dms_GetFSN, 1417
pack_dms_GetFirmwareInfo, 1416
pack_dms_GetFirmwareRevision, 1416
pack_dms_GetFirmwareRevisions, 1416
pack_dms_GetHardwareRevision, 1417
pack_dms_GetIMSI, 1417
pack_dms_GetManufacturer, 1418
pack_dms_GetModelID, 1418
pack_dms_GetNetworkTime, 1418
pack_dms_GetOfflineReason, 1419
pack_dms_GetPRLVersion, 1419
pack_dms_GetPower, 1419
pack_dms_GetSerialNumbers, 1420
pack_dms_GetUSBComp, 1420
pack_dms_GetVoiceNumber, 1420
pack_dms_ResetToFactoryDefaults, 1421
pack_dms_SLQSDmsSwiGetResetInfo, 1424
pack_dms_SLQSDmsSwiIndicationRegister, 1424
pack_dms_SLQSGetBandCapability, 1425
pack_dms_SLQSGetERIFile, 1425
pack_dms_SLQSSwiClearDyingGaspStatistics, 1426
pack_dms_SLQSSwiGetCrashInfo, 1426
pack_dms_SLQSSwiGetDyingGaspCfg, 1426
pack_dms_SLQSSwiGetDyingGaspStatistics, 1427
pack_dms_SLQSSwiGetFirmwareCurr, 1427
pack_dms_SLQSSwiGetFwUpdateStatus, 1427
pack_dms_SLQSSwiGetHostDevInfo, 1428
pack_dms_SLQSSwiGetOSInfo, 1428
pack_dms_SLQSSwiGetSerialNoExt, 1428
pack_dms_SLQSSwiSetDyingGaspCfg, 1429
pack_dms_SLQSSwiSetHostDevInfo, 1429
pack_dms_SLQSSwiSetOSInfo, 1429
pack_dms_SLQSUIMGetState, 1430
pack_dms_SetActivationStatusCallback, 1421
pack_dms_SetCrashAction, 1422
pack_dms_SetCustFeature, 1422
pack_dms_SetCustFeaturesV2, 1422
pack_dms_SetEventReport, 1423
pack_dms_SetFirmwarePreference, 1423
pack_dms_SetPower, 1423
pack_dms_SetUSBComp, 1424
pack_dms_UIMChangePIN, 1430
pack_dms_UIMGetControlKeyStatus, 1430
pack_dms_UIMGetICCID, 1431
pack_dms_UIMGetPINStatus, 1431
pack_dms_UIMSetControlKeyProtection, 1432
pack_dms_UIMSetPINProtection, 1432
pack_dms_UIMUnblockControlKey, 1432
pack_dms_UIMUnblockPIN, 1433
pack_dms_UIMVerifyPIN, 1433
pack_dms_ValidateSPC, 1433
SPC_SIZE, 1411
UNIQUE_ID_LEN, 1411
unpack_dms_ActivateAutomatic, 1434
unpack_dms_GetActivationState, 1434
unpack_dms_GetBandCapability, 1435
unpack_dms_GetCrashAction, 1435
unpack_dms_GetCustFeature, 1435
unpack_dms_GetCustFeaturesV2, 1436
unpack_dms_GetDeviceCap, 1436
unpack_dms_GetDeviceCapabilities, 1436
unpack_dms_GetDeviceHardwareRev, 1437
unpack_dms_GetDeviceMfr, 1437
unpack_dms_GetDeviceSerialNumbers, 1437
unpack_dms_GetFSN, 1439
unpack_dms_GetFirmwareInfo, 1438
unpack_dms_GetFirmwareRevision, 1438
unpack_dms_GetFirmwareRevisions, 1438
unpack_dms_GetHardwareRevision, 1439
unpack_dms_GetIMSI, 1439
unpack_dms_GetManufacturer, 1440
unpack_dms_GetModelID, 1440
unpack_dms_GetNetworkTime, 1440
unpack_dms_GetOfflineReason, 1441
unpack_dms_GetPRLVersion, 1441
unpack_dms_GetPower, 1441
unpack_dms_GetSerialNumbers, 1442
unpack_dms_GetUSBComp, 1442
unpack_dms_GetVoiceNumber, 1442
unpack_dms_ResetToFactoryDefaults, 1443
unpack_dms_SLQSDmsSwiGetResetInfo, 1446
unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 1446
unpack_dms_SLQSDmsSwiIndicationRegister, 1447
unpack_dms_SLQSGetBandCapability, 1447
unpack_dms_SLQSGetERIFile, 1448
unpack_dms_SLQSSwiClearDyingGaspStatistics, 1448
unpack_dms_SLQSSwiGetCrashInfo, 1448
unpack_dms_SLQSSwiGetDyingGaspCfg, 1449
unpack_dms_SLQSSwiGetDyingGaspStatistics, 1449
unpack_dms_SLQSSwiGetFirmwareCurr, 1449
unpack_dms_SLQSSwiGetFwUpdateStatus, 1450
unpack_dms_SLQSSwiGetHostDevInfo, 1450
unpack_dms_SLQSSwiGetOSInfo, 1450
unpack_dms_SLQSSwiGetSerialNoExt, 1451
unpack_dms_SLQSSwiSetDyingGaspCfg, 1451
unpack_dms_SLQSSwiSetHostDevInfo, 1451
unpack_dms_SLQSSwiSetOSInfo, 1452
unpack_dms_SLQSUIMGetState, 1452
unpack_dms_SetActivationStatusCallback, 1443
unpack_dms_SetCrashAction, 1443
unpack_dms_SetCustFeature, 1444
unpack_dms_SetCustFeaturesV2, 1444
unpack_dms_SetEventReport, 1444

unpack_dms_SetEventReport_ind, 1445
unpack_dms_SetFirmwarePreference, 1445
unpack_dms_SetPower, 1445
unpack_dms_SetUSBComp, 1446
unpack_dms_UIMChangePIN, 1452
unpack_dms_UIMGetControlKeyStatus, 1453
unpack_dms_UIMGetICCID, 1453
unpack_dms_UIMGetPINStatus, 1453
unpack_dms_UIMSetControlKeyProtection, 1454
unpack_dms_UIMSetPINProtection, 1454
unpack_dms_UIMUnblockControlKey, 1454
unpack_dms_UIMUnblockPIN, 1455
unpack_dms_UIMVerifyPIN, 1455
unpack_dms_ValidateSPC, 1456
dms_ActivationStatusTlv, 224
activationStatus, 225
TlvPresent, 225
dms_OperatingModeTlv, 225
operatingMode, 225
TlvPresent, 225
dmsCurrentPRLInfo, 225
pPRLPreference, 226
pPRLVersion, 226
qaGobiApiDms.h, 1649
dmsIndicationRegisterReq, 229
pSwiGetResetInd, 229
dmsSwiGetResetInfo, 229
source, 230
type, 230
Domain, 230
domainLen, 230
domainName, 230
domain
DomainNameList, 231
wds_DomainNameList, 1372
domainLen
Domain, 230
wds_Domain, 1372
DomainList
unpack_wds_SLQSGetRuntimeSettings_t, 1250
domainName
Domain, 230
wds_Domain, 1372
DomainNameList, 230
domain, 231
numInstances, 231
dormancyStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
dormancyState
unpack_wds_GetDormancyState_t, 1233
DormancyStatus
DUNCallInfoInd, 233
dormancyStatus
pack_wds_SLQSSetWdsEventCallback_t, 723
unpack_wds_SLQSGetDUNCallInfo_t, 1247
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
downLink
NSSAudioCtrl, 579
DownloadToSlot
qaGobiApiFms.h, 1689
dscp
QosMap, 805
dtmSupp
GSMSysInfo, 295
nas_GSMSysInfo, 445
dtmSuppValid
GSMSysInfo, 295
nas_GSMSysInfo, 445
dunchannelRate, 234
CurrChanRxRate, 234
CurrChanTxRate, 234
MaxChanRxRate, 234
MaxChanTxRate, 234
Duration
pack_nas_SetNetworkPreference_t, 624
unpack_nas_GetNetworkPreference_t, 1097

eCAT
common.h, 1402
eCTL
common.h, 1402
eDMS
common.h, 1402
eGOBI_DEV_SERIES_9X07_9X50
qaGobiApiFms.h, 1687
eGOBI_DEV_SERIES_9X15
qaGobiApiFms.h, 1687
eGOBI_DEV_SERIES_9X30
qaGobiApiFms.h, 1687
eGOBI_DEV_SERIES_G3K
qaGobiApiFms.h, 1687
eGOBI_DEV_SERIES_NON_GOBI
qaGobiApiFms.h, 1687
eGOBI_DEV_SERIES_SIERRA_GOBI
qaGobiApiFms.h, 1687
eGOBI_DEV_SERIES_UNKNOWN
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_3
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_AERIS
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_ALLTEL
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_AMX_TELCEL
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_ATT
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_BELL
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_BHARTI
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_BRASIL_VIVO
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_CHINA_MOBILE
qaGobiApiFms.h, 1688

eGOBI_IMG_CAR_CHINA_TELECOM
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_CHINA_UNICOM
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_EMOBILE
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_FACTORY
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_GENERIC
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_GENERIC_CDMA
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_IUSACELL
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_KDDI
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_KT_FREETEL
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_LEAP
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_METROPCS
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_NETCOM
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_NORF
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_NTT_DOCOMO
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_O2
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_OMH
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_ORANGE
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_RELIANCE1
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_RELIANCE2
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_ROGERS
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_SFR
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_SINGTEL_OPTUS
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_SK_TELCOM1
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_SK_TELCOM2
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_SOFTBANK
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_SPRINT
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_SWISSCOM
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_TATA
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_TELCOM_ITALIA
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_TELCOM_NZ
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_TELEFONICA
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_TELENOR
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_TELIASONERA
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_TELSTRA1
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_TELSTRA2
qaGobiApiFms.h, 1688
eGOBI_IMG_CAR_TELUS
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_TMOBILE
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_US
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_VERIZON
qaGobiApiFms.h, 1687
eGOBI_IMG_CAR_VODAFONE
qaGobiApiFms.h, 1687
eGOBI_IMG_GPS_ASSISTED
qaGobiApiFms.h, 1688
eGOBI_IMG_GPS_NO_XTRA
qaGobiApiFms.h, 1688
eGOBI_IMG_GPS_NONE
qaGobiApiFms.h, 1688
eGOBI_IMG_GPS_STAND_ALONE
qaGobiApiFms.h, 1688
eGOBI_IMG_REG_ASIA
qaGobiApiFms.h, 1688
eGOBI_IMG_REG_AUS
qaGobiApiFms.h, 1688
eGOBI_IMG_REG_EU
qaGobiApiFms.h, 1688
eGOBI_IMG_REG_GLOBAL
qaGobiApiFms.h, 1688
eGOBI_IMG_REG_LA
qaGobiApiFms.h, 1688
eGOBI_IMG_REG_NA
qaGobiApiFms.h, 1688
eGOBI_IMG_TECH_CDMA
qaGobiApiFms.h, 1689
eGOBI_IMG_TECH_UMTS
qaGobiApiFms.h, 1689
eGobi_DEV_SERIES_MC83
qaGobiApiFms.h, 1687
eIMSA
common.h, 1402
eIND
common.h, 1403
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100
nas.h, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15
nas.h, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25
nas.h, 1484

eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50
nas.h, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6
nas.h, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75
nas.h, 1484
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
nas.h, 1484
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
nas.h, 1484
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
nas.h, 1484
eLOC
common.h, 1402
eLOG_DEBUG
common.h, 1401
eLOG_FATAL
common.h, 1402
eLOG_INFO
common.h, 1401
eLOG_WARN
common.h, 1401
eNAS
common.h, 1402
eNAS_LTE_CPHY_CA_BW_NRB_100
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_CA_BW_NRB_15
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_CA_BW_NRB_25
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_CA_BW_NRB_50
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_CA_BW_NRB_6
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_CA_BW_NRB_75
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
nas.h, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
nas.h, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
nas.h, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
nas.h, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
nas.h, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
nas.h, 1484
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE
nas.h, 1485
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
nas.h, 1485
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
qaGobiApiNas.h, 1736
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
nas.h, 1485
eNAS_RADIO_IF_GSM
qaGobiApiNas.h, 1735
eNAS_RADIO_IF_LTE
qaGobiApiNas.h, 1735
eNAS_RADIO_IF_TDSCDMA
qaGobiApiNas.h, 1735
eNAS_RADIO_IF_UMTS
qaGobiApiNas.h, 1735
ePDS
common.h, 1402
eQA_QMI_SVC_NA
qaGobiApiCbk.h, 1598
eQA_QMI_SVC_NAS
qaGobiApiCbk.h, 1598
eQA_QMI_SVC_WDS
qaGobiApiCbk.h, 1598
eQCWWAN_ERR_API_MUTEX_TIMEOUT
qmerrno.h, 1968
eQCWWAN_ERR_BUFFER_SZ
qmerrno.h, 1967
eQCWWAN_ERR_CANCEL_OP
qmerrno.h, 1968
eQCWWAN_ERR_DRIVER
qmerrno.h, 1968
eQCWWAN_ERR_ENUM_BEGIN
qmerrno.h, 1967
eQCWWAN_ERR_ENUM_END
qmerrno.h, 1968
eQCWWAN_ERR_FILE_COPY
qmerrno.h, 1967
eQCWWAN_ERR_FILE_OPEN
qmerrno.h, 1967
eQCWWAN_ERR_GENERAL
qmerrno.h, 1967
eQCWWAN_ERR_INTERNAL
qmerrno.h, 1967
eQCWWAN_ERR_INVALID_ARG
qmerrno.h, 1967
eQCWWAN_ERR_INVALID_DEVID
qmerrno.h, 1967
eQCWWAN_ERR_INVALID_FILE
qmerrno.h, 1967
eQCWWAN_ERR_INVALID_QMI_RSP
qmerrno.h, 1967
eQCWWAN_ERR_INVALID_XID
qmerrno.h, 1968
eQCWWAN_ERR_MALFORMED_QMI_RSP

qmerrno.h, 1967
eQCWWAN_ERR_MEMORY
qmerrno.h, 1967
eQCWWAN_ERR_MULTIPLE_DEVICES
qmerrno.h, 1968
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
qmerrno.h, 1968
eQCWWAN_ERR_NO_CANCELABLE_OP
qmerrno.h, 1968
eQCWWAN_ERR_NO_CONNECTION
qmerrno.h, 1967
eQCWWAN_ERR_NO_DEVICE
qmerrno.h, 1967
eQCWWAN_ERR_NO_SIGNAL
qmerrno.h, 1968
eQCWWAN_ERR_NONE
qmerrno.h, 1967
eQCWWAN_ERR_NULL_TLV
qmerrno.h, 1971
eQCWWAN_ERR_OFFLINE
qmerrno.h, 1968
eQCWWAN_ERR_PDU_GENERATION
qmerrno.h, 1968
eQCWWAN_ERR_QMI_ABORTED
qmerrno.h, 1968
eQCWWAN_ERR_QMI_ACCESS_DENIED
qmerrno.h, 1970
eQCWWAN_ERR_QMI_ACK_NOT_SENT
qmerrno.h, 1970
eQCWWAN_ERR_QMI_ARG_TOO_LONG
qmerrno.h, 1968
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
qmerrno.h, 1969
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
qmerrno.h, 1969
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
qmerrno.h, 1970
eQCWWAN_ERR_QMI_CALL_FAILED
qmerrno.h, 1968
eQCWWAN_ERR_QMI_CARD_BUSY_RSP
qmerrno.h, 1971
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
qmerrno.h, 1970
eQCWWAN_ERR_QMI_CAT_END
qmerrno.h, 1971
eQCWWAN_ERR_QMI_CAT_START
qmerrno.h, 1971
eQCWWAN_ERR_QMI_CAUSE_CODE
qmerrno.h, 1969
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
qmerrno.h, 1968
eQCWWAN_ERR_QMI_CONNECT
qmerrno.h, 1967
eQCWWAN_ERR_QMI_DEVICE_IN_USE
qmerrno.h, 1968
eQCWWAN_ERR_QMI_DEVICE_NOT_READY
qmerrno.h, 1969
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
qmerrno.h, 1969
eQCWWAN_ERR_QMI_DISABLED
qmerrno.h, 1969
eQCWWAN_ERR_QMI_ENCODING
qmerrno.h, 1969
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
qmerrno.h, 1971
eQCWWAN_ERR_QMI_EVENT_REG_FAILED
qmerrno.h, 1971
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
qmerrno.h, 1970
eQCWWAN_ERR_QMI_FDN_RESTRICT
qmerrno.h, 1970
eQCWWAN_ERR_QMI_FLOW_SUSPENDED
qmerrno.h, 1969
eQCWWAN_ERR_QMI_GENERAL
qmerrno.h, 1969
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
qmerrno.h, 1970
eQCWWAN_ERR_QMI_IFACE
qmerrno.h, 1967
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
qmerrno.h, 1970
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INCORRECT_PIN
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
qmerrno.h, 1970
eQCWWAN_ERR_QMI_INJECT_TIMEOUT
qmerrno.h, 1970
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INTERNAL
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_ARG
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
qmerrno.h, 1971
eQCWWAN_ERR_QMI_INVALID_HANDLE
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_ID
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_INDEX
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
qmerrno.h, 1969

eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_OPERATION
qmerrno.h, 1970
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_PINID
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_PROFILE
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_QMI_CMD
qmerrno.h, 1970
eQCWWAN_ERR_QMI_INVALID_QOS_ID
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_TECH_PREF
qmerrno.h, 1968
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
qmerrno.h, 1971
eQCWWAN_ERR_QMI_INVALID_TRANSITION
qmerrno.h, 1969
eQCWWAN_ERR_QMI_INVALID_TX_ID
qmerrno.h, 1968
eQCWWAN_ERR_QMI_MALFORMED_MSG
qmerrno.h, 1968
eQCWWAN_ERR_QMI_MAX
qmerrno.h, 1970
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE
qmerrno.h, 1969
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE
qmerrno.h, 1969
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE
qmerrno.h, 1969
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
qmerrno.h, 1969
eQCWWAN_ERR_QMI_MISSING_ARG
qmerrno.h, 1968
eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, 1970
eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, 1970
eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, 1969
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, 1969
eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, 1968
eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, 1969
eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, 1968
eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, 1968
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
qmerrno.h, 1968
eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, 1970
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, 1970
eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, 1968
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, 1969
eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, 1968
eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, 1970
eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, 1968
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED
qmerrno.h, 1968
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED
qmerrno.h, 1968
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, 1970
eQCWWAN_ERR_QMI_OUT_OF_CALL
qmerrno.h, 1968
eQCWWAN_ERR_QMI_PIN_BLOCKED
qmerrno.h, 1969
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
qmerrno.h, 1969
eQCWWAN_ERR_QMI_POLICY_MISMATCH
qmerrno.h, 1970
eQCWWAN_ERR_QMI_REQ
qmerrno.h, 1967
eQCWWAN_ERR_QMI_REQ_SCH
qmerrno.h, 1967
eQCWWAN_ERR_QMI_REQ_TO
qmerrno.h, 1967
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
qmerrno.h, 1969
eQCWWAN_ERR_QMI_RSP
qmerrno.h, 1967
eQCWWAN_ERR_QMI_RSP_TO
qmerrno.h, 1967
eQCWWAN_ERR_QMI_SEGMENT_ORDER
qmerrno.h, 1970
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
qmerrno.h, 1970
eQCWWAN_ERR_QMI_SESSION_INACTIVE
qmerrno.h, 1969

eQCWWAN_ERR_QMI_SESSION_INVALID
qmerrno.h, 1969
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
qmerrno.h, 1969
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
qmerrno.h, 1970
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
qmerrno.h, 1969
eQCWWAN_ERR_QMI_SMSC_ADDR
qmerrno.h, 1970
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
qmerrno.h, 1970
eQCWWAN_ERR_QMI_TPDU_TYPE
qmerrno.h, 1970
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
qmerrno.h, 1968
eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, 1969
eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, 1971
eQCWWAN_ERR_RESET
qmerrno.h, 1968
eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_END
qmerrno.h, 1971
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_START
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, 1970
eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, 1970
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
qmerrno.h, 1971
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
qmerrno.h, 1971
eQCWWAN_ERR_SWIDCS_END
qmerrno.h, 1971
eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, 1971
eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, 1971
eQCWWAN_ERR_SWIDCS_START
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_END
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, 1971
eQCWWAN_ERR_SWIIM_START

qmerrno.h, 1971
eQCWWAN_ERR_SWISM_END
qmerrno.h, 1971
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND
qmerrno.h, 1971
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, 1971
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, 1971
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED
qmerrno.h, 1971
eQCWWAN_ERR_SWISMS_START
qmerrno.h, 1971
eQMI_LOC_SESS_STATUS_FAILURE
loc.h, 1465
eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, 1465
eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, 1465
eQMI_LOC_SESS_STATUS_TIMEOUT
loc.h, 1465
eQOS
common.h, 1402
eREQ
common.h, 1403
eRMS
common.h, 1402
eRSP
common.h, 1403
eSAR
common.h, 1402
eSMS
common.h, 1402
eSWILOC
common.h, 1402
eSWIM2MCMD
common.h, 1402
eSWIOMA
common.h, 1402
eSWIOMAEXT
common.h, 1402
eSYS_SRV_DOMAIN_CAMPED
qaGobiApiNas.h, 1735
eSYS_SRV_DOMAIN_CS_ONLY
qaGobiApiNas.h, 1735
eSYS_SRV_DOMAIN_CS_PS
qaGobiApiNas.h, 1735
eSYS_SRV_DOMAIN_NO_SRV
qaGobiApiNas.h, 1735
eSYS_SRV_DOMAIN_PS_ONLY
qaGobiApiNas.h, 1735
eSYS_SRV_DOMAIN_UNKNOWN
qaGobiApiNas.h, 1735
eSetServiceAutomaticTrackingDisable
qaGobiApiPds.h, 1769
eSetServiceAutomaticTrackingEnable
qaGobiApiPds.h, 1769
eTIMEOUT_10_S
common.h, 1402
eTIMEOUT_20_S
common.h, 1402
eTIMEOUT_2_S
common.h, 1402
eTIMEOUT_300_S
common.h, 1402
eTIMEOUT_30_S
common.h, 1402
eTIMEOUT_5_S
common.h, 1402
eTIMEOUT_60_S
common.h, 1402
eTIMEOUT_8_S
common.h, 1402
eTIMEOUT_DEFAULT
common.h, 1402
eTLV_3GPP_NETWORK_INFO
qaNasPerformNetworkScan.h, 1965
eTLV_CBK_ALPHA_IDENTIFIER
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_DISPLAY_TEXT
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_END_PROACTIVE_SESSION
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_GET_IN_KEY
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_GET_INPUT
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_LANGUAGE_NOTIFICATION
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_REFRESH
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_SELECT_ITEM
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_SETUP_EVENT_LIST
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_SETUP_IDLE_MODE_TEXT
qaCbkCatEventReportInd.h, 1542
eTLV_CBK_SETUP_MENU
qaCbkCatEventReportInd.h, 1542
eTLV_END_PROACTIVE_SESSION_LENGTH
qaCbkCatEventReportInd.h, 1542
eTLV_IND_OMA_DM_CONFIG
qaCbkSwiOmaDmEventReportInd.h, 1543
eTLV_IND_OMA_DM_FOTA
qaCbkSwiOmaDmEventReportInd.h, 1543
eTLV_IND_OMA_DM_NOT
qaCbkSwiOmaDmEventReportInd.h, 1543
eTLV_REFRESH_LENGTH
qaCbkCatEventReportInd.h, 1542
eTLV_RF_BAND_INFO
qaNasGetRFBandInfo.h, 1963
eTLV_SETUP_EVENT_LIST_LENGTH
qaCbkCatEventReportInd.h, 1542
eTMD

common.h, 1402
eUIM
common.h, 1402
eVOICE
common.h, 1402
eWDS
common.h, 1402
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_END
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_FAIL
qmerrno.h, 1972
eWDS_ERR_PROFILE_REG_RESULT_LIST_END
qmerrno.h, 1972
ECIOThresListLen
ECIOThresh, 235
nas_ECIOThresh, 437
ECIOThresh, 235
ECIOThresListLen, 235
pECIOThresList, 235
ECTCallState
ECTNum, 236
ECTNum, 236
ECTCallState, 236
number, 236
presentationInd, 236
eDevState
qaGobiApiCbk.h, 1560
eDevice
sGetDeviceSeriesResult, 863
eGetDeviceSeries
qaGobiApiFms.h, 1690
eGobiDeviceSeries
qaGobiApiFms.h, 1687
eGobiImageCarrier
qaGobiApiFms.h, 1687
eGobiImageGPS
qaGobiApiFms.h, 1688
eGobiImageRegion
qaGobiApiFms.h, 1688
eGobiImageTech
qaGobiApiFms.h, 1688
eLOG_LEVEL
common.h, 1401
EMTlv
NASQmiCbkNasSystemSelPrefInd, 560
eQCWWANError
qmerrno.h, 1967
eQMI_SVC
common.h, 1402
eQMISARRFState
qaGobiApiSar.h, 1791
eQaQMIService
qaGobiApiCbk.h, 1598
ERI_DATA_MAX_SIZE
dms.h, 1411
ERIFileparams, 237
pFile, 238
pFileSize, 238
qaGobiApiDms.h, 1650
eSMSEventType
qaGobiApiCbk.h, 1560
ESNString
unpack_dms_GetDeviceSerialNumbers_t, 1034
eSYS_SRV_DOMAIN
qaGobiApiNas.h, 1735
ETWSPLMNInfo
eTWSPLMNInfoTlv, 239

eTWSPLMNInfoTlv, 239
ETWSPLMNInfo, 239
TlvPresent, 239
ETWSPLMNTlv
unpack_sms_SetNewSMSCallback_ind_t, 1179
ETWSTlv
unpack_sms_SetNewSMSCallback_ind_t, 1179
eTimeout
common.h, 1402
EVENT_MASK_CARD
qaGobiApiCbk.h, 1557
eValid
LibPackTFTIDParams, 357
TFTIDParams, 952
EarMute
GetAudioProfileResp, 258
GetM2MAudioProfileResp, 275
GetM2MAVMuteResp, 277
SetAudioProfileReq, 842
SetM2MAVMuteReq, 853
earfcn
infoInterFreq, 327
LTEInfoIntrafreq, 395
ltePCI, 398
nas_infoInterFreq, 452
nas_LTEInfoIntrafreq, 460
nas_ltePCI, 462
nas_umtsLTENbrCell, 520
umtsLTENbrCell, 1015
earfcn0
lteEARFCN, 389
nas_lteEARFCN, 454
earfcn1
lteEARFCN, 390
nas_lteEARFCN, 454
ecio
CDMASSInfo, 170
cdmaSSInfo, 170
ecioListElement, 235
HDRSSInfo, 303
hdrSSInfo, 304
nas_ecioListElement, 436
nas_rxInfo, 499
nas_UMTSInfo, 519
rxInfo, 820
TDSCDMASigInfoExt, 948
tdscdmaSigInfoExt, 947
UMTSInfo, 1013
ecioDelta
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
ecioInfo
nas_SLQSSignalStrengthsInformation, 508
SLQSSignalStrengthsInformation, 884
ecioList
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
ecioListElement, 234
ecio, 235
radioIf, 235
ecioListLen
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
ecioThresholdList
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
ecioThresholdListLen
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
egprsSupp
GSMSysInfo, 295
nas_GSMSysInfo, 446
egprsSuppValid
GSMSysInfo, 295
nas_GSMSysInfo, 446
elevation
loc_satelliteInfo, 371
satelliteInfo, 827
EmerMode
NASEmergencyModeTlv, 531
emmConnState
LTEInfo, 393
nas_LTEInfo, 457
emmState
LTEInfo, 393
nas_LTEInfo, 457
emmSubState
LTEInfo, 393
nas_LTEInfo, 457
Enable
SetM2MAudioLPBKReq, 851
enable
pack_pds_SetPDSState_t, 649
pack_qos_SLQSSetQosEventCallback_t, 659
enabled
pack_rms_SetSMSWake_t, 659
unpack_rms_GetSMSWake_t, 1175
unpack_wds_GetMobileIPProfile_t, 1235
EnabledStatus
unpack_pds_SetPdsState_Ind_t, 1149
EncryptProt
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763
EncryptedPIN1
pack_uim_ChangePin_t, 675
pack_uim_SetPinProtection_t, 677
pack_uim_UnblockPin_t, 684
encryptedPIN1, 236
pin1Len, 237
pin1Val, 237
EndProactiveSession
CatEndProactiveSessionTlv, 159
EngineState
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
engineState

QmiCbkLocEngineStateInd, 776
unpack_loc_EngineState_Ind_t, 1078
entries
t_Sv, 945
eqmiCbkSetStatus
sms.h, 1989
eriData
eriDataparams, 237
eriDataLen
eriDataparams, 237
eriDataparams, 237
eriData, 237
eriDataLen, 237
eriFile
unpack_dms_SLQSGetERIFile_t, 1056
error
unpack_wds_GetLastMobileIPError_t, 1233
errorClass
SMSAsyncRawSend_s, 901
errorRate
errorRateListElement, 239
nas_errorRateListElement, 438
errorRateInfo
nas_SLQSSignalStrengthsInformation, 508
SLQSSignalStrengthsInformation, 885
errorRateList
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
errorRateListElement, 238
errorRate, 239
radioIf, 239
errorRateListLen
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
errorState
slotInf, 868
slotInfo, 870
uim_slotInfo, 988
esn
unpack_dms_GetSerialNumbers_t, 1044
esnSize
serialNumbersInfo, 833
unpack_dms_GetDeviceSerialNumbers_t, 1034
EspSpi
unpack_qos_swiQosFilter_t, 1167
EtwsMessageInfo
sMSEtwsMessageTlv, 904
event
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 1164
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1165
event_Index
QmiCbkCatEventStatusReportInd, 769
EventID
CatCommonEventTlv, 158
EventLength
CatCommonEventTlv, 158
eventMask
CATEventDataType, 159
pack_uim_SLQSUIMEventRegister_t, 678
UIMEventRegisterReqResp, 994
unpack_uim_SLQSUIMEventRegister_t, 1197
eventRegister
LOCEventRegisterReqResp, 379
pack_loc_EventRegister_t, 609
eventType
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1187
evrcCapability
prefVoiceSO, 750
voice_prefVoiceSO, 1299
executingImage
FMSImageIDEntries, 248
ImageIDEntries, 310
exponent
pktErrRate, 743
unpack_qos_pktErrRate_t, 1157
extBit
calledPartySubAdd, 146
voice_calledPartySubAdd, 1278
extDispInfo
extDispRecInfo, 240
extDispInfoLen
extDispRecInfo, 240
extDispRecInfo, 239
dispType, 240
extDispInfo, 240
extDispInfoLen, 240
ExtErrorCode
PackCreateProfileOut, 729
extPowerState
LOCExtPowerStateReqResp, 379
pack_loc_SetExtPowerState_t, 609
extendedErrorCode
unpack_wds_SLQSDeleteProfile_t, 1241

FIRST_INSTANCE
qaGobiApiCbk.h, 1557
FLOAT
SwiDataTypes.h, 2007
FMSImageElement, 246
buildId, 246
buildIdLength, 247
imageId, 247
imageType, 247
FMSImageIDEntries, 248
executingImage, 248
imageIDElement, 248
imageIDSize, 248
imageType, 248
maxImages, 248
FMSImageIdElement, 247
buildID, 247
buildIDLength, 247
failureCount, 247
imageID, 247

storageIndex, 247
FMSImageList, 248
imageIDEntries, 249
listSize, 249
FMSPrefImageList, 249
listEntries, 249
listSize, 249
FORBIDDEN_INDEX
qaNasPerformNetworkScan.h, 1964
FOTAUpdate
_SLQSOMADMSettingsReqParams, 94
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 673
unpack_swioma_SLQSOMADMGetSettings_t, 1192
FOTAdownload
_SLQSOMADMSettingsReqParams, 93
_SLQSOMADMSettingsReqParams3, 94
pack_swioma_SLQSOMADMSetSettings_t, 673
unpack_swioma_SLQSOMADMGetSettings_t, 1192
FSNumber
FactorySequenceNumber, 240
facility
pack_dms_UIMGetControlKeyStatus_t, 601
pack_dms_UIMSetControlKeyProtection_t, 602
pack_dms_UIMUnblockControlKey_t, 603
facilityCk
pack_dms_UIMSetControlKeyProtection_t, 602
pack_dms_UIMUnblockControlKey_t, 603
facilityState
pack_dms_UIMSetControlKeyProtection_t, 602
unpack_dms_UIMGetControlKeyStatus_t, 1065
FactorySequenceNumber, 240
FSNumber, 240
failureCount
FMSImageIdElement, 247
ImageIdElement, 310
failureReason
ssdatasession_params, 920
failureReasonv4
ssdatasession_params, 920
failureReasonv6
ssdatasession_params, 920
family
pack_wds_GetDefaultProfileNum_t, 705
pack_wds_SetDefaultProfileNum_t, 711
feature
depersonalizationInformation, 219
personalizationStatus, 737
uim_depersonalizationInformation, 974
uim_personalizationStatus, 980
fileAttributes, 240
fileID, 243
fileSize, 243
fileType, 243
rawLen, 243
rawValue, 243
recordCount, 243
recordSize, 243
secActivate, 243
secActivateMask, 243
secDeactivate, 243
secDeactivateMask, 243
secIncrease, 244
secIncreaseMask, 244
secRead, 244
secReadMask, 244
secWrite, 244
secWriteMask, 244
fileID
fileAttributes, 243
fileInfo, 244
uim_fileAttributes, 978
uim_fileInfo, 979
fileIndex
pack_uim_ReadTransparent_t, 676
pack_uim_SLQSUIMGetFileAttributes_t, 680
UIMGetFileAttributesReq, 996
UIMReadTransparentReq, 1000
fileInfo, 244
fileID, 244
path, 244
pathLen, 244
fileSize
fileAttributes, 243
uim_fileAttributes, 978
fileType
fileAttributes, 243
uim_fileAttributes, 978
fill_pack_ctx
common.h, 1403
fill_sdu_hdr
common.h, 1403
filterId
LibPackTFTIDParams, 357
TFTIDParams, 952
Firmware Management Service (FMS), 46
FirmwareID
fwinfo_s, 250
FirmwareUpdatStat, 244
pImgType, 246
pLogString, 246
pLogStringLen, 246
pRefData, 246
pRefString, 246
pRefStringLen, 246
ResCode, 246
fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 670
SwiLocGetAutoStartResp, 927
SwiLocSetAutoStartReq, 928
unpack_swiloc_SwiLocGetAutoStart_t, 1186
fix_rate_reported
SwiLocGetAutoStartResp, 927
unpack_swiloc_SwiLocGetAutoStart_t, 1187

fix_type
pack_swiloc_SwiLocSetAutoStart_t, 670
SwiLocGetAutoStartResp, 927
SwiLocSetAutoStartReq, 928
unpack_swiloc_SwiLocGetAutoStart_t, 1187
fix_type_reported
SwiLocGetAutoStartResp, 927
unpack_swiloc_SwiLocGetAutoStart_t, 1187
fixAccuracy
pack_pds_StartPDSTrackingSessionExt_t, 657
fixCount
pack_pds_StartPDSTrackingSessionExt_t, 657
fixInterval
pack_pds_StartPDSTrackingSessionExt_t, 657
fixTimeout
pack_pds_StartPDSTrackingSessionExt_t, 657
flags
sensorData, 830
sensorData_t, 831
flowLabel
LibPackTFTIDParams, 357
TFTIDParams, 953
fms.h, 1456
GetValidFwPriCombinations, 1457
pack_fms_GetImagesPreference, 1458
pack_fms_GetStoredImages, 1458
pack_fms_SetImagesPreference, 1458
unpack_fms_GetImagesPreference, 1458
unpack_fms_GetStoredImages, 1459
unpack_fms_SetImagesPreference, 1459
Forbidden
nas_QmiNas3GppNetworkInfo, 492
SlqsNas3GppNetworkInfo, 874
forceFlag
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 652
ForceRev0
unpack_nas_GetCDMANetworkParameters_t, 1093
ForceXTRADownload
qaGobiApiPds.h, 1770
format
SMSTransferRouteMTMessage, 914
sMSTransferRouteMTMessage, 915
ForwardMac
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763
fqdnAddr
PCSCFFQDNAddress, 731
wds_PCSCFFQDNAddress, 1375
fqdnLen
PCSCFFQDNAddress, 731
wds_PCSCFFQDNAddress, 1375
freeSlots
sms_maxStorageSizeResp, 892
smsMaxStorageSizeResp, 908
freq
nas_PhyCaAggPcellInfo, 481
nas_PhyCaAggScellIndType, 483
nas_PhyCaAggScellInfo, 486
NASPhyCaAggPcellInfo, 551
NASPhyCaAggScellIndType, 553
NASPhyCaAggScellInfo, 554
PhyCaAggPcellInfo, 738
PhyCaAggScellIndType, 740
PhyCaAggScellInfo, 742
freqsLen
LTEInfoInterfreq, 393
LTEInfoNeighboringGSM, 396
LTEInfoNeighboringWCDMA, 397
nas_LTEInfoInterfreq, 458
nas_LTEInfoNeighboringGSM, 461
nas_LTEInfoNeighboringWCDMA, 462
fromServiceId
BroadcastConfig, 142
sms_BroadcastConfig, 887
fumoResultCode
omaDmFotaTlvExt, 587
function
pack_swiloc_SwiLocSetAutoStart_t, 670
SwiLocGetAutoStartResp, 927
SwiLocSetAutoStartReq, 928
unpack_swiloc_SwiLocGetAutoStart_t, 1187
function_reported
SwiLocGetAutoStartResp, 927
unpack_swiloc_SwiLocGetAutoStart_t, 1187
FwAutoCheck
unpack_swioma_SLQSOMADMGetSettings_t, 1192
FwAvailability
unpack_swioma_SLQSOMADMStartSession_t, 1192
fwdloadsize
omaDmFotaTlv, 584
unpack_omaDmFotaTlv_t, 1142
fwinfo_s, 249
Carrier, 250
FirmwareID, 250
GPSCapability, 250
Region, 250
Technology, 250
fwloadComplete
omaDmFotaTlv, 585
unpack_omaDmFotaTlv_t, 1142
fwvers
CurrentImgList, 198
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1059

g
qmifwinfo_s, 792
gDlBitRate
LibPackQosClassID, 356
QosClassID, 801
GERANInfo, 250
arfcn, 251
bsic, 251
cellID, 252

insNmrCellInfo, 252
lac, 252
nmrInst, 252
plmn, 252
rxLev, 252
timingAdvance, 252
GPRSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1250
GPRSQoS, 284
delayClass, 285
meanThroughputClass, 285
peakThroughputClass, 285
precedenceClass, 285
reliabilityClass, 285
GPRSRequestedQoS, 285
delayClass, 286
meanThroughputClass, 286
peakThroughputClass, 286
precedenceClass, 286
reliabilityClass, 286
GPSCapability
fwinfo_s, 250
GPSLPM
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
GPSSel
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
GPSStateInfo, 286
Altitude, 289
EngineState, 289
glo_almanac_sv_msk, 289
glo_ephemeris_sv_msk, 289
glo_health_sv_msk, 289
glo_visible_sv_msk, 289
gps_almanac_sv_msk, 289
gps_ephemeris_sv_msk, 289
gps_health_sv_msk, 289
gps_visible_sv_msk, 289
HorizontalUncertainty, 289
Iono_valid, 289
Latitude, 289
Longitude, 289
sbas_almanac_sv_msk, 289
sbas_ephemeris_sv_msk, 289
sbas_health_sv_msk, 289
sbas_visible_sv_msk, 290
Time_uncert_ms, 290
TimeStmp_gps_week, 290
TimeStmp_tow_ms, 290
ValidMask, 290
VerticalUncertainty, 290
xtra_start_gps_minutes, 290
xtra_start_gps_week, 290
xtra_valid_duration_hours, 290
GSMRSSIThresh, 292
GSMRSSIThreshListLen, 292
pGSMRSSIThreshList, 292
GSMRSSIThreshListLen
GSMRSSIThresh, 292
nas_GSMRSSIThresh, 442
GSMSSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1127
GSMSrvStatusInfo, 292
isPrefDataPath, 293
srvStatus, 293
trueSrvStatus, 293
GSMSysInfo, 293
cellId, 295
cellIdValid, 295
dtmSupp, 295
dtmSuppValid, 295
egprsSupp, 295
egprsSuppValid, 295
lac, 295
lacValid, 296
MCC, 296
MNC, 296
networkIdValid, 296
regRejectInfoValid, 296
rejCause, 296
rejectSrvDomain, 296
sysInfoGSM, 296
gUlBitRate
LibPackQosClassID, 356
QosClassID, 801
GWAOPTlv
NASQmiCbkNasSystemSelPrefInd, 560
GWAcqOrderPref
NASGWAcqOrderPrefTlv, 539
GWAddressV4
unpack_wds_SLQSGetRuntimeSettings_t, 1250
gcDumpStrLen
CrashInfo, 189
gcdumpString
crashInformation, 191
gcdumpStrlen
crashInformation, 191
Generator
GetAudioProfileReq, 257
GetAudioVolTLBConfigReq, 259
GetM2MAudioProfileResp, 275
GetM2MAudioVolumeReq, 276
SetAudioProfileReq, 842
SetAudioVolTLBConfigReq, 843
SetM2MAudioVolumeReq, 853
geoSysIdx
AddCDMASysInfo, 116
AddSysInfo, 116
nas_AddCDMASysInfo, 416
nas_AddSysInfo, 417
geranArfcn
geranInstInfo, 252
nas_geranInstInfo, 440
geranBsicBcc
geranInstInfo, 252

nas_geranInstInfo, 440
geranBsicNcc
geranInstInfo, 252
nas_geranInstInfo, 441
geranInst
nas_UMTSInfo, 519
UMTSInfo, 1013
GeranInstInfo
nas_UMTSInfo, 519
UMTSInfo, 1013
geranInstInfo, 252
geranArfcn, 252
geranBsicBcc, 252
geranBsicNcc, 252
geranRssi, 252
geranRssi
geranInstInfo, 252
nas_geranInstInfo, 441
get_version
common.h, 1403
GetACCOLC
qaGobiApiNas.h, 1736
GetANAAAAuthenticationStatus
qaGobiApiNas.h, 1736
GetActivationState
qaGobiApiDms.h, 1654
getAllCallInfo
arrCallInfo, 131
voice_arrCallInfo, 1273
getAllCallInformation, 253
ALS, 253
Callinfo, 253
isEmpty, 253
GetAllCallRmtPtyName
arrRemotePartyName, 133
voice_arrRemotePartyName, 1275
getAllCallRmtPtyName, 253
callID, 254
RemotePartyName, 254
getAllCallRmtPtyNum, 254
callID, 254
RemotePartyNum, 254
GetAudioPathConfigReq, 254
Item, 255
Profile, 255
GetAudioPathConfigResp, 255
pCodecSTGain, 256
pDTMFTXGain, 257
pECMode, 257
pMICGainSelect, 257
pNSEnable, 257
pRXAGCList, 257
pRXAVCAGCSwitch, 257
pRXAVCList, 257
pRXPCMIIRFltr, 257
pTXAGCList, 257
pTXAVCSwitch, 257
pTXGain, 257
pTXPCMIIRFltr, 257
GetAudioProfileReq, 257
Generator, 257
GetAudioProfileResp, 257
EarMute, 258
MicMute, 258
Profile, 258
Volume, 258
GetAudioVolTLBConfigReq, 259
Generator, 259
Item, 259
Profile, 259
Volume, 259
GetAudioVolTLBConfigResp, 259
ResCode, 260
GetAutoconnect
qaGobiApiWds.h, 1927
GetByteTotals
qaGobiApiWds.h, 1927
GetCDMANetworkParameters
qaGobiApiNas.h, 1737
getCallFWExtInfo, 260
CallFWExtInfo, 260
numInstances, 260
getCallFWInfo, 260
CallFWInfo, 261
numInstances, 261
GetConnectionRate
qaGobiApiWds.h, 1928
GetCustomFeatureV2
unpack_dms_GetCustFeaturesV2_t, 1029
getCustomFeatureV2, 261
pCustSettingInfo, 261
pCustSettingList, 261
pGetCustomInput, 261
getCustomInput, 262
cust_id, 262
list_type, 262
getDUNCallInfoReq, 262
Mask, 263
pReportChannelRate, 263
pReportConnStatus, 263
pReportDataBearerTech, 263
pReportDormStatus, 263
pTransferStatInd, 263
getDUNCallInfoResp, 264
pCallEndReason, 266
pChannelRate, 266
pConnectionStatus, 266
pDataBearerTech, 266
pDormancyStatus, 266
pLastCallDataBearerTech, 266
pLastCallRXOKBytesCnt, 266
pLastCallTXOKBytesCnt, 266
pMdmCallDurationActive, 266
pRXOKBytesCount, 266
pTXOKBytesCount, 266
GetDataBearerTechnology

qaGobiApiWds.h, 1929
GetDefaultProfile
qaGobiApiWds.h, 1929
GetDefaultProfileLTE
qaGobiApiWds.h, 1931
GetDefaultProfileNum
qaGobiApiWds.h, 1933
GetDeviceCapabilities
qaGobiApiDms.h, 1654
GetDormancyState
qaGobiApiWds.h, 1933
getDyingGaspCfg, 266
pDestSMSContent, 267
pDestSMSNum, 267
getDyingGaspStatistics, 267
pSMSAttemptedFlag, 267
pTimeStamp, 268
GetErrRateResp, 268
pCDMAFrameErrRate, 268
pGSMBER, 268
pHDRPackErrRate, 268
pWCDMABER, 268
GetFirmwareRevision
qaGobiApiDms.h, 1655
GetFirmwareRevisions
qaGobiApiDms.h, 1656
GetHRPDStatsResp, 269
pDRCParams, 269
pPilotSetData, 269
pUATI, 269
GetHardwareRevision
qaGobiApiDms.h, 1656
GetHomeNetwork
qaGobiApiNas.h, 1738
GetHomeNetwork3GPP2
qaGobiApiNas.h, 1739
GetIMSI
qaGobiApiDms.h, 1657
GetIMSSMSConfigParams, 269
pPhoneCtxtURI, 270
pPhoneCtxtURILen, 270
pSMSFormat, 270
pSMSOverIPNwInd, 270
pSettingResp, 270
GetIMSUserConfigParams, 270
pIMSDomain, 271
pIMSDomainLen, 271
pSettingResp, 271
GetIMSVoIPConfigResp, 271
pAmrMode, 273
pAmrOctetAligned, 273
pAmrWBMode, 273
pAmrWBOctetAligned, 273
pAmrWbEnable, 273
pMinSessionExpiryTimer, 273
pRTPRTCPInactTimer, 273
pRingBackTimer, 273
pRingingTimer, 273
pScrAmrEnable, 273
pScrAmrWbEnable, 273
pSessionExpiryTimer, 273
pSettingResp, 273
GetIPAddressLTE
qaGobiApiWds.h, 1934
GetImageStore
qaGobiApiFms.h, 1691
GetImagesPreference
qaGobiApiFms.h, 1690
getIndicationRegResp
qaGobiApiSms.h, 1795
GetInstIDResp, 273
pIPFamily, 273
pInstanceID, 273
GetLastMobileIPError
qaGobiApiWds.h, 1934
GetM2MAVMuteReq, 276
Profile, 276
GetM2MAVMuteResp, 277
CwtMute, 277
EarMute, 277
MicMute, 277
GetM2MAudioProfileReq, 274
pGenerator, 274
GetM2MAudioProfileResp, 274
CwtMute, 275
EarMute, 275
Generator, 275
MicMute, 275
Profile, 275
Volume, 275
GetM2MAudioVolumeReq, 275
Generator, 276
Profile, 276
GetM2MAudioVolumeResp, 276
Level, 276
GetM2MSpkrGainReq, 277
Profile, 278
GetM2MSpkrGainResp, 278
Value, 278
GetManufacturer
qaGobiApiDms.h, 1657
GetMobileIP
qaGobiApiWds.h, 1935
GetMobileIPProfile
qaGobiApiWds.h, 1935
GetModelID
qaGobiApiDms.h, 1658
getMsgWaitingInfo, 278
msgWaitInfo, 279
numInstances, 279
GetNetworkPreference
qaGobiApiNas.h, 1741
GetNetworkTime
qaGobiApiDms.h, 1658
GetNetworkTimeResp, 279
p3GPP2TimeInfo, 279

p3GPPTimeInfo, 279
GetOfflineReason
qaGobiApiDms.h, 1659
GetPDSDefaults
qaGobiApiPds.h, 1770
GetPDSState
qaGobiApiPds.h, 1771
GetPRLVersion
qaGobiApiDms.h, 1660
GetPacketStatistics
qaGobiApiWds.h, 1937
GetPacketStatus
qaGobiApiWds.h, 1937
GetPortAutomaticTracking
qaGobiApiPds.h, 1771
GetPower
qaGobiApiDms.h, 1660
GetProfileSettingIn
qaGobiApiWds.h, 1923
GetProfileSettingOut
qaGobiApiWds.h, 1924
GetRFInfo
qaGobiApiNas.h, 1742
GetRegMgrConfigParams, 279
pIMSTestMode, 280
pPCSCFPort, 280
pPriCSCFPortName, 280
pPriCSCFPortNameLen, 280
pSettingResp, 280
GetSIPConfigResp, 280
pSIPLocalPort, 281
pSettingResp, 281
pSigCompEnabled, 281
pSubscribeTimer, 281
pTimerSIPReg, 281
pTimerT1, 281
pTimerT2, 281
pTimerTf, 281
GetSMSCAddress
qaGobiApiSms.h, 1799
GetSMSWake
qaGobiApiRms.h, 1790
GetSerialNumbers
qaGobiApiDms.h, 1660
GetServiceAutomaticTracking
qaGobiApiPds.h, 1772
GetServingNetwork
qaGobiApiNas.h, 1742
GetServingNetworkCapabilities
qaGobiApiNas.h, 1743
GetSessionDuration
qaGobiApiWds.h, 1938
GetSessionIDResp, 280
pSessionIDv4, 280
pSessionIDv6, 280
GetSessionState
qaGobiApiWds.h, 1938
GetSignalStrengths
qaGobiApiNas.h, 1744
GetStoredImages
qaGobiApiFms.h, 1691
getTransLayerInfoResp
qaGobiApiSms.h, 1796
getTransNWRegInfoResp
qaGobiApiSms.h, 1796
GetValidFwPriCombinations
fms.h, 1457
GetVoiceNumber
qaGobiApiDms.h, 1661
GetXTRAAutomaticDownload
qaGobiApiPds.h, 1772
GetXTRANetwork
qaGobiApiPds.h, 1773
GetXTRAValidity
qaGobiApiPds.h, 1773
glo_almanac_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
glo_ephemeris_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
glo_health_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
glo_visible_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
globalCellId
LTEInfoIntrafreq, 395
nas_LTEInfoIntrafreq, 460
glog
common.h, 1404
gloglvl
common.h, 1404
GnssData, 282
mask, 283
gnssSvId
loc_satelliteInfo, 371
satelliteInfo, 827
gnssSvInfoNotification, 283
bAltitudeAssumed, 284
pSatelliteInfo, 284
gnssSvUsedList
loc_svUsedforFix, 374
svUsedforFix_s, 924
gnssSvUsedList_len
loc_svUsedforFix, 374
svUsedforFix_s, 924
Gpp2TimeZone
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1113
GppNetworkDSTAdjustment
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1113
GppTimeZone
qaQmiServingSystemParam, 769

unpack_nas_SLQSGetServingSystem_t, 1113
gps_almanac_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
gps_ephemeris_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1153
gps_health_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
gps_visible_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
GpsEnable
custFeaturesInfo, 205
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
gpsTime
qaGobiApiCbk.h, 1560
gpsTime_s, 290
gpsTimeOfWeekMs, 290
gpsWeek, 290
gpsTimeOfWeekMs
gpsTime_s, 290
loc_gpsTime, 366
t_gpsTime, 944
gpsWeek
gpsTime_s, 290
loc_gpsTime, 366
t_gpsTime, 944
grntDownlinkBitrate
LibPackUMTSQoS, 360
UMTSMinQoS, 1017
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
grntUplinkBitrate
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
gsmAmrStat
curAMRConfig, 196
voice_curAMRConfig, 1291
GsmCellInfo
lteGsmCellInfo, 391
nas_lteGsmCellInfo, 455
gsmCellInfo, 290
arfcn, 291
band1900, 291
bsicId, 291
cellIdValid, 291
rssi, 291
srxlev, 291
gsmUmtsDl
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
gsmUmtsUl
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
guaranteedRate
dataRate, 214
unpack_qos_dataRate_t, 1155
gwAddressV6
IPV6GWAddressInfo, 331
wds_IPV6GWAddressInfo, 1375
gwV6PrefixLen
IPV6GWAddressInfo, 331
wds_IPV6GWAddressInfo, 1375
gyroAcceptReady
qaGobiApiCbk.h, 1560
gyroAcceptReady_s, 296
batchPerSec, 296
injectEnable, 296
samplesPerBatch, 297
gyroData
pack_loc_SLQSLOCInjectSensorData_t, 617
gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 617
gyroTempAcceptReady
qaGobiApiCbk.h, 1561
gyroTempAcceptReady_s, 297
batchPerSec, 297
injectEnable, 297
samplesPerBatch, 297
gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 618

HASPI
unpack_wds_GetMobileIPProfile_t, 1235
HAState
unpack_wds_GetMobileIPProfile_t, 1236
HDOP
loc_precisionDilution, 369
precisionDilution_s, 748
HDRECIOThresh, 297
HDRECIOThreshListLen, 298
pHDRECIOThreshList, 298
HDRECIOThreshListLen
HDRECIOThresh, 298
nas_HDRECIOThresh, 446
HDRIOThresh, 298
HDRIOThreshListLen, 298
pHDRIOThreshList, 298
HDRIOThreshListLen
HDRIOThresh, 298
nas_HDRIOThresh, 447
HDRPersonalityInd, 298
pCurrentPersonality, 299
pPersonalityListLength, 299
pProtocolSubtypeElement, 299
HDRPersonalityResp, 299
pCurrentPersonality, 299
pPersonalityListLength, 299
pProtocolSubtypeElement, 299
HDRProtSubtypResp, 299
pAppSubType, 300
pCurrentPrsnlty, 300

pPersonalityListLength, 300
pProtoSubTypElmnt, 300
HDRRSSIThresh, 300
HDRRSSIThreshListLen, 301
pHDRRSSIThreshList, 301
HDRRSSIThreshListLen
HDRRSSIThresh, 301
nas_HDRRSSIThresh, 447
HDRSINRThresListLen
HDRSINRThresh, 301
nas_HDRSINRThresh, 448
HDRSINRThresh, 301
HDRSINRThresListLen, 301
pHDRSINRThresList, 301
HDRSINRThreshListLen
HDRSINRThreshold, 302
nas_HDRSINRThreshold, 449
HDRSINRThreshold, 301
HDRSINRThreshListLen, 302
pHDRSINRThreshList, 302
HDRSSInfo, 302
ecio, 303
io, 303
rssi, 303
sinr, 303
unpack_nas_SLQSNasGetSigInfo_t, 1127
HDRSysInfo, 304
hdrActiveProt, 306
hdrActiveProtValid, 306
hdrPersonality, 306
hdrPersonalityValid, 306
is856SysId, 306
is856SysIdValid, 306
isSysPrlMatch, 306
isSysPrlMatchValid, 306
sysInfoHDR, 306
HTTPStatus
_SwiAvmsEventInd, 100
HWVersion
DeviceConfigDetail, 221
nas_DeviceConfigDetail, 435
HardwareControlledMode
unpack_dms_GetPower_t, 1042
has_PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 1149
has_SessionStatus
unpack_pds_SetEventReport_Ind_t, 1149
has_accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 618
has_acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 618
has_accleroData
pack_loc_SLQSLOCInjectSensorData_t, 618
has_altitudeSrcInfo
pack_loc_SLQSLOCInjectPosition_t, 615
has_altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 615
has_altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 615
has_confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 619
has_dLatitude
unpack_pds_SetEventReport_Ind_t, 1148
has_dLongitude
unpack_pds_SetEventReport_Ind_t, 1148
has_gyroData
pack_loc_SLQSLOCInjectSensorData_t, 618
has_gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 618
has_gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 618
has_horConfidence
pack_loc_SLQSLOCInjectPosition_t, 615
has_horReliability
pack_loc_SLQSLOCInjectPosition_t, 615
has_horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 615
has_latitude
pack_loc_SLQSLOCInjectPosition_t, 615
has_longitude
pack_loc_SLQSLOCInjectPosition_t, 615
has_opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 618
has_posSrc
unpack_pds_SetEventReport_Ind_t, 1149
has_positionSrc
pack_loc_SLQSLOCInjectPosition_t, 615
has_rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 615
has_rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 615
has_serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 611
has_timestampAge
pack_loc_SLQSLOCInjectPosition_t, 615
has_timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 615
has_vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 615
has_vertRelicability
pack_loc_SLQSLOCInjectPosition_t, 615
has_vertUnc
pack_loc_SLQSLOCInjectPosition_t, 616
hdrActiveProt
HDRSysInfo, 306
nas_HDRSysInfo, 451
hdrActiveProtValid
HDRSysInfo, 306
nas_HDRSysInfo, 451
hdrHybrid
detailSvcInfo, 220
nas_detailSvcInfo, 434
HdrPersonality
unpack_nas_SLQSGetServingSystem_t, 1113
hdrPersonality
HDRSysInfo, 306

nas_HDRSysInfo, 451
NASServingSystemInfo, 562
qaQmiServingSystemParam, 769
ServingSystemInfo, 835
hdrPersonalityValid
HDRSysInfo, 306
nas_HDRSysInfo, 451
hdrSSInfo, 303
ecio, 304
io, 304
rssi, 304
sinr, 304
hdrSrvStatus
detailSvcInfo, 220
nas_detailSvcInfo, 434
healthStatus
loc_satelliteInfo, 371
satelliteInfo, 827
helper_get_resp_ctx
common.h, 1403
helper_get_xid
common.h, 1403
helper_set_log_func
common.h, 1403
helper_set_log_lvl
common.h, 1403
homeOrigVoiceSO
prefVoiceSO, 750
voice_prefVoiceSO, 1299
homePageVoiceSO
prefVoiceSO, 750
voice_prefVoiceSO, 1299
homeSIDNID, 306
numInstances, 307
SidNid, 307
horConfidence
pack_loc_SLQSLOCInjectPosition_t, 616
horReliability
pack_loc_SLQSLOCInjectPosition_t, 616
horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 616
HorizontalUncertainty
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
hotSwap
hotSwapStatus, 307
uim_hotSwapStatus, 979
hotSwapLength
hotSwapStatus, 307
uim_hotSwapStatus, 979
hotSwapStatus, 307
hotSwap, 307
hotSwapLength, 307
hour
nas_timeInfo, 516
nas_UniversalTime, 522
timeInfo, 954
UniversalTime, 1023
hsCallStatus
nas_WCDMASysInfo, 527
WCDMASysInfo, 1364
hsCallStatusValid
nas_WCDMASysInfo, 527
WCDMASysInfo, 1364
hsInd
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
hsIndValid
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
hwType
wds_DHCPv4HWConfig, 1370
WdsDHCPv4HWConfig, 1386
wdsDhcpv4HwConfig, 1386
hwVer
unpack_dms_GetHardwareRevision_t, 1038

iFaceTab
PCMparams, 730
iFaceTabLen
PCMparams, 730
iGetByteTotals
qaGobiApiWds.h, 1939
iGetConnectionRate
qaGobiApiWds.h, 1939
iGetPacketStatistics
qaGobiApiWds.h, 1939
iLTEbandValue
nas_PhyCaAggPcellInfo, 481
nas_PhyCaAggScellInfo, 486
NASPhyCaAggPcellInfo, 551
NASPhyCaAggScellInfo, 554
PhyCaAggPcellInfo, 738
PhyCaAggScellInfo, 742
IMCNflag
unpack_wds_SLQSGetRuntimeSettings_t, 1250
IMEIString
unpack_dms_GetDeviceSerialNumbers_t, 1034
IMG_ID_LEN
qaGobiApiFms.h, 1686
IMGDETAILS_LEN
qaGobiApiFms.h, 1686
IMS Service (IMS), 55
IMSAIndRegisterInfo, 311
pPdpStatusConfig, 312
pRatHandoverStatusConfig, 312
pRegStatusConfig, 312
pServiceStatusConfig, 312
IMSARegistrationStatus, 314
pImsRegErrCode, 314
pImsRegStatus, 314
pNewImsRegStatus, 314
IMSAServiceStatus, 315
pSmsServiceRat, 317
pSmsServiceStatus, 317
pUtServiceRat, 317
pUtServiceStatus, 317

pVoipServiceRat, 317
pVoipServiceStatus, 317
pVsServiceRat, 317
pVsServiceStatus, 317
pVtServiceRat, 317
pVtServiceStatus, 317
IMSASupportedFieldsResp, 317
pIndFieldsList, 318
pReqFieldsList, 318
pRespFieldsList, 318
IMSASupportedMsgInfo, 318
pSupportedMsgList, 318
IMSI_M_S1_LENGTH
qaGobiApiNas.h, 1723
IMSI_M_S2_LENGTH
qaGobiApiNas.h, 1723
IMSInfo
sMSOnIMSTlv, 912
IMSTlv
unpack_sms_SetNewSMSCallback_ind_t, 1179
INDEX_ZERO
qaNasPerformNetworkScan.h, 1964
INT32
SwiDataTypes.h, 2007
INT8
SwiDataTypes.h, 2007
INVALID_INSTACNE
qaGobiApiCbk.h, 1558
IOThresListLen
IOThresh, 328
nas_IOThresh, 453
IOThresh, 327
IOThresListLen, 328
pIOThresList, 328
IPAddress
DataStatusDetail, 216
nas_DataStatusDetail, 432
IPAddressV6
IPV6AddressInfo, 330
ipv6AddressInfo, 330
wds_IPV6AddressInfo, 1374
IPFamSupport
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
IPFamilyPreference
pack_wds_SLQSSetIPFamilyPreference_t, 722
unpack_wds_SLQSGetRuntimeSettings_t, 1250
IPSECSPI
LibPackTFTIDParams, 357
TFTIDParams, 953
IPV4
qaGobiApiCbk.h, 1558
IPV4V6
qaGobiApiCbk.h, 1558
IPV6
qaGobiApiCbk.h, 1558
IPV6AddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1250
IPV6AddressInfo, 330
IPAddressV6, 330
IPV6PrefixLen, 330
IPV6GWAddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 1250
IPV6GWAddressInfo, 330
gwAddressV6, 331
gwV6PrefixLen, 331
IPV6PrefixLen
IPV6AddressInfo, 330
ipv6AddressInfo, 330
wds_IPV6AddressInfo, 1374
IPv4
unpack_wds_SLQSGetRuntimeSettings_t, 1250
IPv4Addr, 328
addr, 328
subnetMask, 328
wds_IPv4AdTlv, 1374
IPv4AddrTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1228
IPv4DstAddr
unpack_qos_swiQosFilter_t, 1168
IPv4Info, 328
address, 329
port, 329
TlvPresent, 329
IPv4SrcAddr
unpack_qos_swiQosFilter_t, 1168
IPv4Tos
unpack_qos_swiQosFilter_t, 1168
IPv6Addr, 329
addr, 329
prefixLen, 329
IPv6DstAddr
unpack_qos_swiQosFilter_t, 1168
IPv6Info, 331
address, 331
port, 332
TlvPresent, 332
IPv6Label
unpack_qos_swiQosFilter_t, 1168
IPv6SrcAddr
unpack_qos_swiQosFilter_t, 1168
IPv6TrafCls, 332
mask, 332
unpack_qos_swiQosFilter_t, 1168
val, 332
iSLQSMISetIPFamilyPreference
qaGobiApiWds.h, 1939
iSLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1599
iSLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1599
iSLQSSetWdsFirstInstEventCallback
qaGobiApiCbk.h, 1599
iSLQSSetWdsSecondInstEventCallback
qaGobiApiCbk.h, 1599
iSLQSSetWdsThirdInstEventCallback

qaGobiApiCbk.h, 1599
iSLQSSetWdsXferStatsFirstInstCallback
qaGobiApiCbk.h, 1599
iSLQSSetWdsXferStatsSecondInstCallback
qaGobiApiCbk.h, 1599
iSetCATEventCallback
qaGobiApiCbk.h, 1599
iSetSignalStrengthCallback
qaGobiApiCbk.h, 1599
Id
unpack_qos_swiQosFilter_t, 1167
id
BdsSV, 140
CSGID, 194
loc_BdsSV, 362
loc_SV, 373
nas_CSGID, 429
pack_dms_UIMChangePIN_t, 600
pack_dms_UIMSetPINProtection_t, 603
pack_dms_UIMUnblockPIN_t, 604
pack_dms_UIMVerifyPIN_t, 604
QosFlowInfoState, 804
SV, 923
swiQosModifyReq, 940
unpack_qos_QosFlowInfoState_t, 1159
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1165
id_length
custSettingInfo, 208
DMScustSettingInfo, 227
IdleState
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763
image_info_t, 307
buildID, 308
buildIDLen, 308
imageType, 308
uniqueID, 308
ImageElement, 308
buildId, 309
buildIdLength, 309
imageId, 309
imageType, 309
imageID
FMSImageIdElement, 247
ImageIdElement, 310
imageIDElement
FMSImageIDEntries, 248
ImageIDEntries, 310
ImageIDEntries, 310
executingImage, 310
imageIDElement, 310
imageIDSize, 310
imageType, 310
maxImages, 310
imageIDEntries
FMSImageList, 249
ImageList, 311
imageIDSize
FMSImageIDEntries, 248
ImageIDEntries, 310
imageId
FMSImageElement, 247
ImageElement, 309
ImageIdElement, 309
buildID, 309
buildIDLength, 310
failureCount, 310
imageID, 310
storageIndex, 310
ImageList, 311
imageIDEntries, 311
listSize, 311
imageList
unpack_fms_GetStoredImages_t, 1070
ImageListSize
unpack_fms_GetImagesPreference_t, 1070
imageListSize
pack_fms_SetImagesPreference_t, 606
imageType
CurrImageInfo, 200
FMSImageElement, 247
FMSImageIDEntries, 248
image_info_t, 308
ImageElement, 309
ImageIDEntries, 310
ImageTypes
unpack_fms_SetImagesPreference_t, 1071
ImageTypesSize
unpack_fms_SetImagesPreference_t, 1071
imagelistSize
unpack_fms_GetStoredImages_t, 1070
imei_no
unpack_dms_GetSerialNumbers_t, 1044
imeiSize
serialNumbersInfo, 833
unpack_dms_GetDeviceSerialNumbers_t, 1034
imeiSvnSize
serialNumbersInfo, 833
unpack_dms_GetDeviceSerialNumbers_t, 1034
ImeiSvnString
unpack_dms_GetDeviceSerialNumbers_t, 1034
imeisv_svn
unpack_dms_GetSerialNumbers_t, 1044
imgType
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1060
imsCfgIndRegisterInfo, 319
pRegMgrConfigEvents, 320
pSIPConfigEvents, 320
pSMSConfigEvents, 320
pUserConfigEvents, 320
pVoIPConfigEvents, 320
imsRegMgrConfigInfo, 320
pCSCFPortName, 321
pIMSTestMode, 321

pPriCSCFPort, 321
imsRegState
CommInfo, 185
nas_CommInfo, 428
imsSIPConfigInfo, 321
pSIPLocalPort, 322
pSigCompEnabled, 322
pSubscribeTimer, 322
pTimerSIPReg, 322
pTimerT1, 322
pTimerT2, 322
pTimerTf, 322
imsSMSConfigInfo, 322
pPhoneCtxtURI, 323
pSMSFormat, 323
pSMSOverIPNwInd, 323
imsUserConfigInfo, 323
pIMSDomain, 323
imsVoIPConfigInfo, 323
pAmrMode, 325
pAmrOctetAligned, 325
pAmrWBMode, 325
pAmrWBOctetAligned, 325
pAmrWbEnable, 325
pMinSessionExpiryTimer, 325
pRTPRTCPInactTimer, 326
pRingBackTimer, 325
pRingingTimer, 326
pScrAmrEnable, 326
pScrAmrWbEnable, 326
pSessionExpiryTimer, 326
imsaPdpStatusInfo, 312
connetionState, 313
pFailErrorCode, 313
imsaRatStatusInfo, 313
pErrorCodeStr, 313
pRATStatus, 313
pSrcRAT, 313
pTgtRAT, 313
imsaRegStatusInfo, 314
pImsRegStatus, 315
pRegStatusErrorCode, 315
pbIMSRegistered, 315
imsaSvcStatusInfo, 318
pSMSSvcRAT, 319
pSMSSvcStatus, 319
pUTSvcRAT, 319
pUTSvcStatus, 319
pVOIPSvcRAT, 319
pVOIPSvcStatus, 319
pVTSvcRAT, 319
pVTSvcStatus, 319
imsi
unpack_dms_GetIMSI_t, 1039
imsi_11_12
CDMASysInfoExt, 174
nas_CDMASysInfoExt, 425
imsiM1112
minBasedIMSI, 411
nas_minBasedIMSI, 472
imsiMS1
minBasedIMSI, 411
nas_minBasedIMSI, 472
imsiMS2
minBasedIMSI, 411
nas_minBasedIMSI, 472
imsiT1112
nas_trueIMSI, 517
trueIMSI, 962
imsiTS1
nas_trueIMSI, 517
trueIMSI, 962
imsiTS2
nas_trueIMSI, 517
trueIMSI, 962
imsiTaddrNum
nas_trueIMSI, 517
trueIMSI, 962
InUse
nas_QmiNas3GppNetworkInfo, 492
SlqsNas3GppNetworkInfo, 874
includes_pcs_digit
nas_QmisNasPcsDigit, 494
SlqsNasPcsDigit, 875
IndFieldsList, 326
indicationFields, 326
indicationFieldsLen, 326
index
pack_wds_GetMobileIPProfile_t, 706
pack_wds_SetDefaultProfileNum_t, 711
pack_wds_SetMobileIPProfile_t, 714
swiQosFilter, 935
swiQosFlow, 938
swiQosReq, 941
unpack_qos_swiQosFilter_t, 1168
unpack_qos_swiQosFlow_t, 1172
unpack_wds_GetDefaultProfileNum_t, 1233
index1xPri
cardStatus, 156
uim_cardStatus, 972
index1xSec
cardStatus, 156
uim_cardStatus, 972
indexGwPri
cardStatus, 156
uim_cardStatus, 972
indexGwSec
cardStatus, 156
uim_cardStatus, 972
indicationFields
IndFieldsList, 326
indicationFieldsLen
IndFieldsList, 326
Info
unpack_nas_SLQSNasSwiOTAMessageCallback-
_ind_t, 1131

unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind_t, 1132
infoInterFreq, 326
cell_resel_priority, 327
cellInterFreqParams, 327
cells_len, 327
earfcn, 327
threshXHigh, 327
threshXLow, 327
InfoInterfreq
LTEInfoInterfreq, 393
nas_LTEInfoInterfreq, 458
InitiateDomainAttach
qaGobiApiNas.h, 1745
InitiateNetworkRegistration
qaGobiApiNas.h, 1745
injectEnable
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 296
gyroTempAcceptReady_s, 297
injectSensorDataStatus
QmiCbkLocInjectSensorDataInd, 780
injectTimeSyncStatus
QmiCbkLocInjectTimeInd, 781
insNmrCellInfo
GERANInfo, 252
nas_GERANInfo, 440
instanceId
ssdatasession_params, 920
instancesSize
unpack_nas_GetRFInfo_t, 1098
interval
pack_pds_SetPDSDefaults_t, 649
pack_pds_SetXTRAAutomaticDownload_t, 651
pack_wds_SLQSSetWdsEventCallback_t, 723
TransferStatsDataType, 961
Io
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
io
HDRSSInfo, 303
hdrSSInfo, 304
nas_SLQSSignalStrengthsInformation, 508
SLQSSignalStrengthsInformation, 885
ioDelta
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
Iono_valid
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
ip
WdsIpAddressInfoReq, 1391
ipAddress
pack_wds_SetDefaultProfile_t, 710
ipFamily
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1252
ipVersion
LibPackTFTIDParams, 357
TFTIDParams, 953
ipaddr
unpack_wds_GetDefaultProfile_t, 1232
ipaddrv6
unpack_wds_GetDefaultProfile_t, 1232
ipfamily
ssdatasession_params, 920
ipv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1257
ipv4GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1257
ipv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
ipv6AddressInfo, 330
IPAddressV6, 330
IPV6PrefixLen, 330
ipv6GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
is856SysId
HDRSysInfo, 306
nas_HDRSysInfo, 451
is856SysIdValid
HDRSysInfo, 306
nas_HDRSysInfo, 451
is_DataRate_Available
unpack_qos_swiQosFlow_t, 1172
is_EspSpi_Available
unpack_qos_swiQosFilter_t, 1168
is_IPv4DstAddr_Available
unpack_qos_swiQosFilter_t, 1168
is_IPv4SrcAddr_Available
unpack_qos_swiQosFilter_t, 1168
is_IPv4Tos_Available
unpack_qos_swiQosFilter_t, 1168
is_IPv6DstAddr_Available
unpack_qos_swiQosFilter_t, 1168
is_IPv6Label_Available
unpack_qos_swiQosFilter_t, 1168
is_IPv6SrcAddr_Available
unpack_qos_swiQosFilter_t, 1168
is_IPv6TrafCls_Available
unpack_qos_swiQosFilter_t, 1168
is_Id_Available
unpack_qos_swiQosFilter_t, 1168
is_Jitter_Available
unpack_qos_swiQosFlow_t, 1172
is_Latency_Available
unpack_qos_swiQosFlow_t, 1172
is_LteBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1056

is_LteQci_Available
unpack_qos_swiQosFlow_t, 1172
is_MaxAllowedPktSz_Available
unpack_qos_swiQosFlow_t, 1172
is_MinPolicedPktSz_Available
unpack_qos_swiQosFlow_t, 1172
is_NxtHdrProto_Available
unpack_qos_swiQosFilter_t, 1168
is_PktErrRate_Available
unpack_qos_swiQosFlow_t, 1172
is_Precedence_Available
unpack_qos_swiQosFilter_t, 1168
is_ProfileId3GPP2_Available
unpack_qos_swiQosFlow_t, 1172
is_RxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1159
is_TCPDstPort_Available
unpack_qos_swiQosFilter_t, 1168
is_TCPSrcPort_Available
unpack_qos_swiQosFilter_t, 1168
is_TdsBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 1056
is_TokenBucket_Available
unpack_qos_swiQosFlow_t, 1172
is_TrafficClass_Available
unpack_qos_swiQosFlow_t, 1172
is_TranDstPort_Available
unpack_qos_swiQosFilter_t, 1168
is_TranSrcPort_Available
unpack_qos_swiQosFilter_t, 1168
is_TxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 1159
is_UDPDstPort_Available
unpack_qos_swiQosFilter_t, 1168
is_UDPSrcPort_Available
unpack_qos_swiQosFilter_t, 1168
is_val_3GPP2Pri_Available
unpack_qos_swiQosFlow_t, 1172
is_val_3GPPImCn_Available
unpack_qos_swiQosFlow_t, 1172
is_val_3GPPResResidualBER_Available
unpack_qos_swiQosFlow_t, 1172
is_val_3GPPSigInd_Available
unpack_qos_swiQosFlow_t, 1172
is_val_3GPPTraHdlPri_Available
unpack_qos_swiQosFlow_t, 1172
isEmpty
getAllCallInformation, 253
voice_getAllCallInformation, 1293
isInTraffic
nas_txInfo, 517
txInfo, 964
isModByCC
SUPSInfo, 922
isNewFlow
QosFlowInfoState, 804
unpack_qos_QosFlowInfoState_t, 1159
isPrefDataPath
GSMSrvStatusInfo, 293
nas_GSMSrvStatusInfo, 443
nas_SrvStatusInfo, 509
SrvStatusInfo, 918
isRadioTuned
nas_rxInfo, 499
nas_RxSigInfo, 500
rxInfo, 820
RxSigInfo, 823
isSysForbidden
detailSvcInfo, 220
nas_detailSvcInfo, 434
nas_sysInfoCommon, 511
sysInfoCommon, 944
isSysForbiddenValid
nas_sysInfoCommon, 511
sysInfoCommon, 944
isSysPrlMatch
CDMASysInfo, 174
HDRSysInfo, 306
nas_CDMASysInfo, 424
nas_HDRSysInfo, 451
isSysPrlMatchValid
CDMASysInfo, 174
HDRSysInfo, 306
nas_CDMASysInfo, 424
nas_HDRSysInfo, 451
IsVoiceEnabled
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
Item
GetAudioPathConfigReq, 255
GetAudioVolTLBConfigReq, 259
SetAudioVolTLBConfigReq, 843

Jitter
unpack_qos_swiQosFlow_t, 1172

KeyExchange
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763

LITE_VOICE_SUPS_SRV_CLASS_DATA
voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITAS-YNC
voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSY-NC
voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_FAX
voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_NONE
voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS
voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_PADACCESS
voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_SMS

voice.h, 2041
LITE_VOICE_SUPS_SRV_CLASS_VOICE
voice.h, 2041
LITEQMI_QMI_CBK_PARAM_NOCHANGE
sms.h, 1989
LITEQMI_QMI_CBK_PARAM_RESET
sms.h, 1989
LITEQMI_QMI_CBK_PARAM_SET
sms.h, 1989
LBPTlv
NASQmiCbkNasSystemSelPrefInd, 560
LEN
qaGobiApiDcs.h, 1637
LOCEVENTMASKGNSSSVINFO
loc.h, 1463
LOCEVENTMASKNMEA
loc.h, 1464
LOCEVENTMASKWIFIREQ
loc.h, 1464
LOCEventRegisterReqResp, 377
eventRegister, 379
LOCExtPowerStateReqResp, 379
extPowerState, 379
LOCStartReq, 386
pApplicationInfo, 387
pConfigAltitudeAssumed, 387
pHorizontalAccuracyLvl, 387
pIntermediateReportState, 388
pMinIntervalTime, 388
pRecurrenceType, 388
SessionId, 388
LOCStopReq, 388
sessionId, 388
LPCSTR
SwiDataTypes.h, 2007
LTEAttachProfile
unpack_wds_SLQSGet3GPPConfigItem_t, 1242
LTEAttachProfileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1243
LTEAttachProfileListLen
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 721
unpack_wds_SLQSGet3GPPConfigItem_t, 1243
LTEBandPref
NASLTEBandPreferenceTlv, 544
LTECphyCaInfo
unpack_nas_SlqsGetLTECphyCAInfo_t, 1108
LTEInfo, 391
band, 393
bandwidth, 393
emmConnState, 393
emmState, 393
emmSubState, 393
RXChan, 393
TXChan, 393
LTEInfoInterfreq, 393
freqsLen, 393
InfoInterfreq, 393
ueInIdle, 393
LTEInfoIntrafreq, 393
CellParams, 395
cellReselPriority, 395
cellsLen, 395
earfcn, 395
globalCellId, 395
plmn, 395
sIntraSearch, 395
sNonIntraSearch, 395
servingCellId, 395
tac, 395
threshServingLow, 395
ueInIdle, 395
LTEInfoNeighboringGSM, 396
freqsLen, 396
LteGsmCellInfo, 396
ueInIdle, 396
LTEInfoNeighboringWCDMA, 396
freqsLen, 397
LTEWCDMACellInfo, 397
ueInIdle, 397
LTERSRPThresh, 399
LTERSRPThreshListLen, 399
pLTERSRPThreshList, 399
LTERSRPThreshListLen
LTERSRPThresh, 399
nas_LTERSRPThresh, 463
LTERSRQThresh, 399
LTERSRQThreshListLen, 400
pLTERSRQThreshList, 400
LTERSRQThreshListLen
LTERSRQThresh, 400
nas_LTERSRQThresh, 464
LTERSSIThresh, 400
LTERSSIThreshListLen, 400
pLTERSSIThreshList, 400
LTERSSIThreshListLen
LTERSSIThresh, 400
nas_LTERSSIThresh, 464
LTESNRThresListLen
LTESNRThresh, 403
nas_LTESNRThresh, 467
LTESNRThresh, 403
LTESNRThresListLen, 403
pLTESNRThresList, 403
LTESNRThreshListLen
LTESNRThreshold, 404
nas_LTESNRThreshold, 468
LTESNRThreshold, 403
LTESNRThreshListLen, 404
pLTESNRThreshList, 404
LTESSInfo, 404
rsrp, 405
rsrq, 405
rssi, 405
snr, 405
unpack_nas_SLQSNasGetSigInfo_t, 1127

LTESigRptCfg, 401
avgPeriod, 401
rptRate, 401
LTESigRptConfig, 401
avgPeriod, 402
rptRate, 402
LTESysInfo, 406
cellId, 407
cellIdValid, 407
lac, 407
lacValid, 407
MCC, 407
MNC, 408
networkIdValid, 408
regRejectInfoValid, 408
rejCause, 408
rejectSrvDomain, 408
sysInfoLTE, 408
tac, 408
tacValid, 408
LTEWCDMACellInfo
LTEInfoNeighboringWCDMA, 397
nas_LTEInfoNeighboringWCDMA, 462
Lac
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1113
lac
GERANInfo, 252
GSMSysInfo, 295
LTESysInfo, 407
nas_GERANInfo, 440
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_UMTSInfo, 519
nas_WCDMASysInfo, 528
UMTSInfo, 1013
WCDMASysInfo, 1364
lac1
nas_OperatorPLMNData, 480
OperatorPLMNData, 589
lac2
nas_OperatorPLMNData, 480
OperatorPLMNData, 589
lacValid
GSMSysInfo, 296
LTESysInfo, 407
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
language
CDMABroadcastConfig, 162
sms_CDMABroadcastConfig, 888
lastCallDataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 1247
lastCallDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 1245
lastCallRXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1247
lastCallTXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 1247
LastErrCode
DataStatusDetail, 216
nas_DataStatusDetail, 432
Latency
unpack_qos_swiQosFlow_t, 1172
Latitude
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
latitude
pack_loc_SLQSLOCInjectPosition_t, 616
leapSeconds
nas_qaQmi3Gpp2TimeZone, 491
qaQmi3Gpp2TimeZone, 764
leaseState
wds_DHCPLeaseStateTlv, 1368
len
BdsSVInfo, 141
loc_BdsSVInfo, 362
loc_SVInfo, 373
SVInfo, 923
t_Sv, 945
unpack_nas_GetSignalStrengths_t, 1101
length
readTransparentInfo, 806
SMSCAddress, 902
sMSCAddress, 902
SMSEtwsMessage, 904
sMSEtwsMessage, 903
SMSTransferRouteMTMessage, 914
sMSTransferRouteMTMessage, 915
uim_readTransparentInfo, 981
Level
GetM2MAudioVolumeResp, 276
SetM2MAudioVolumeReq, 853
LibPackGPRSRequestedQoS, 332
delayClass, 333
meanThroughputClass, 333
peakThroughputClass, 333
precedenceClass, 333
reliabilityClass, 333
LibPackQosClassID, 355
gDlBitRate, 356
gUlBitRate, 356
maxDlBitRate, 356
maxUlBitRate, 356
QCI, 356
LibPackTFTIDParams, 356
destPortRangeEnd, 357
destPortRangeStart, 357
eValid, 357
filterId, 357
flowLabel, 357
IPSECSPI, 357
ipVersion, 357

nextHeader, 358
pSourceIP, 358
sourceIPMask, 358
srcPortRangeEnd, 358
srcPortRangeStart, 358
tosMask, 358
LibPackUMTSQoS, 358
deliveryErrSDU, 360
grntDownlinkBitrate, 360
grntUplinkBitrate, 360
maxDownlinkBitrate, 360
maxSDUSize, 360
maxUplinkBitrate, 360
qosDeliveryOrder, 360
resBerRatio, 360
sduErrorRatio, 360
trafficClass, 360
trafficPriority, 360
transferDelay, 360
LibPackUMTSReqQoSSigInd, 360
SigInd, 361
UMTSReqQoS, 361
LibPackprofile_3GPP, 344
pAPNClass, 348
pAPNDisabledFlag, 348
pAPNName, 348
pAPNnameSize, 348
pAddrAllocPref, 348
pAuthenticationPref, 349
pGPRSMinimumQoS, 349
pGPRSRequestedQos, 349
pIPv4AddrPref, 349
pIPv6AddPref, 349
pImCnFlag, 349
pPDNInactivTimeout, 349
pPDPtype, 349
pPassword, 349
pPasswordSize, 349
pPcscfAddrUsingDhcp, 349
pPcscfAddrUsingPCO, 349
pPdpAccessConFlag, 349
pPdpContext, 349
pPdpDataCompType, 349
pPdpHdrCompType, 349
pPriDNSIPv4AddPref, 349
pPriDNSIPv6addpref, 349
pPrimaryID, 349
pProfilename, 349
pProfilenameSize, 349
pQosClassID, 349
pSecDNSIPv4AddPref, 349
pSecDNSIPv6addpref, 349
pSecondaryFlag, 349
pTFTID1Params, 349
pTFTID2Params, 349
pUMTSMinQoS, 349
pUMTSMinQosSigInd, 350
pUMTSReqQoS, 350
pUMTSReqQoSSigInd, 350
pUsername, 350
pUsernameSize, 350
LibPackprofile_3GPP2, 350
pAPNClass3GPP2, 354
pAPNEnabled3GPP2, 354
pAllowLinger, 354
pApnString, 354
pApnStringSize, 354
pAppPriority, 354
pAppType, 354
pAuthPassword, 354
pAuthPassword_tSize, 354
pAuthProtocol, 354
pAuthRetryCount, 354
pAuthTimeout, 354
pDataMode, 354
pDataRate, 354
pIpcpAckTimeout, 354
pIpcpCreqRetryCount, 354
pIsPcscfAddressNedded, 354
pLcpAckTimeout, 354
pLcpCreqRetryCount, 354
pNegoDnsSrvrPref, 355
pPDNInactivTimeout3GPP2, 355
pPdnType, 355
pPppSessCloseTimer1x, 355
pPppSessCloseTimerDO, 355
pPriV6DnsAddress, 355
pPrimaryV4DnsAddress, 355
pRATType, 355
pSecV6DnsAddress, 355
pSecondaryV4DnsAddress, 355
pUserId, 355
pUserIdSize, 355
LibpackProfile3GPP, 333
pAPNClass, 337
pAPNDisabledFlag, 337
pAPNName, 337
pAPNnameSize, 337
pAddrAllocPref, 337
pAuthenticationPref, 338
pGPRSMinimumQoS, 338
pGPRSRequestedQos, 338
pIPv4AddrPref, 338
pIPv6AddPref, 338
pImCnFlag, 338
pPDNInactivTimeout, 338
pPDPtype, 338
pPassword, 338
pPasswordSize, 338
pPcscfAddrUsingDhcp, 338
pPcscfAddrUsingPCO, 338
pPdpAccessConFlag, 338
pPdpContext, 338
pPdpDataCompType, 338
pPdpHdrCompType, 338
pPriDNSIPv4AddPref, 338

pPriDNSIPv6addpref, 338
pPrimaryID, 338
pProfilename, 338
pProfilenameSize, 338
pQosClassID, 338
pSecDNSIPv4AddPref, 338
pSecDNSIPv6addpref, 338
pSecondaryFlag, 338
pTFTID1Params, 338
pTFTID2Params, 338
pUMTSMinQoS, 338
pUMTSMinQosSigInd, 339
pUMTSReqQoS, 339
pUMTSReqQoSSigInd, 339
pUsername, 339
pUsernameSize, 339
LibpackProfile3GPP2, 339
pAPNClass3GPP2, 343
pAPNEnabled3GPP2, 343
pAllowLinger, 343
pApnString, 343
pApnStringSize, 343
pAppPriority, 343
pAppType, 343
pAuthPassword, 343
pAuthPasswordSize, 343
pAuthProtocol, 343
pAuthRetryCount, 343
pAuthTimeout, 343
pDataMode, 343
pDataRate, 343
pIpcpAckTimeout, 343
pIpcpCreqRetryCount, 343
pIsPcscfAddressNedded, 343
pLcpAckTimeout, 343
pLcpCreqRetryCount, 343
pNegoDnsSrvrPref, 344
pPDNInactivTimeout3GPP2, 344
pPdnType, 344
pPppSessCloseTimer1x, 344
pPppSessCloseTimerDO, 344
pPriV6DnsAddress, 344
pPrimaryV4DnsAddress, 344
pRATType, 344
pSecV6DnsAddress, 344
pSecondaryV4DnsAddress, 344
pUserId, 344
pUserIdSize, 344
lineCtrlInfo, 361
polarityIncluded, 361
pwrDenialTime, 361
revPolarity, 361
toggleMode, 361
lineValue
pack_voice_SLQSVoiceALSSelectLine_t, 687
voiceALSSelectLineInfo, 1303
linkage
altitudeSrcInfo, 120
altSrcInfo_t, 121
list_type
custSettingList, 208
DMScustSettingList, 227
DMSgetCustomInput, 228
getCustomInput, 262
pack_dms_GetCustFeaturesV2_t, 591
listEntries
FMSPrefImageList, 249
PrefImageList, 748
listSize
FMSImageList, 249
FMSPrefImageList, 249
ImageList, 311
PrefImageList, 748
liteQmiDataBearerMasks
wds.h, 2070
liteServiceClassInformation
voice.h, 2041
liteqmi_GetVersion
common.h, 1403
liteqmi_helper_decode7bitAsciiEncString
common.h, 1403
liteqmi_log
common.h, 1404
loc.h
eQMI_LOC_SESS_STATUS_FAILURE, 1465
eQMI_LOC_SESS_STATUS_IN_PROGRESS, 1465
eQMI_LOC_SESS_STATUS_SUCCESS, 1465
eQMI_LOC_SESS_STATUS_TIMEOUT, 1465
loc.h, 1459
LOCEVENTMASKNMEA, 1464
LOCEVENTMASKWIFIREQ, 1464
MAX_TEMP_DATA_LEN, 1464
pack_loc_DeleteAssistData, 1465
pack_loc_EventRegister, 1465
pack_loc_SLQSLOCGetBestAvailPos, 1466
pack_loc_SLQSLOCGetServer, 1466
pack_loc_SLQSLOCInjectPosition, 1467
pack_loc_SLQSLOCInjectSensorData, 1467
pack_loc_SLQSLOCInjectUTCTime, 1468
pack_loc_SLQSLOCSetCradleMountConfig, 1468
pack_loc_SetExtPowerState, 1465
pack_loc_SetOperationMode, 1466
pack_loc_Start, 1468
pack_loc_Stop, 1469
unpack_loc_BestAvailPos_Ind, 1469
unpack_loc_DeleteAssistData, 1469
unpack_loc_DeleteAssistData_Ind, 1470
unpack_loc_EngineState_Ind, 1470
unpack_loc_EventNMEA_Ind, 1470
unpack_loc_EventRegister, 1471
unpack_loc_GetServer_Ind, 1471
unpack_loc_GnssSvInfo_Ind, 1471
unpack_loc_PositionRpt_Ind, 1472
unpack_loc_SLQSLOCGetBestAvailPos, 1473
unpack_loc_SLQSLOCGetServer, 1474

unpack_loc_SLQSLOCInjectPosition, 1474
unpack_loc_SLQSLOCInjectSensorData, 1474
unpack_loc_SLQSLOCInjectUTCTime, 1474
unpack_loc_SLQSLOCSetCradleMountConfig, 1475
unpack_loc_SetExtPowerConfig_Ind, 1472
unpack_loc_SetExtPowerState, 1472
unpack_loc_SetOperationMode, 1473
unpack_loc_SetOperationMode_Ind, 1473
unpack_loc_Start, 1475
unpack_loc_Stop, 1475
loc_BdsSV, 361
id, 362
mask, 362
loc_BdsSVInfo, 362
len, 362
pSV, 363
loc_CellDb, 363
mask, 363
loc_ClkInfo, 363
mask, 364
loc_GnssData, 364
mask, 365
loc_IPv4Info, 366
address, 367
port, 367
TlvPresent, 367
loc_IPv6Info, 367
address, 367
port, 367
TlvPresent, 367
loc_LocApplicationInfo, 367
appNameLength, 368
appProviderLength, 368
appVersionLength, 368
appVersionValid, 368
pAppName, 368
pAppProvider, 368
pAppVersion, 368
loc_SV, 372
id, 373
mask, 373
system, 373
loc_SVInfo, 373
len, 373
pSV, 373
loc_gpsTime, 366
gpsTimeOfWeekMs, 366
gpsWeek, 366
loc_precisionDilution, 368
HDOP, 369
PDOP, 369
VDOP, 369
loc_satelliteInfo, 369
azimuth, 371
elevation, 371
gnssSvId, 371
healthStatus, 371
snr, 371
svInfoMask, 371
svListLen, 371
svStatus, 371
system, 371
validMask, 371
loc_sensorDataUsage, 371
aidingIndicatorMask, 372
usageMask, 372
loc_svUsedforFix, 373
gnssSvUsedList, 374
gnssSvUsedList_len, 374
loc_urlAddr, 374
address, 374
TlvPresent, 374
LocApplicationInfo, 375
appNameLength, 375
appProviderLength, 375
appVersionLength, 375
appVersionValid, 375
pAppName, 376
pAppProvider, 376
pAppVersion, 376
LocDelAssDataReq, 376
pBdsSVInfo, 376
pCellDb, 376
pClkInfo, 376
pGnssData, 376
pSVInfo, 377
LocGetServerReq, 379
pServerAddrTypeMask, 380
serverType, 380
LocInjectPositionReq, 380
pAltitudeSrcInfo, 383
pAltitudeWrtEllipsoid, 383
pAltitudeWrtMeanSeaLevel, 383
pHorConfidence, 383
pHorReliability, 384
pHorUncCircular, 384
pLatitude, 384
pLongitude, 384
pPositionSrc, 384
pRawHorConfidence, 384
pRawHorUncCircular, 384
pTimestampAge, 384
pTimestampUtc, 384
pVertConfidence, 384
pVertReliability, 384
pVertUnc, 384
LocInjectSensorDataReq, 384
pAcceleroData, 385
pAcceleroTempData, 385
pAcceleroTimeSrc, 385
pGyroData, 385
pGyroTempData, 385
pGyroTimeSrc, 385
pOpaqueIdentifier, 385
LocSetCradleMountReq, 385

pConfidence, 386
state, 386
localTimeOffset
nas_qaQmi3Gpp2TimeZone, 491
qaQmi3Gpp2TimeZone, 764
Location Service(LOC), 57
logString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1060
logger
common.h, 1401
longName
nas_PLMNNetworkNameData, 489
nasPLMNNameResp, 557
PLMNNetworkNameData, 746
unpack_nas_SLQSGetPLMNName_t, 1109
longNameCI
nasPLMNNameResp, 557
unpack_nas_SLQSGetPLMNName_t, 1109
longNameEn
nasPLMNNameResp, 557
unpack_nas_SLQSGetPLMNName_t, 1110
longNameLen
nas_PLMNNetworkNameData, 489
nasPLMNNameResp, 557
PLMNNetworkNameData, 746
unpack_nas_SLQSGetPLMNName_t, 1110
longNameSB
nasPLMNNameResp, 557
unpack_nas_SLQSGetPLMNName_t, 1110
longNameSpareBits
nas_PLMNNetworkNameData, 489
PLMNNetworkNameData, 746
Longitude
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
longitude
pack_loc_SLQSLOCInjectPosition_t, 616
loopbackMode
pack_wds_SLQSSSetLoopback_t, 724
loopbackMultiplier
pack_wds_SLQSSSetLoopback_t, 724
LteBandCapability
unpack_dms_SLQSGetBandCapability_t, 1056
LteCQIParm, 388
CQIValueCW0, 389
CQIValueCW1, 389
ValidityCW0, 389
ValidityCW1, 389
lteEARFCN, 389
earfcn0, 389
earfcn1, 390
status, 390
lteEmmDl
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
lteEmmUl
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
lteEsmDl
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
lteEsmUl
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
LteGsmCellInfo
LTEInfoNeighboringGSM, 396
nas_LTEInfoNeighboringGSM, 461
lteGsmCellInfo, 390
cellReselPriority, 391
cells_len, 391
GsmCellInfo, 391
nccPermitted, 391
threshGsmHigh, 391
threshGsmLow, 391
LteNasReleaseInfo
qaGobiApiCbk.h, 1561
LteNasReleaseInfo_s, 397
nas_major, 397
nas_minor, 397
nas_release, 397
ltePCI, 398
earfcn, 398
pci, 398
status, 398
LteQci
unpack_qos_swiQosFlow_t, 1172
lteRsrpDelta
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
lteRsrpinfo
nas_SLQSSignalStrengthsInformation, 508
SLQSSignalStrengthsInformation, 885
lteRsrpinformation, 398
rsrplevel, 399
lteSSInfo, 405
rsrp, 405
rsrq, 405
rssi, 405
snr, 405
LteSccRxInfoResp, 400
pSccRxInfo, 400
lteSnrDelta
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
lteSnrinfo
nas_SLQSSignalStrengthsInformation, 508
SLQSSignalStrengthsInformation, 885
lteSnrinformation, 402
snrlevel, 403
lteWcdmaCellInfo, 408
cellReselPriority, 409
cellsLen, 409
threshXhigh, 409
threshXlow, 409
uarfcn, 409

WCDMACellInfo, 409
ltersrp
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
ltesnr
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116

m_FwBuildId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 925
m_FwImageId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 925
m_PriBuildId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 925
m_PriImageId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 925
m_nCarrierId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 925
m_nFolderId
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 925
m_nStorage
CarrierImage_t, 157
SWI_STRUCT_CarrierImage, 925
MACIndex
nas_NetworkStatEVDO, 477
NetworkStatEVDO, 576
MAX_BUILD_ID_LEN
dms.h, 1411
MAX_CALL_NO_LEN
qaGobiApiVoice.h, 1902
MAX_CONTENT_LENGTH
qaGobiApiUim.h, 1889
MAX_CUST_ID_LEN
qaGobiApiDms.h, 1646
MAX_FSN_LENGTH
qaGobiApiDms.h, 1646
MAX_ICCID_LENGTH
qaGobiApiUim.h, 1889
uim.h, 2021
MAX_MSE_TWS_MSG
sms.h, 1987
MAX_NMEA_STR_LEN
qaGobiApiCbk.h, 1558
MAX_NO_OF_CALLS
qaGobiApiCbk.h, 1558
qaGobiApiVoice.h, 1902
MAX_NO_OF_FILES
qaGobiApiCbk.h, 1558
MAX_NO_OF_SLOTS
qaGobiApiCbk.h, 1558
qaGobiApiUim.h, 1889
uim.h, 2021
MAX_NO_OF_UUSINFO
qaGobiApiCbk.h, 1558
MAX_PATH_LENGTH
qaGobiApiCbk.h, 1558
qaGobiApiUim.h, 1889
MAX_PILOT_SETS
qaGobiApiNas.h, 1723
MAX_PUK_LENGTH
qaGobiApiUim.h, 1889
MAX_SENSOR_DATA_LEN
loc.h, 1464
MAX_SLOTS_STATUS
qaGobiApiUim.h, 1889
uim.h, 2021
MAX_SMS_LIST_SIZE
sms.h, 1987
MAX_SMS_ROUTES
qaGobiApiSms.h, 1795
MAX_TEMP_DATA_LEN
loc.h, 1464
qaGobiApiLoc.h, 1711
MAXUSSDLENGTH
qaGobiApiCbk.h, 1558
qaGobiApiVoice.h, 1902
MAXVOICEUSSDLENGTH
voice.h, 2041
MCC
SlqsNas3GppNetworkRAT, 87
CDMASysInfo, 174
CDMASysInfoExt, 174
currentPLMN, 199
GSMSysInfo, 296
LTESysInfo, 407
nas_CDMASysInfo, 424
nas_CDMASysInfoExt, 425
nas_currentPLMN, 430
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_QmiNas3GppNetworkInfo, 492
nas_QmiNas3GppNetworkRAT, 493
nas_QmisNasPcsDigit, 494
nas_WCDMASysInfo, 528
SlqsNas3GppNetworkInfo, 874
SlqsNasPcsDigit, 875
unpack_nas_GetServingNetwork_t, 1100
WCDMASysInfo, 1364
MDMCallDuration
ConnectionStatus, 186
connectionStatus, 186
MDMConnStatus
ConnectionStatus, 186
connectionStatus, 186
MEID_MAX_SIZE
dms.h, 1411
MEIDString
unpack_dms_GetDeviceSerialNumbers_t, 1034
MIN
unpack_dms_GetVoiceNumber_t, 1046
MINREQBKLEN

common.h, 1401
MMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1179
MNC
SlqsNas3GppNetworkRAT, 87
CDMASysInfo, 174
currentPLMN, 199
GSMSysInfo, 296
LTESysInfo, 408
nas_CDMASysInfo, 425
nas_currentPLMN, 430
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_QmiNas3GppNetworkInfo, 493
nas_QmiNas3GppNetworkRAT, 493
nas_QmisNasPcsDigit, 494
nas_WCDMASysInfo, 528
SlqsNas3GppNetworkInfo, 874
SlqsNasPcsDigit, 875
unpack_nas_GetServingNetwork_t, 1100
WCDMASysInfo, 1364
MNRInfo, 412
mcc, 412
mnc, 412
rat, 412
MPTlv
NASQmiCbkNasSystemSelPrefInd, 560
MSGID_AND_LEN
common.h, 1401
MSGID_DONT_CARE
common.h, 1401
MTMessageInfo
newMTMessageTlv, 577
manString
pack_dms_SLQSSwiSetHostDevInfo_t, 599
unpack_dms_SLQSSwiGetHostDevInfo_t, 1061
manufacturer
unpack_dms_GetManufacturer_t, 1039
Mask
getDUNCallInfoReq, 263
pack_wds_SLQSGetDUNCallInfo_t, 717
mask
BdsSV, 140
CellDb, 175
ClkInfo, 181
GnssData, 283
IPv6TrafCls, 332
loc_BdsSV, 362
loc_CellDb, 363
loc_ClkInfo, 364
loc_GnssData, 365
loc_SV, 373
SV, 923
Tos, 959
unpack_qos_IPv6TrafCls_t, 1156
unpack_qos_Tos_t, 1174
max
_SLQSAVMSSettingsPeriodsInfo, 87
max_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1244
WDSSWICurrentChannelRates, 1398
max_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 1244
WDSSWICurrentChannelRates, 1398
max_dist
pack_swiloc_SwiLocSetAutoStart_t, 671
SwiLocGetAutoStartResp, 927
SwiLocSetAutoStartReq, 928
unpack_swiloc_SwiLocGetAutoStart_t, 1187
max_dist_reported
SwiLocGetAutoStartResp, 927
unpack_swiloc_SwiLocGetAutoStart_t, 1187
max_time
pack_swiloc_SwiLocSetAutoStart_t, 671
SwiLocGetAutoStartResp, 927
SwiLocSetAutoStartReq, 928
unpack_swiloc_SwiLocGetAutoStart_t, 1187
max_time_reported
SwiLocGetAutoStartResp, 927
unpack_swiloc_SwiLocGetAutoStart_t, 1187
MaxAllowedPktSz
unpack_qos_swiQosFlow_t, 1172
MaxChanRxRate
ChannelRate, 178
dunchannelRate, 234
MaxChanTxRate
ChannelRate, 178
dunchannelRate, 234
maxChannelRXRate
unpack_wds_GetConnectionRate_t, 1229
maxChannelTXRate
unpack_wds_GetConnectionRate_t, 1229
maxDlBitRate
LibPackQosClassID, 356
QosClassID, 801
maxDownlinkBitrate
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
maxImages
FMSImageIDEntries, 248
ImageIDEntries, 310
maxMitigationLevel
mitigationDevList, 411
MaxRXChannelRate
unpack_dms_GetDeviceCap_t, 1030
maxRxChannelRate
unpack_dms_GetDeviceCapabilities_t, 1031
maxSDUSize
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380

maxStorageSize
sms_maxStorageSizeResp, 892
smsMaxStorageSizeResp, 908
MaxTXChannelRate
unpack_dms_GetDeviceCap_t, 1030
maxTxChannelRate
unpack_dms_GetDeviceCapabilities_t, 1031
maxUlBitRate
LibPackQosClassID, 356
QosClassID, 801
maxUplinkBitrate
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
mcTimeStamp
cdmaMsgDecodingParams, 166
mcc
CSGID, 194
MNRInfo, 412
nas_CSGID, 429
nas_MNRInfo, 473
nas_netSelectionPref, 474
nas_OperatorPLMNData, 480
nasPLMNNameReq, 555
netSelectionPref, 570
OperatorPLMNData, 589
pack_nas_SLQSGetPLMNName_t, 627
PlmnID, 744
unpack_nas_GetHomeNetwork_t, 1096
mccM
minBasedIMSI, 411
nas_minBasedIMSI, 472
mccT
nas_trueIMSI, 517
trueIMSI, 962
mdmCallDurationActive
unpack_wds_SLQSGetDUNCallInfo_t, 1247
MdmConnStatus
DUNCallInfoInd, 233
meanThroughputClass
GPRSQoS, 285
GPRSRequestedQoS, 286
LibPackGPRSRequestedQoS, 333
wds_GPRSQoS, 1373
meid
unpack_dms_GetSerialNumbers_t, 1044
meidLength
_SLQSSwiGetSerialNoExtParams, 97
meidSize
serialNumbersInfo, 833
unpack_dms_GetDeviceSerialNumbers_t, 1034
meidString
unpack_dms_SLQSSwiGetSerialNoExt_t, 1063
message
unpack_sms_SLQSGetSMS_t, 1181
message_type
NASOTAMessageTlv, 551
messageClass
sms_routeEntry, 895
smsRouteEntry, 913
messageFailureCode
slqssendsmsparams_s, 878
unpack_sms_SendSMS_t, 1178
messageFormat
pack_sms_SaveSMS_t, 661
pack_sms_SendSMS_t, 662
slqssendasyncsmsparams_s, 877
slqssendsmsparams_s, 878
sms_sendAsyncsmsParams, 897
unpack_sms_SLQSGetSMS_t, 1181
messageID
slqssendsmsparams_s, 878
SMSAsyncRawSend_s, 901
unpack_sms_SendSMS_t, 1178
messageId
cdmaMsgEncodingParams, 168
messageIndex
pack_sms_SLQSGetSMS_t, 664
pack_sms_SLQSModifySMSStatus_t, 666
qmiSmsMessageList, 795
SMSMTMessage, 910
sMSMTMessage, 910
messageLength
cdmaMsgDecodingParams, 166
messageList
unpack_sms_SLQSGetSMSList_t, 1182
messageListSize
unpack_sms_SLQSGetSMSList_t, 1182
messageMode
SMSMemoryInfo, 908
SMSMessageMode, 909
sMSMessageMode, 909
unpack_sms_SLQSWmsMemoryFullCallBack_ind-
_t, 1185
MessageModeInfo
messageModeTlv, 409
messageModeTlv, 409
MessageModeInfo, 409
TlvPresent, 409
messageSize
pack_sms_SaveSMS_t, 661
pack_sms_SendSMS_t, 662
slqssendasyncsmsparams_s, 877
slqssendsmsparams_s, 878
sms_sendAsyncsmsParams, 897
unpack_sms_SLQSGetSMS_t, 1181
wcdmaMsgEncodingParams, 1360
messageTag
pack_sms_SLQSModifySMSStatus_t, 666
qmiSmsMessageList, 795
unpack_sms_SLQSGetSMS_t, 1181
messageType
sms_routeEntry, 895
smsRouteEntry, 913
messageWaitingInfoContent, 409

activeInd, 410
msgCount, 410
msgType, 410
MicMute
GetAudioProfileResp, 258
GetM2MAudioProfileResp, 275
GetM2MAVMuteResp, 277
SetAudioProfileReq, 842
SetM2MAVMuteReq, 853
min
_SLQSAVMSSettingsPeriodsInfo, 87
minBasedIMSI, 410
imsiM1112, 411
imsiMS1, 411
imsiMS2, 411
mccM, 411
MinPolicedPktSz
unpack_qos_swiQosFlow_t, 1172
minSize
unpack_dms_GetVoiceNumber_t, 1046
minute
nas_timeInfo, 516
nas_UniversalTime, 522
timeInfo, 954
UniversalTime, 1023
mipMode
unpack_wds_GetMobileIP_t, 1234
mipStatus
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
mipstatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
mitigationDevID
TmdDeRegNotMitigationLvlReq, 955
TmdGetMitigationLvlReq, 956
TmdMitigationLvlIndReq, 957
TmdRegNotMitigationLvlReq, 958
mitigationDevIDLen
TmdDeRegNotMitigationLvlReq, 955
TmdGetMitigationLvlReq, 956
TmdMitigationLvlIndReq, 957
TmdRegNotMitigationLvlReq, 958
mitigationDevId
mitigationDevList, 411
mitigationDevIdLen
mitigationDevList, 412
MitigationDevInfo
QmiCbkTmdMitiLvlRptInd, 791
mitigationDevList, 411
maxMitigationLevel, 411
mitigationDevId, 411
mitigationDevIdLen, 412
mnc
CSGID, 194
MNRInfo, 412
nas_CSGID, 429
nas_MNRInfo, 473
nas_netSelectionPref, 474
nas_OperatorPLMNData, 480
nasPLMNNameReq, 555
netSelectionPref, 571
OperatorPLMNData, 589
pack_nas_SLQSGetPLMNName_t, 627
PlmnID, 744
unpack_nas_GetHomeNetwork_t, 1096
mncPcsDigits
CSGID, 194
nas_CSGID, 429
mobileCountryCode
SMSEtwsPlmn, 905
sMSEtwsPlmn, 905
mobileIP
pack_wds_SLQSSetWdsEventCallback_t, 723
mobileNetworkCode
SMSEtwsPlmn, 905
sMSEtwsPlmn, 905
mode
callInfo, 153
pack_dms_SetEventReport_t, 595
pack_dms_SetPower_t, 596
pack_loc_SetOperationMode_t, 610
pack_sms_SLQSGetSmsBroadcastConfig_t, 664
pack_sms_SLQSSetSmsBroadcastActivation_t, 667
pack_sms_SLQSSetSmsBroadcastConfig_t, 668
pack_wds_SetMobileIP_t, 711
uim_refreshevent, 983
UIMRefreshEvent, 1003
voice_callInfo, 1284
ModePref
NASModePreferenceTlv, 545
modelString
pack_dms_SLQSSwiSetHostDevInfo_t, 599
unpack_dms_SLQSSwiGetHostDevInfo_t, 1061
modelid
unpack_dms_GetModelID_t, 1040
modelid_str
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
modemMode
CommInfo, 185
nas_CommInfo, 428
modemTempNotification
qaGobiApiCbk.h, 1562
ModemTempState
_modemTempNotification, 68
ModemTemperature
_modemTempNotification, 68
modemindex
pack_fms_SetImagesPreference_t, 606
ModifyProfileIn, 412
curProfile, 413
pProfileID, 413
pProfileType, 413
ModifyProfileOut, 413

pExtErrorCode, 413
month
nas_timeInfo, 516
nas_UniversalTime, 522
timeInfo, 954
UniversalTime, 1023
msgCount
messageWaitingInfoContent, 410
sms_messageWaitingInfoContent, 892
msgDelFailureCause
SMSAsyncRawSend_s, 901
msgDelFailureType
SMSAsyncRawSend_s, 901
msgProtocol
sms_msgProtocolResp, 893
smsMsgprotocolResp, 910
msgType
messageWaitingInfoContent, 410
sms_messageWaitingInfoContent, 892
msgWaitInfo
getMsgWaitingInfo, 279
msgWaitingInfo, 414
sms_getMsgWaitingInfo, 889
msgWaitingInfo, 413
msgWaitInfo, 414
numInstances, 414
msgid
pack_qmi_t, 658
unpack_qmi_t, 1155
msgtype
common.h, 1402
Mtu
unpack_wds_SLQSGetRuntimeSettings_t, 1250
MultDisc
nas_protocolSubtypeElement, 490
protocolSubtypeElement, 763
multiplier
pktErrRate, 743
unpack_qos_pktErrRate_t, 1157

NAI
unpack_wds_GetMobileIPProfile_t, 1236
NAM_NAME_LENGTH
qaGobiApiNas.h, 1723
NAS_NAM_NAME_LENGTH
nas.h, 1484
NAS_PLMN_LENGTH
nas.h, 1484
NAS_SRV
qaGobiApiCbk.h, 1558
NASBandPreferenceTlv, 529
band_pref, 529
TlvPresent, 529
NASEmergencyModeTlv, 530
EmerMode, 531
TlvPresent, 531
NASGWAcqOrderPrefTlv, 539
GWAcqOrderPref, 539
TlvPresent, 539
NASLTEBandPreferenceTlv, 543
LTEBandPref, 544
TlvPresent, 544
NASLteNasReleaseInfoTlv, 544
nas_major, 544
nas_minor, 544
nas_release, 544
TlvPresent, 544
NASModePreferenceTlv, 545
ModePref, 545
TlvPresent, 545
NASNetSelPreferenceTlv, 545
NetSelPref, 545
TlvPresent, 545
NASOTAMessageTlv, 550
data_buf, 551
data_len, 551
message_type, 551
TlvPresent, 551
NASPRLPreferenceTlv, 558
PRLPref, 558
TlvPresent, 558
NASPhyCaAggPcellInfo, 551
dl_bw_value, 551
freq, 551
iLTEbandValue, 551
pci, 551
TlvPresent, 552
NASPhyCaAggScellDlBw, 552
dl_bw_value, 552
TlvPresent, 552
NASPhyCaAggScellIndType, 553
freq, 553
pci, 553
scell_state, 553
TlvPresent, 553
NASPhyCaAggScellIndex, 552
scell_idx, 552
TlvPresent, 553
NASPhyCaAggScellInfo, 553
dl_bw_value, 554
freq, 554
iLTEbandValue, 554
pci, 554
scell_state, 554
TlvPresent, 554
NASQmiCbkNasSwiOTAMessageInd, 558
nasRelInfoTlv, 558
otaMsgTlv, 559
timeTlv, 559
NASQmiCbkNasSystemSelPrefInd, 559
BPTlv, 560
EMTlv, 560
GWAOPTlv, 560
LBPTlv, 560
MPTlv, 560
NSPTlv, 560
PRLPTlv, 560

RPTlv, 560
SDPTlv, 560
NASRoamPreferenceTlv, 560
RoamPref, 560
TlvPresent, 560
NASServDomainPrefTlv, 560
SrvDomainPref, 561
TlvPresent, 561
NASServingSystemInfo, 561
csAttachState, 562
hdrPersonality, 562
psAttachState, 562
radioInterfaceList, 562
radioInterfaceNo, 562
registrationState, 562
selectedNetwork, 562
NASTimeInfoTlv, 569
time, 569
TlvPresent, 569
NMEAData
unpack_loc_EventNMEA_Ind_t, 1079
NSPTlv
NASQmiCbkNasSystemSelPrefInd, 560
NSSAudioCtrl, 579
downLink, 579
upLink, 579
NUM_OF_SET
qaGobiApiCbk.h, 1558
qaGobiApiSms.h, 1795
NWProfile, 579
pProfSz, 580
pProfValues, 580
tech, 580
NWQoSStatus
unpack_qos_SLQSQosGetNetworkStatus_t, 1160
NWRegStat
_transNWRegInfoNotification, 112
naiSize
unpack_wds_GetMobileIPProfile_t, 1236
namID
airTimer, 117
nasGet3GPP2SubscriptionInfoReq, 531
pack_nas_SLQSNasGet3GPP2Subscription_t, 633
prefVoiceSO, 750
roamTimer, 816
voice_airTimer, 1267
voice_prefVoiceSO, 1299
voice_roamTimer, 1301
namName, 414
namName, 414
namNameLen, 414
namName, 414
nas_namName, 473
namNameLen
namName, 414
nas_namName, 473
Name
unpack_nas_GetServingNetwork_t, 1100
name
unpack_nas_GetHomeNetwork_t, 1096
unpack_wds_GetDefaultProfile_t, 1232
nameLen
remotePartyName, 810
voice_remotePartyName, 1300
namePI
remotePartyName, 810
voice_remotePartyName, 1300
nameSize
unpack_nas_GetHomeNetwork3GPP2_t, 1095
unpack_nas_GetServingNetwork_t, 1100
nameString
pack_dms_SLQSSwiSetOSInfo_t, 600
unpack_dms_SLQSSwiGetOSInfo_t, 1062
namelength
omaDmFotaTlv, 585
omaDmFotaTlvExt, 587
unpack_omaDmFotaTlv_t, 1142
namesize
unpack_wds_GetDefaultProfile_t, 1232
nas.h
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6, 1484
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75, 1484
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1484
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1484
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6, 1484
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75, 1484
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE, 1485
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE, 1485
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE, 1485
nas.h, 1476
NAS_PLMN_LENGTH, 1484
pack_nas_GetACCOLC, 1485
pack_nas_GetANAAAAuthenticationStatus, 1485

pack_nas_GetCDMANetworkParameters, 1485
pack_nas_GetHomeNetwork, 1486
pack_nas_GetHomeNetwork3GPP2, 1486
pack_nas_GetNetworkPreference, 1486
pack_nas_GetRFInfo, 1486
pack_nas_GetServingNetwork, 1487
pack_nas_GetServingNetworkCapabilities, 1487
pack_nas_GetSignalStrengths, 1487
pack_nas_InitiateDomainAttach, 1488
pack_nas_PerformNetworkScan, 1488
pack_nas_SLQSConfigSigInfo, 1490
pack_nas_SLQSGetErrorRate, 1490
pack_nas_SLQSGetNetworkTime, 1491
pack_nas_SLQSGetOperatorNameData, 1491
pack_nas_SLQSGetPLMNName, 1492
pack_nas_SLQSGetServingSystem, 1492
pack_nas_SLQSGetSignalStrength, 1492
pack_nas_SLQSGetSysInfo, 1493
pack_nas_SLQSGetSysSelectionPref, 1493
pack_nas_SLQSInitiateNetworkRegistration, 1493
pack_nas_SLQSNASSwiGetChannelLock, 1496
pack_nas_SLQSNASSwiSetChannelLock, 1497
pack_nas_SLQSNasConfigSigInfo2, 1494
pack_nas_SLQSNasGet3GPP2Subscription, 1494
pack_nas_SLQSNasGetCellLocationInfo, 1494
pack_nas_SLQSNasGetHDRColorCode, 1495
pack_nas_SLQSNasGetSigInfo, 1495
pack_nas_SLQSNasGetTxRxInfo, 1495
pack_nas_SLQSNasIndicationRegisterExt, 1496
pack_nas_SLQSNasSwiIndicationRegister, 1496
pack_nas_SLQSNasSwiModemStatus, 1497
pack_nas_SLQSSetBandPreference, 1497
pack_nas_SLQSSetSignalStrengthsCallback, 1498
pack_nas_SLQSSetSysSelectionPref, 1498
pack_nas_SLQSSwiGetHDRPersonality, 1498
pack_nas_SLQSSwiGetHDRProtSubtype, 1499
pack_nas_SLQSSwiGetHRPDStats, 1499
pack_nas_SLQSSwiGetLteCQI, 1499
pack_nas_SLQSSwiGetLteSccRxInfo, 1500
pack_nas_SLQSSwiNetworkDebug, 1500
pack_nas_SLQSSwiPSDetach, 1500
pack_nas_SetACCOLC, 1488
pack_nas_SetCDMANetworkParameters, 1489
pack_nas_SetLURejectCallback, 1489
pack_nas_SetNetworkPreference, 1489
pack_nas_SetRFInfoCallback, 1490
pack_nas_SlqsGetLTECphyCAInfo, 1491
unpack_nas_GetACCOLC, 1501
unpack_nas_GetANAAAAuthenticationStatus, 1501
unpack_nas_GetCDMANetworkParameters, 1501
unpack_nas_GetHomeNetwork, 1502
unpack_nas_GetHomeNetwork3GPP2, 1502
unpack_nas_GetNetworkPreference, 1502
unpack_nas_GetRFInfo, 1503
unpack_nas_GetServingNetwork, 1503
unpack_nas_GetServingNetworkCapabilities, 1504
unpack_nas_GetSignalStrengths, 1504
unpack_nas_InitiateDomainAttach, 1504
unpack_nas_PerformNetworkScan, 1505
unpack_nas_SLQSConfigSigInfo, 1508
unpack_nas_SLQSGetErrorRate, 1508
unpack_nas_SLQSGetNetworkTime, 1509
unpack_nas_SLQSGetOperatorNameData, 1509
unpack_nas_SLQSGetPLMNName, 1510
unpack_nas_SLQSGetServingSystem, 1510
unpack_nas_SLQSGetSignalStrength, 1510
unpack_nas_SLQSGetSysInfo, 1511
unpack_nas_SLQSGetSysSelectionPref, 1511
unpack_nas_SLQSInitiateNetworkRegistration, 1511
unpack_nas_SLQSNASSwiGetChannelLock, 1515
unpack_nas_SLQSNASSwiSetChannelLock, 1516
unpack_nas_SLQSNasConfigSigInfo2, 1512
unpack_nas_SLQSNasGet3GPP2Subscription, 1512
unpack_nas_SLQSNasGetCellLocationInfo, 1512
unpack_nas_SLQSNasGetHDRColorCode, 1513
unpack_nas_SLQSNasGetSigInfo, 1513
unpack_nas_SLQSNasGetTxRxInfo, 1513
unpack_nas_SLQSNasIndicationRegisterExt, 1514
unpack_nas_SLQSNasNetworkTimeCallBack_ind, 1514
unpack_nas_SLQSNasSigInfoCallback_ind, 1514
unpack_nas_SLQSNasSwiIndicationRegister, 1515
unpack_nas_SLQSNasSwiModemStatus, 1515
unpack_nas_SLQSNasSwiOTAMessageCallback-_ind, 1516
unpack_nas_SLQSNasSysInfoCallback_ind, 1516
unpack_nas_SLQSNasTimerCallback_ind, 1517
unpack_nas_SLQSSetBandPreference, 1517
unpack_nas_SLQSSetSignalStrengthsCallback, 1517
unpack_nas_SLQSSetSysSelectionPref, 1517
unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind, 1518
unpack_nas_SLQSSwiGetHDRPersonality, 1518
unpack_nas_SLQSSwiGetHDRProtSubtype, 1518
unpack_nas_SLQSSwiGetHRPDStats, 1519
unpack_nas_SLQSSwiGetLteCQI, 1519
unpack_nas_SLQSSwiGetLteSccRxInfo, 1519
unpack_nas_SLQSSwiNetworkDebug, 1520
unpack_nas_SLQSSwiPSDetach, 1520
unpack_nas_SetACCOLC, 1505
unpack_nas_SetCDMANetworkParameters, 1505
unpack_nas_SetDataCapabilitiesCallback_ind, 1506
unpack_nas_SetEventReportInd, 1506
unpack_nas_SetLURejectCallback, 1506
unpack_nas_SetNasLTECphyCaIndCallback_ind, 1506

unpack_nas_SetNetworkPreference, 1507
unpack_nas_SetRFInfoCallback, 1507
unpack_nas_SetRoamingIndicatorCallback_ind, 1507
unpack_nas_SetServingSystemCallback_ind, 1508
unpack_nas_SlqsGetLTECphyCAInfo, 1509
nas_ActPilotPNElement, 415
ActSetPilotPN, 415
ActSetPilotPNStrength, 416
nas_AddCDMASysInfo, 416
geoSysIdx, 416
regPrd, 416
nas_AddSysInfo, 416
cellBroadcastCap, 417
geoSysIdx, 417
nas_CDMAChannel, 418
priChA, 419
priChB, 419
secChA, 419
secChB, 419
nas_CDMAECIOThresh, 419
CDMAECIOThreshListLen, 419
pCDMAECIOThreshList, 420
nas_CDMAInfo, 420
baseId, 420
baseLat, 420
baseLong, 420
nid, 421
refpn, 421
sid, 421
nas_CDMARSSIThresh, 421
CDMARSSIThreshListLen, 421
pCDMARSSIThreshList, 421
nas_CDMASysInfo, 421
baseId, 424
baseLat, 424
baseLong, 424
bsInfoValid, 424
bsPRev, 424
bsPRevValid, 424
ccsSupported, 424
ccsSupportedValid, 424
cdmaSysIdValid, 424
isSysPrlMatch, 424
isSysPrlMatchValid, 424
MCC, 424
MNC, 425
networkID, 425
networkIdValid, 425
pRevInUse, 425
pRevInUseValid, 425
packetZone, 425
packetZoneValid, 425
sysInfoCDMA, 425
systemID, 425
nas_CDMASysInfoExt, 425
imsi_11_12, 425
MCC, 425
nas_CSGID, 428
id, 429
mcc, 429
mnc, 429
mncPcsDigits, 429
rat, 429
nas_CallBarringSysInfo, 417
csBarStatus, 417
psBarStatus, 417
nas_CommInfo, 426
imsRegState, 428
modemMode, 428
psState, 428
systemMode, 428
temperature, 428
nas_DRCParams, 435
DRCCover, 436
DRCValue, 436
nas_DataStatusDetail, 431
IPAddress, 432
LastErrCode, 432
nas_DeviceConfigDetail, 434
Chipset, 435
HWVersion, 435
QLIC, 435
Technology, 435
nas_ECIOThresh, 437
ECIOThresListLen, 437
pECIOThresList, 437
nas_GERANInfo, 438
arfcn, 440
bsic, 440
cellID, 440
insNmrCellInfo, 440
lac, 440
nmrInst, 440
plmn, 440
rxLev, 440
timingAdvance, 440
nas_GSMRSSIThresh, 442
GSMRSSIThreshListLen, 442
pGSMRSSIThreshList, 442
nas_GSMSrvStatusInfo, 442
isPrefDataPath, 443
srvStatus, 443
trueSrvStatus, 443
nas_GSMSysInfo, 443
cellId, 445
cellIdValid, 445
dtmSupp, 445
dtmSuppValid, 445
egprsSupp, 446
egprsSuppValid, 446
lac, 446
lacValid, 446
MCC, 446
MNC, 446

networkIdValid, 446
regRejectInfoValid, 446
rejCause, 446
rejectSrvDomain, 446
sysInfoGSM, 446
nas_HDRECIOThresh, 446
HDRECIOThreshListLen, 446
pHDRECIOThreshList, 446
nas_HDRIOThresh, 446
HDRIOThreshListLen, 447
pHDRIOThreshList, 447
nas_HDRRSSIThresh, 447
HDRRSSIThreshListLen, 447
pHDRRSSIThreshList, 447
nas_HDRSINRThresh, 448
HDRSINRThresListLen, 448
pHDRSINRThresList, 448
nas_HDRSINRThreshold, 448
HDRSINRThreshListLen, 449
pHDRSINRThreshList, 449
nas_HDRSysInfo, 449
hdrActiveProt, 451
hdrActiveProtValid, 451
hdrPersonality, 451
hdrPersonalityValid, 451
is856SysId, 451
is856SysIdValid, 451
isSysPrlMatch, 451
isSysPrlMatchValid, 451
sysInfoHDR, 451
nas_IOThresh, 453
IOThresListLen, 453
pIOThresList, 453
nas_LTEInfo, 455
band, 457
bandwidth, 457
emmConnState, 457
emmState, 457
emmSubState, 457
RXChan, 457
TXChan, 457
nas_LTEInfoInterfreq, 458
freqsLen, 458
InfoInterfreq, 458
ueInIdle, 458
nas_LTEInfoIntrafreq, 458
CellParams, 460
cellReselPriority, 460
cellsLen, 460
earfcn, 460
globalCellId, 460
plmn, 460
sIntraSearch, 460
sNonIntraSearch, 460
servingCellId, 460
tac, 460
threshServingLow, 460
ueInIdle, 460
nas_LTEInfoNeighboringGSM, 460
freqsLen, 461
LteGsmCellInfo, 461
ueInIdle, 461
nas_LTEInfoNeighboringWCDMA, 461
freqsLen, 462
ueInIdle, 462
nas_LTERSRPThresh, 463
LTERSRPThreshListLen, 463
pLTERSRPThreshList, 463
nas_LTERSRQThresh, 463
LTERSRQThreshListLen, 464
pLTERSRQThreshList, 464
nas_LTERSSIThresh, 464
LTERSSIThreshListLen, 464
pLTERSSIThreshList, 464
nas_LTESNRThresh, 467
LTESNRThresListLen, 467
pLTESNRThresList, 467
nas_LTESNRThreshold, 467
LTESNRThreshListLen, 468
pLTESNRThreshList, 468
nas_LTESigRptCfg, 464
avgPeriod, 465
rptRate, 465
nas_LTESigRptConfig, 465
avgPeriod, 466
rptRate, 466
nas_LTESysInfo, 468
cellId, 470
cellIdValid, 470
lac, 470
lacValid, 470
MCC, 470
MNC, 470
networkIdValid, 470
regRejectInfoValid, 470
rejCause, 470
rejectSrvDomain, 470
sysInfoLTE, 470
tac, 470
tacValid, 470
nas_MNRInfo, 472
mcc, 473
mnc, 473
rat, 473
nas_NetworkStat1x, 474
ActSetCnt, 476
NeighborSetCnt, 476
pActPilotPNElements, 476
pNeighborSetPilotPN, 476
RX_EC_IO, 476
RX_PWR, 476
SO, 476
State, 476
TX_PWR, 476
nas_NetworkStatEVDO, 476
MACIndex, 477

PER, 478
pSectorID, 478
PilotEnergy, 478
RX_PWR, 478
SNR, 478
SectorIDLen, 478
State, 478
nas_OperatorPLMNData, 479
lac1, 480
lac2, 480
mcc, 480
mnc, 480
PLMNRecID, 480
nas_PLMNNetworkName, 487
numInstance, 487
PLMNNetName, 487
nas_PLMNNetworkNameData, 488
codingScheme, 489
countryInitials, 489
longName, 489
longNameLen, 489
longNameSpareBits, 489
shortName, 489
shortNameLen, 489
shortNameSpareBits, 489
nas_PhyCaAggPcellInfo, 481
dl_bw_value, 481
freq, 481
iLTEbandValue, 481
pci, 482
TlvPresent, 482
nas_PhyCaAggScellDlBw, 482
dl_bw_value, 482
TlvPresent, 482
nas_PhyCaAggScellIndType, 483
freq, 483
pci, 483
scell_state, 483
TlvPresent, 483
nas_PhyCaAggScellIndex, 482
scell_idx, 483
TlvPresent, 483
nas_PhyCaAggScellInfo, 483
dl_bw_value, 486
freq, 486
iLTEbandValue, 486
pci, 486
scell_state, 486
TlvPresent, 486
nas_PilotSetData, 486
NumPilots, 486
pPilotSetInfo, 486
nas_PilotSetParams, 486
PilotPN, 487
PilotStrength, 487
PilotType, 487
nas_QmiNas3GppNetworkInfo, 491
Desription, 492
Forbidden, 492
InUse, 492
MCC, 492
MNC, 493
Preferred, 493
Roaming, 493
nas_QmiNas3GppNetworkRAT, 493
MCC, 493
MNC, 493
RAT, 493
nas_QmisNasPcsDigit, 493
includes_pcs_digit, 494
MCC, 494
MNC, 494
nas_RFInfoTlv, 494
activeBandClass, 495
activeChannel, 495
radioInterface, 495
radioInterfaceSize, 495
TlvPresent, 495
nas_RSRPThresh, 496
pRSRPThresList, 497
RSRPThresListLen, 497
nas_RSRQThresh, 497
pRSRQThresList, 498
RSRQThresListLen, 498
nas_RSSIThresh, 498
pRSSIThresList, 498
RSSIThresListLen, 498
nas_RejectReasonTlv, 494
rejectCause, 494
serviceDomain, 494
TlvPresent, 494
nas_RxSigInfo, 500
isRadioTuned, 500
rsrp, 500
rxChainIndex, 500
rxPower, 501
nas_SLQSSignalStrengthsIndReq, 505
ecioDelta, 507
ecioThresholdList, 507
ecioThresholdListLen, 507
ioDelta, 507
lteRsrpDelta, 507
lteSnrDelta, 507
rsrqDelta, 507
rxSignalStrengthDelta, 507
sinrDelta, 507
sinrThresholdList, 507
sinrThresholdListLen, 507
nas_SLQSSignalStrengthsInformation, 507
ecioInfo, 508
errorRateInfo, 508
io, 508
lteRsrpinfo, 508
lteSnrinfo, 508
rsrqInfo, 508
rxSignalStrengthInfo, 508

sinr, 508
nas_SLQSSignalStrengthsTlv, 508
sSLQSSignalStrengthsInfo, 508
TlvPresent, 508
nas_SccRxInfo, 501
numInstances, 502
rsrq, 502
sigInfo, 502
snr, 502
TlvPresent, 502
nas_SignalStrengthTlv, 505
radioInterface, 505
signalStrength, 505
TlvPresent, 505
nas_SrvStatusInfo, 509
isPrefDataPath, 509
srvStatus, 509
nas_TDSCDMAECIOThresh, 512
nas_TDSCDMARSCPThresh, 512
nas_TDSCDMARSSIThresh, 513
nas_TDSCDMASINRCONFThresh, 513
nas_TDSCDMASINRThresh, 514
nas_UMTSInfo, 517
cellID, 518
ecio, 519
geranInst, 519
GeranInstInfo, 519
lac, 519
plmn, 519
psc, 519
rscp, 519
UMTSInstInfo, 519
uarfcn, 519
umtsInst, 519
nas_UMTSinstInfo, 519
umtsEcio, 519
umtsPsc, 519
umtsRscp, 520
umtsUarfcn, 520
nas_UniversalTime, 521
day, 522
dayOfWeek, 522
hour, 522
minute, 522
month, 522
second, 522
year, 522
nas_WCDMAECIOThresh, 523
nas_WCDMAInfoLTENeighborCell, 523
umtsLTENbrCellLen, 524
wcdmaRRCState, 524
nas_WCDMARSSIThresh, 524
nas_WCDMASysInfo, 525
cellId, 527
cellIdValid, 527
hsCallStatus, 527
hsCallStatusValid, 527
hsInd, 528
hsIndValid, 528
lac, 528
lacValid, 528
MCC, 528
MNC, 528
networkIdValid, 528
psc, 528
pscValid, 528
regRejectInfoValid, 528
rejCause, 528
rejectSrvDomain, 528
sysInfoWCDMA, 528
nas_acqOrderPref, 415
acqOrdeLen, 415
pAcqOrder, 415
nas_callBarStatus, 418
csBarStatus, 418
psBarStatus, 418
nas_cellParams, 425
pci, 426
rsrp, 426
rsrq, 426
rssi, 426
srxlev, 426
nas_currentPLMN, 429
MCC, 430
MNC, 430
netDescr, 430
netDescrLength, 430
nas_dataSrvCapabilities, 430
dataCapabilities, 431
dataCapabilitiesLen, 431
nas_detailSvcInfo, 433
hdrHybrid, 434
hdrSrvStatus, 434
isSysForbidden, 434
srvCapability, 434
srvStatus, 434
nas_dirNum, 435
dirNum, 435
dirNumLen, 435
nas_ecioListElement, 436
ecio, 436
radioIf, 436
nas_errorRateListElement, 437
errorRate, 438
radioIf, 438
nas_geranInstInfo, 440
geranArfcn, 440
geranBsicBcc, 440
geranBsicNcc, 441
geranRssi, 441
nas_gsmCellInfo, 441
arfcn, 441
band1900, 442
bsicId, 442
cellIdValid, 442
rssi, 442

srxlev, 442
nas_homeSIDNID, 451
numInstances, 451
SidNid, 451
nas_infoInterFreq, 452
cell_resel_priority, 452
cellInterFreqParams, 452
cells_len, 452
earfcn, 452
threshXHigh, 453
threshXLow, 453
nas_lteEARFCN, 453
earfcn0, 454
earfcn1, 454
status, 454
nas_lteGsmCellInfo, 454
cellReselPriority, 455
cells_len, 455
GsmCellInfo, 455
nccPermitted, 455
threshGsmHigh, 455
threshGsmLow, 455
nas_ltePCI, 462
earfcn, 462
pci, 462
status, 462
nas_lteRsrpinformation, 462
rsrplevel, 463
nas_lteSnrinformation, 466
snrlevel, 466
nas_lteWcdmaCellInfo, 470
cellReselPriority, 471
cellsLen, 471
threshXhigh, 471
threshXlow, 471
uarfcn, 471
WCDMACellInfo, 471
nas_major
LteNasReleaseInfo_s, 397
NASLteNasReleaseInfoTlv, 544
nas_minBasedIMSI, 471
imsiM1112, 472
imsiMS1, 472
imsiMS2, 472
mccM, 472
nas_minor
LteNasReleaseInfo_s, 397
NASLteNasReleaseInfoTlv, 544
nas_namName, 473
namName, 473
namNameLen, 473
nas_netSelectionPref, 473
mcc, 474
mnc, 474
netReg, 474
nas_nmrCellInfo, 478
nmrArfcn, 479
nmrBsic, 479
nmrCellID, 479
nmrLac, 479
nmrPlmn, 479
nmrRxLev, 479
nas_operatorNameString, 479
PLMNName, 479
nas_operatorPLMNList, 480
numInstance, 481
PLMNData, 481
nas_protocolSubtypeElement, 489
AccessMac, 490
AuthProt, 490
ControlMac, 490
EncryptProt, 490
ForwardMac, 490
IdleState, 490
KeyExchange, 490
MultDisc, 490
PhysicalLayer, 491
ReverseMac, 491
SecProt, 491
VirtStream, 491
nas_qaQmi3Gpp2TimeZone, 491
daylightSavings, 491
leapSeconds, 491
localTimeOffset, 491
nas_release
LteNasReleaseInfo_s, 397
NASLteNasReleaseInfoTlv, 544
nas_roamIndList, 495
numInstances, 496
radioInterface, 496
roamIndicator, 496
nas_rsrqInformation, 497
radioIf, 497
rsrq, 497
nas_rxInfo, 499
ecio, 499
isRadioTuned, 499
phase, 499
rscp, 499
rsrp, 499
rxPower, 500
nas_rxSignalStrengthListElement, 501
radioIf, 501
rxSignalStrength, 501
nas_servSystem, 503
csAttachState, 504
numRadioInterfaces, 504
psAttachState, 504
radioInterface, 504
regState, 504
selNetwork, 504
nas_serviceProviderName, 502
displayCondition, 503
spn, 503
spnLength, 503
nas_sidNid, 505

nid, 505
sid, 505
nas_sysInfoCommon, 509
isSysForbidden, 511
isSysForbiddenValid, 511
roamStatus, 511
roamStatusValid, 511
srvCapability, 511
srvCapabilityValid, 511
srvDomain, 511
srvDomainValid, 511
nas_timeInfo, 514
day, 516
dayLtSavingAdj, 516
dayOfWeek, 516
hour, 516
minute, 516
month, 516
radioInterface, 516
second, 516
timeZone, 516
TlvPresent, 516
year, 516
nas_trueIMSI, 516
imsiT1112, 517
imsiTS1, 517
imsiTS2, 517
imsiTaddrNum, 517
mccT, 517
nas_txInfo, 517
isInTraffic, 517
txPower, 517
nas_umtsLTENbrCell, 520
cellIsTDD, 520
earfcn, 520
pci, 520
rsrp, 520
rsrq, 521
srxlev, 521
nas_wcdmaCellInfo, 522
cpich_ecno, 522
cpich_rscp, 522
psc, 523
srxlev, 523
nas_wcdmaUARFCN, 528
status, 528
uarfcn, 528
nasCellLocationInfoResp, 529
pCDMAInfo, 530
pGERANInfo, 530
pLTEInfoInterfreq, 530
pLTEInfoIntrafreq, 530
pLTEInfoNeighboringGSM, 530
pLTEInfoNeighboringWCDMA, 530
pUMTSCellID, 530
pUMTSInfo, 530
pWCDMAInfoLTENeighborCell, 530
nasGet3GPP2SubscriptionInfoReq, 531
namID, 531
nasGet3GPP2SubscriptionInfoResp, 531
pCDMAChannel, 532
pDirNum, 532
pHomeSIDNID, 532
pMinBasedIMSI, 532
pNAMNameInfo, 532
pTrueIMSI, 532
nasGetHDRColorCodeResp, 532
pColorCode, 532
nasGetLTECphyCa, 532
sPhyCaAggPcellInfo, 533
sPhyCaAggScellDlBw, 533
sPhyCaAggScellIndType, 533
sPhyCaAggScellIndex, 533
sPhyCaAggScellInfo, 533
NasGetLTECphyCaInfo, 533
PhyCaAggPcellInfo, 533
PhyCaAggScellDlBw, 533
PhyCaAggScellIndType, 533
PhyCaAggScellIndex, 533
PhyCaAggScellInfo, 533
nasGetLTECphyCaResp, 533
pPhyCaAggPcellInfo, 534
pPhyCaAggScellDlBw, 534
pPhyCaAggScellIndType, 534
pPhyCaAggScellIndex, 534
pPhyCaAggScellInfo, 534
nasGetSigInfoResp, 534
pCDMASSInfo, 535
pGSMSSInfo, 535
pHDRSSInfo, 535
pLTESSInfo, 535
pTDSCDMASigInfoExt, 535
pTDSCDMASigInfoRscp, 535
pWCDMASSInfo, 535
nasGetSysInfoResp, 535
pAddCDMASysInfo, 537
pAddGSMSysInfo, 537
pAddHDRSysInfo, 537
pAddLTESysInfo, 537
pAddWCDMASysInfo, 537
pCDMASrvStatusInfo, 537
pCDMASysInfo, 537
pGSMCallBarringSysInfo, 537
pGSMCipherDomainSysInfo, 537
pGSMSrvStatusInfo, 537
pGSMSysInfo, 537
pHDRSrvStatusInfo, 537
pHDRSysInfo, 537
pLTESrvStatusInfo, 537
pLTESysInfo, 537
pLTEVoiceSupportSysInfo, 537
pWCDMACallBarringSysInfo, 538
pWCDMACipherDomainSysInfo, 538
pWCDMASrvStatusInfo, 538
pWCDMASysInfo, 538
nasGetTxRxInfoReq, 538

radio_if, 538
nasGetTxRxInfoResp, 538
pRXChain0Info, 539
pRXChain1Info, 539
pTXInfo, 539
nasIndicationRegisterReq, 539
pDDTMInd, 542
pDualStandByPrefInd, 542
pErrorRateInd, 542
pHDRNewUATIAssInd, 542
pHDRSessionCloseInd, 542
pLTECphyCa, 542
pManagedRoamingInd, 542
pNetworkRejectInd, 542
pNetworkTimeInd, 542
pServingSystemInd, 542
pSignalStrengthInd, 542
pSubscriptionInfoInd, 542
pSuppressSysInfoInd, 542
pSysInfoInd, 542
pSystemSelectionInd, 542
nasInitNetworkReg, 542
pChangeDuration, 543
pMNRInfo, 543
pMncPcsDigitStatus, 543
regAction, 543
nasNetworkReject, 545
pCsgId, 548
pPlmnId, 548
radioIf, 548
rejectCause, 548
serviceDomain, 548
nasNetworkTime, 548
pDayltSavAdj, 549
pRadioInterface, 549
pTimeZone, 549
universalTime, 549
nasOperatorNameResp, 549
pNITZInformation, 550
pOperatorNameString, 550
pOperatorPLMNList, 550
pPLMNNetworkName, 550
pSrvcProviderName, 550
nasPLMNNameReq, 554
mcc, 555
mnc, 555
pMncPcsStatus, 555
nasPLMNNameResp, 555
longName, 557
longNameCI, 557
longNameEn, 557
longNameLen, 557
longNameSB, 557
shortName, 557
shortNameCI, 557
shortNameEn, 557
shortNameLen, 557
shortNameSB, 557
spn, 557
spnEncoding, 557
spnLength, 558
nasRelInfoTlv
NASQmiCbkNasSwiOTAMessageInd, 558
nasSigInfo, 562
pCDMASigInfo, 563
pGSMSigInfo, 563
pHDRSigInfo, 563
pLTESigInfo, 563
pRscp, 563
pTDSCDMASigInfoExt, 563
pWCDMASigInfo, 563
nasSwiGetChannelLockResp, 563
pLteEARFCN, 564
pLtePCI, 564
pWcdmaUARFCN, 564
NasSwiIndReg, 564
gsmUmtsDl, 565
gsmUmtsUl, 565
lteEmmDl, 565
lteEmmUl, 565
lteEsmDl, 565
lteEsmUl, 565
pRankIndicatorInd, 565
pTimer, 565
nasSwiSetChannelLockReq, 565
pLteEARFCN, 566
pLtePCI, 566
pWcdmaUARFCN, 566
nasSysInfo, 566
pAddCDMASysInfo, 568
pAddGSMSysInfo, 568
pAddHDRSysInfo, 568
pAddLTESysInfo, 568
pAddWCDMASysInfo, 568
pCDMASrvStatusInfo, 568
pCDMASysInfo, 568
pGSMCallBarringSysInfo, 568
pGSMCipherDomainSysInfo, 568
pGSMSrvStatusInfo, 568
pGSMSysInfo, 568
pHDRSrvStatusInfo, 568
pHDRSysInfo, 569
pLTESrvStatusInfo, 569
pLTESysInfo, 569
pLTEVoiceSupportSysInfo, 569
pSysInfoNoChange, 569
pWCDMACallBarringSysInfo, 569
pWCDMACipherDomainSysInfo, 569
pWCDMASrvStatusInfo, 569
pWCDMASysInfo, 569
nasTimers, 569
t3396_apn, 570
t3396_plmn_id, 570
t3396_val, 570
nccPermitted
lteGsmCellInfo, 391

nas_lteGsmCellInfo, 455
NeighborSetCnt
nas_NetworkStat1x, 476
NetworkStat1x, 574
netDescr
currentPLMN, 199
nas_currentPLMN, 430
netDescrLength
currentPLMN, 199
nas_currentPLMN, 430
netInfoLen
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
netReg
nas_netSelectionPref, 474
netSelectionPref, 571
NetSelPref
NASNetSelPreferenceTlv, 545
netSelectionPref, 570
mcc, 570
mnc, 571
netReg, 571
NetStats, 571
rx_bytes, 571
rx_errors, 571
rx_overflows, 572
rx_packets, 572
tx_bytes, 572
tx_errors, 572
tx_overflows, 572
tx_packets, 572
Network Access Service (NAS), 40
NetworkDebugResp, 572
pDataStatusDetail, 572
pDeviceConfigDetail, 572
pNetworkStat1x, 572
pNetworkStatEVDO, 573
pObjectVer, 573
NetworkID
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1113
networkID
CDMASysInfo, 174
nas_CDMASysInfo, 425
networkIdValid
CDMASysInfo, 174
GSMSysInfo, 296
LTESysInfo, 408
nas_CDMASysInfo, 425
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
networkInfoLen
unpack_wds_SLQSGetCurrDataSystemStat_t, 1243
NetworkStat1x, 573
ActSetCnt, 574
NeighborSetCnt, 574
pActPilotPNElements, 574
pNeighborSetPilotPN, 574
RX_EC_IO, 575
RX_PWR, 575
SO, 575
State, 575
TX_PWR, 575
NetworkStatEVDO, 575
MACIndex, 576
PER, 576
pSectorID, 576
PilotEnergy, 576
RX_PWR, 576
SNR, 576
SectorIDLen, 576
State, 576
NetworkType
CurrNetworkInfo, 201
currNetworkInfo, 203
wds_currNetworkInfo, 1366
NewMMTlv
unpack_sms_SetNewSMSCallback_ind_t, 1179
newMTMessageTlv, 576
MTMessageInfo, 577
TlvPresent, 577
newPINLen
uim_unblockUIMPIN, 989
unblockUIMPIN, 1022
newPINVal
uim_unblockUIMPIN, 989
unblockUIMPIN, 1022
newPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 699
voiceSetCallBarringPwdInfo, 1341
newPasswdAgain
pack_voice_SLQSVoiceSetCallBarringPassword_t, 699
voiceSetCallBarringPwdInfo, 1341
newPin
pack_dms_UIMUnblockPIN_t, 604
newPwd
newPwdData, 577
newPwdAgain
newPwdData, 577
newPwdData, 577
newPwd, 577
newPwdAgain, 577
newValue
pack_dms_UIMChangePIN_t, 600
nextHeader
LibPackTFTIDParams, 358
TFTIDParams, 953
nid
CDMAInfo, 164
nas_CDMAInfo, 421
nas_sidNid, 505

sidNid, 864
unpack_nas_GetHomeNetwork_t, 1096
NmeaPort
DcsUsbPortNames, 217
nmrArfcn
nas_nmrCellInfo, 479
nmrCellInfo, 578
nmrBsic
nas_nmrCellInfo, 479
nmrCellInfo, 578
nmrCellID
nas_nmrCellInfo, 479
nmrCellInfo, 578
nmrCellInfo, 577
nmrArfcn, 578
nmrBsic, 578
nmrCellID, 578
nmrLac, 579
nmrPlmn, 579
nmrRxLev, 579
nmrInst
GERANInfo, 252
nas_GERANInfo, 440
nmrLac
nas_nmrCellInfo, 479
nmrCellInfo, 579
nmrPlmn
nas_nmrCellInfo, 479
nmrCellInfo, 579
nmrRxLev
nas_nmrCellInfo, 479
nmrCellInfo, 579
noReplyTimer
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1281
voice_callFWInfo, 1282
Non-service specific APIs (SWI), 51
notifType
voiceSUPSNotification, 1354
Notification
_SwiAvmsEventInd, 100
notification
omaDmNotificationsTlv, 588
unpack_omaDmNotificationsTlv_t, 1143
notificationType
SMSEtwsMessage, 904
sMSEtwsMessage, 903
notused
unpack_dms_SetCrashAction_t, 1047
num_instances
_qaQmi3GPP2BroadcastCfgInfo, 71
_qaQmi3GPPBroadcastCfgInfo, 71
custSettingList, 209
DMScustSettingList, 227
sms_qaQmi3GPP2BroadcastCfgInfo, 894
sms_qaQmi3GPPBroadcastCfgInfo, 894
numApp
slotInf, 868
slotInfo, 870
uim_slotInfo, 988
numCrashes
CrashInfo, 189
crashInformation, 191
numEntries
CurrentImgList, 198
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1059
numFeatures
personalizationStatus, 737
uim_personalizationStatus, 980
numFiles
registerRefresh, 809
uim_registerRefresh, 984
NumFlows
unpack_qos_SLQSSetQosEventCallback_ind_t, 1163
numInstance
nas_operatorPLMNList, 481
nas_PLMNNetworkName, 487
operatorPLMNList, 590
PLMNNetworkName, 745
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, 231
getCallFWExtInfo, 260
getCallFWInfo, 261
getMsgWaitingInfo, 279
homeSIDNID, 307
msgWaitingInfo, 414
nas_homeSIDNID, 451
nas_roamIndList, 496
nas_SccRxInfo, 502
PCSCFFQDNAddressList, 731
PCSCFIPv4ServerAddressList, 732
roamIndList, 815
SccRxInfo, 828
sms_getMsgWaitingInfo, 889
voice_arrAlertingPattern, 1270
voice_arrAlertingType, 1271
voice_arrAlphaID, 1271
voice_arrCalledPartyNum, 1272
voice_arrCallEndReason, 1272
voice_arrCallInfo, 1273
voice_arrConnectPartyNum, 1273

voice_arrDiagInfo, 1274
voice_arrRedirPartyNum, 1274
voice_arrRemotePartyName, 1275
voice_arrRemotePartyNum, 1276
voice_arrSvcOption, 1276
voice_arrUUSInfo, 1277
voice_getCallFWExtInfo, 1294
voice_getCallFWInfo, 1295
wds_DomainNameList, 1372
wds_PCSCFFQDNAddressList, 1376
wds_PCSCFIPv4ServerAddressList, 1376
numLen
calledPartyInfo, 145
callFWExtInfo, 150
callFWInfo, 151
callingPartyInfo, 154
peerNumberInfo, 736
redirNumInfo, 808
remotePartyNum, 811
voice_callFWExtInfo, 1281
voice_callFWInfo, 1282
voice_peerNumberInfo, 1297
voice_remotePartyNum, 1300
numOfFiles
uim_refreshevent, 983
UIMRefreshEvent, 1003
numOfRoutes
sms_setRoutesReq, 899
smsSetRoutesReq, 913
numOpt
DHCPOptionList, 223
wds_DHCPLeaseOptTlv, 1367
wds_DHCPv4OptionList, 1371
WdsDHCPv4OptionList, 1388
wdsDhcpv4OptionList, 1388
numPI
peerNumberInfo, 736
voice_peerNumberInfo, 1297
NumPilots
nas_PilotSetData, 486
PilotSetData, 742
numPlan
calledPartyInfo, 145
callFWExtInfo, 150
callingPartyInfo, 154
connectNumInfo, 188
peerNumberInfo, 736
redirNumInfo, 808
voice_callFWExtInfo, 1281
voice_connectNumInfo, 1289
voice_peerNumberInfo, 1297
numPresInd
connectNumInfo, 188
voice_connectNumInfo, 1289
numQosFlow
sQosStat, 917
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
numRadioInterfaces
nas_servSystem, 504
servSystem, 837
NumRxFilters
unpack_qos_QosFlowInfo_t, 1159
numSI
peerNumberInfo, 736
voice_peerNumberInfo, 1297
numSlot
cardStatus, 156
uim_cardStatus, 972
NumSupUSBComps
unpack_dms_GetUSBComp_t, 1045
NumTxFilters
unpack_qos_QosFlowInfo_t, 1159
numType
calledPartyInfo, 145
callFWExtInfo, 150
callingPartyInfo, 154
connectNumInfo, 188
peerNumberInfo, 736
redirNumInfo, 808
voice_callFWExtInfo, 1281
voice_connectNumInfo, 1290
voice_peerNumberInfo, 1297
number
calledPartyInfo, 145
callFWExtInfo, 150
callFWInfo, 151
callingPartyInfo, 154
ECTNum, 236
peerNumberInfo, 736
redirNumInfo, 808
voice_callFWExtInfo, 1281
voice_callFWInfo, 1282
voice_peerNumberInfo, 1297
numberPlan
callFwdTypeAndPlan, 148
voice_callFwdTypeAndPlan, 1279
numberType
callFwdTypeAndPlan, 148
voice_callFwdTypeAndPlan, 1279
NxtHdrProto
unpack_qos_swiQosFilter_t, 1168

OKtoRefresh
pack_uim_SLQSUIMRefreshOK_t, 682
UIMRefreshOKReq, 1005
OMADMCancelSession
qaGobiApiOmadm.h, 1765
OMADMEnabled
unpack_swioma_SLQSOMADMGetSettings_t, 1192
OMADMGetPendingNIA
qaGobiApiOmadm.h, 1766
OMADMGetSessionInfo
qaGobiApiOmadm.h, 1766
OMADMStartSession
qaGobiApiOmadm.h, 1768
OTASPStatus

voiceOTASPStatusInfo, 1337
oddEvenInd
calledPartySubAdd, 146
voice_calledPartySubAdd, 1278
OfflineReason
unpack_dms_GetPower_t, 1042
offset
readTransparentInfo, 806
uim_readTransparentInfo, 981
oldPINLen
changeUIMPIN, 177
uim_changeUIMPIN, 973
oldPINVal
changeUIMPIN, 177
uim_changeUIMPIN, 973
oldPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 699
voiceSetCallBarringPwdInfo, 1341
oldValue
pack_dms_UIMChangePIN_t, 600
omaDmConfig
sessionInfo, 838
sessionInfoExt, 838
omaDmConfigTlv, 580
alertmsg, 580
alertmsglength, 580
state, 580
userInputReq, 580
userInputTimeout, 580
omaDmConfigTlvExt, 581
alertmsg, 583
alertmsglength, 583
state, 583
userInputReq, 583
userInputTimeout, 583
omaDmFota
sessionInfo, 838
sessionInfoExt, 838
omaDmFotaTlv, 583
description, 584
descriptionlength, 584
fwdloadsize, 584
fwloadComplete, 585
namelength, 585
package_name, 585
sessionType, 585
severity, 585
state, 585
updateCompleteStatus, 585
userInputReq, 585
userInputTimeout, 585
version, 585
versionlength, 585
omaDmFotaTlvExt, 585
description, 587
descriptionlength, 587
fumoResultCode, 587
namelength, 587
package_name, 587
packageSize, 587
receivedBytes, 587
reserved, 587
state, 587
userInputTimeout, 587
version, 587
versionlength, 587
omaDmNotifications
sessionInfo, 838
omaDmNotificationsTlv, 587
notification, 588
sessionStatus, 588
opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 618
Open Mobile Alliance Service (OMA), 47
operatingMode
dms_OperatingModeTlv, 225
OperatingModeTlv
unpack_dms_SetEventReport_ind_t, 1049
operation
depersonalizationInformation, 219
pack_pds_SetPDSDefaults_t, 649
uim_depersonalizationInformation, 974
OperationMode
unpack_dms_GetPower_t, 1042
operatorNameString, 588
PLMNName, 588
OperatorPLMNData, 589
lac1, 589
lac2, 589
mcc, 589
mnc, 589
PLMNRecID, 589
operatorPLMNList, 589
numInstance, 590
PLMNData, 590
optCode
DHCPOption, 222
wds_DHCPOpt, 1369
wds_DHCPv4Option, 1370
WdsDHCPv4Option, 1387
wdsDhcpv4Option, 1387
optList
wds_DHCPLeaseOptTlv, 1367
optListData
wds_DHCPLeaseOptTlv, 1367
optVal
wds_DHCPv4Option, 1370
WdsDHCPv4Option, 1387
wdsDhcpv4Option, 1387
optValLen
DHCPOption, 222
wds_DHCPOpt, 1369
wds_DHCPv4Option, 1370
WdsDHCPv4Option, 1387
wdsDhcpv4Option, 1387

OriginateUSSD
qaGobiApiVoice.h, 1904
otaMsgTlv
NASQmiCbkNasSwiOTAMessageInd, 559

p1Status
unpack_dms_UIMGetPINStatus_t, 1067
p1UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1067
p1VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1067
p2Status
unpack_dms_UIMGetPINStatus_t, 1067
p2UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1067
p2VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 1067
p3GPP2Pri
swiQosFlow, 938
p3GPP2TimeInfo
GetNetworkTimeResp, 279
unpack_nas_SLQSGetNetworkTime_t, 1108
p3GPPImCn
swiQosFlow, 938
p3GPPResResidualBER
swiQosFlow, 938
p3GPPSigInd
swiQosFlow, 938
p3GPPTimeInfo
GetNetworkTimeResp, 279
unpack_nas_SLQSGetNetworkTime_t, 1108
p3GPPTraHdlPri
swiQosFlow, 938
p3GppNetworkInfoInstances
unpack_nas_PerformNetworkScan_t, 1102
p3GppNetworkInstanceSize
unpack_nas_PerformNetworkScan_t, 1103
p3gppRelease
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 721
pAAASPI
pack_wds_SetMobileIPProfile_t, 714
PACK_WDS_IPV4
wds.h, 2069
PACK_WDS_IPV6
wds.h, 2069
pAMRStatus
pack_voice_SLQSVoiceGetConfig_t, 695
voiceGetConfigReq, 1329
pAPN
_SLQSAVMSSettingsAPNInfo, 86
pAPNClass
LibpackProfile3GPP, 337
LibPackprofile_3GPP, 348
Profile3GPP, 755
pAPNClass3GPP2
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pAPNDisabledFlag
LibpackProfile3GPP, 337
LibPackprofile_3GPP, 348
Profile3GPP, 755
pAPNEnabled3GPP2
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pAPNInfo
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAPNName
LibpackProfile3GPP, 337
LibPackprofile_3GPP, 348
Profile3GPP, 755
qmiWdsRunTimeSettings, 799
swiPDPRuntimeSettingsResp, 932
pAPNnameSize
LibpackProfile3GPP, 337
LibPackprofile_3GPP, 348
Profile3GPP, 755
PASSWORD_LENGTH
qaGobiApiVoice.h, 1903
pAVRXAVCHeadroom
RXAVCList, 819
pAVRXAVCSens
RXAVCList, 820
pAccelAcceptReady
QmiCbkLocSensorStreamingInd, 788
pAccelSamplesAccepted
QmiCbkLocInjectSensorDataInd, 780
pAccelTempAcceptReady
QmiCbkLocSensorStreamingInd, 788
pAccelTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 780
pAcceleroData
LocInjectSensorDataReq, 385
pAcceleroTempData
LocInjectSensorDataReq, 385
pAcceleroTimeSrc
LocInjectSensorDataReq, 385
pAccuracy
unpack_pds_GetPDSDefaults_t, 1144
pAcqOrder
acqOrderPref, 115
nas_acqOrderPref, 415
pAcqOrderPref
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
pActPilotPNElements
nas_NetworkStat1x, 476
NetworkStat1x, 574
pAddCDMASysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1118
unpack_nas_SLQSSysInfoCallback_ind_t, 1138
pAddGSMSysInfo

nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1118
unpack_nas_SLQSSysInfoCallback_ind_t, 1138
pAddHDRSysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1118
unpack_nas_SLQSSysInfoCallback_ind_t, 1138
pAddLTESysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1118
unpack_nas_SLQSSysInfoCallback_ind_t, 1138
pAddWCDMASysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1118
unpack_nas_SLQSSysInfoCallback_ind_t, 1138
pAddrAllocPref
LibpackProfile3GPP, 337
LibPackprofile_3GPP, 348
Profile3GPP, 755
pAddress
pack_wds_SetMobileIPProfile_t, 714
pAirTimer
pack_voice_SLQSVoiceGetConfig_t, 695
voiceGetConfigReq, 1329
pAirTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1221
voiceGetConfigResp, 1331
pAirTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 701
voiceSetConfigReq, 1343
pAirTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1225
voiceSetConfigResp, 1345
pAlertPriority
cdmaMsgDecodingParams, 166
pAlertType
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pAlertingPattern
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pAllowLinger
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pAlphaID
SMSAsyncRawSend_s, 901
pAlphaIDInfo
unpack_voice_SLQSOriginateUSSD_t, 1202
unpack_voice_SLQSVoiceDialCall_t, 1203
unpack_voice_SLQSVoiceGetCallBarring_t, 1208
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1209
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
unpack_voice_SLQSVoiceGetCallWaiting_t, 1213
unpack_voice_SLQSVoiceGetCLIP_t, 1215
unpack_voice_SLQSVoiceGetCLIR_t, 1216
unpack_voice_SLQSVoiceGetCNAP_t, 1217
unpack_voice_SLQSVoiceGetCOLP_t, 1218
unpack_voice_SLQSVoiceGetCOLR_t, 1220
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1223
unpack_voice_SLQSVoiceSetSUPSService_t, 1226
USSResp, 1264
voiceCallInfoResp, 1308
voiceCallResponseParams, 1311
voiceGetCallBarringResp, 1317
voiceGetCallFWResp, 1320
voiceGetCallWaitInfo, 1321
voiceGetCLIPResp, 1322
voiceGetCLIRResp, 1323
voiceGetCNAPResp, 1325
voiceGetCOLPResp, 1326
voiceGetCOLRResp, 1327
voiceSetCallBarringPwdResp, 1342
voiceSetSUPSServiceResp, 1349
voiceSUPSInfo, 1352
pAlphaIdentifier
USSDNoWaitIndicationInfo, 1263
pAltitudeAssumed
QmiCbkLocPositionReportInd, 786
unpack_loc_PositionRpt_Ind_t, 1086
pAltitudeSrcInfo
LocInjectPositionReq, 383
pAltitudeWrtEllipsoid
LocInjectPositionReq, 383
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 733
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 786
unpack_loc_BestAvailPos_Ind_t, 1075
unpack_loc_PositionRpt_Ind_t, 1086
pAltitudeWrtMeanSeaLevel
LocInjectPositionReq, 383
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 786
unpack_loc_BestAvailPos_Ind_t, 1075
unpack_loc_PositionRpt_Ind_t, 1086
pAltitudeWrtSealevel
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 733
pAmrMode
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 325
SetIMSVoIPConfigReq, 849
pAmrOctetAligned
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 325
SetIMSVoIPConfigReq, 849
pAmrWBMode
GetIMSVoIPConfigResp, 273

imsVoIPConfigInfo, 325
SetIMSVoIPConfigReq, 849
pAmrWBOctetAligned
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 325
SetIMSVoIPConfigReq, 849
pAmrWbEnable
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 325
SetIMSVoIPConfigReq, 849
pApnString
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pApnStringSize
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pApnname
pack_wds_SetDefaultProfile_t, 710
pAppName
loc_LocApplicationInfo, 368
LocApplicationInfo, 376
pAppPriority
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pAppProvider
loc_LocApplicationInfo, 368
LocApplicationInfo, 376
pAppSubType
HDRProtSubtypResp, 300
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1133
pAppType
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pAppVersion
loc_LocApplicationInfo, 368
LocApplicationInfo, 376
pApplication
pack_nas_SetCDMANetworkParameters_t, 623
pApplicationInfo
LOCStartReq, 387
pack_loc_Start_t, 620
pArrAlertingPattern
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrAlertingType
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrAlphaID
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrCallEndReason
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrCallInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
pArrCalledPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrConnectPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrDiagInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrRedirPartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrRemotePartyName
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrRemotePartyNum
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrSvcOption
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1206
voiceGetAllCallInfo, 1315
voiceSetAllCallStatusCbkInfo, 1340
pArrUUSInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1207
voiceGetAllCallInfo, 1316
pAuth
pack_wds_SLQSStartDataSession_t, 725
pAuthPassword
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pAuthPassword_tSize
LibPackprofile_3GPP2, 354
pAuthPasswordSize
LibpackProfile3GPP2, 343
Profile3GPP2, 760
pAuthProtocol
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 760
pAuthRetryCount
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pAuthTimeout

LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pAuthenticateResult
UIMAuthenticateResp, 992
unpack_uim_SLQSUIMAuthenticate_t, 1196
pAuthentication
qmiWdsRunTimeSettings, 799
ssdatasession_params, 920
pAuthenticationPref
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pAutoAnsStatus
unpack_voice_SLQSVoiceSetConfig_t, 1225
voiceSetConfigResp, 1345
pAutoAnswer
pack_voice_SLQSVoiceGetConfig_t, 695
pack_voice_SLQSVoiceSetConfig_t, 701
voiceGetConfigReq, 1329
voiceSetConfigReq, 1344
pAutoAnswerStat
unpack_voice_SLQSVoiceGetConfig_t, 1221
voiceGetConfigResp, 1331
pAutoConnect
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAutoReboot
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pAutoSelection
UIMGetConfigurationResp, 996
unpack_uim_SLQSUIMGetConfiguration_t, 1198
pAutosdm
_SLQSOMADMSettings, 93
_SLQSOMADMSettingsReqParams, 94
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 673
pBandPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pBdsSVInfo
LocDelAssDataReq, 376
pack_loc_Delete_Assist_Data_t, 607
pBearerID
QosFlowInfo, 803
pBearerId
swiPDPRuntimeSettingsResp, 933
pBearerTech
DataBearerTechExt, 212
pBroadcast
pack_nas_SetCDMANetworkParameters_t, 623
pBroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 668
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1182
pBurstDTMFLengths
pack_voice_SLQSVoiceBurstDTMF_t, 689
voiceBurstDTMFInfo, 1305
pCCResType
unpack_voice_SLQSVoiceGetCallBarring_t, 1208
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1209
unpack_voice_SLQSVoiceGetCallWaiting_t, 1213
unpack_voice_SLQSVoiceGetCLIP_t, 1215
unpack_voice_SLQSVoiceGetCLIR_t, 1216
unpack_voice_SLQSVoiceGetCNAP_t, 1217
unpack_voice_SLQSVoiceGetCOLP_t, 1218
unpack_voice_SLQSVoiceGetCOLR_t, 1220
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1223
voiceGetCallBarringResp, 1318
voiceGetCallFWResp, 1320
voiceGetCallWaitInfo, 1321
voiceGetCLIPResp, 1322
voiceGetCLIRResp, 1324
voiceGetCNAPResp, 1325
voiceGetCOLPResp, 1326
voiceGetCOLRResp, 1327
voiceSetCallBarringPwdResp, 1342
pCCResultType
unpack_voice_SLQSVoiceDialCall_t, 1204
unpack_voice_SLQSVoiceSetSUPSService_t, 1226
voiceCallResponseParams, 1311
voiceSetSUPSServiceResp, 1349
pCCSUPSType
unpack_voice_SLQSVoiceDialCall_t, 1204
unpack_voice_SLQSVoiceGetCallBarring_t, 1208
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1209
unpack_voice_SLQSVoiceGetCallWaiting_t, 1213
unpack_voice_SLQSVoiceGetCLIP_t, 1215
unpack_voice_SLQSVoiceGetCLIR_t, 1216
unpack_voice_SLQSVoiceGetCNAP_t, 1217
unpack_voice_SLQSVoiceGetCOLP_t, 1218
unpack_voice_SLQSVoiceGetCOLR_t, 1220
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1223
unpack_voice_SLQSVoiceSetSUPSService_t, 1226
voiceCallResponseParams, 1311
voiceGetCallBarringResp, 1318
voiceGetCallFWResp, 1320
voiceGetCallWaitInfo, 1321
voiceGetCLIPResp, 1322
voiceGetCLIRResp, 1324
voiceGetCNAPResp, 1325
voiceGetCOLPResp, 1326
voiceGetCOLRResp, 1327
voiceSetCallBarringPwdResp, 1342
voiceSetSUPSServiceResp, 1349
pCCSuppsType
unpack_voice_SLQSOriginateUSSD_t, 1202

USSResp, 1264
pCDMABroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 668
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1182
pCDMAChannel
nasGet3GPP2SubscriptionInfoResp, 532
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1124
pCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pCDMAECIOThreshList
CDMAECIOThresh, 163
nas_CDMAECIOThresh, 420
pCDMAFrameErrRate
GetErrRateResp, 268
unpack_nas_SLQSGetErrorRate_t, 1107
pCDMAInfo
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pCDMARSSIThreshList
CDMARSSIThresh, 169
nas_CDMARSSIThresh, 421
pCDMASSInfo
nasGetSigInfoResp, 535
pCDMASigInfo
nasSigInfo, 563
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1129
pCDMASrvStatusInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1138
pCDMASysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1138
pCLIPResp
unpack_voice_SLQSVoiceGetCLIP_t, 1215
voiceGetCLIPResp, 1322
pCLIPstatus
voiceSUPSInfo, 1352
pCLIRCause
voiceInfoRec, 1334
pCLIRResp
unpack_voice_SLQSVoiceGetCLIR_t, 1216
voiceGetCLIRResp, 1324
pCLIRType
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
pCLIRstatus
voiceSUPSInfo, 1352
PCMparams, 730
iFaceTab, 730
iFaceTabLen, 730
pCNAPResp
unpack_voice_SLQSVoiceGetCNAP_t, 1217
voiceGetCNAPResp, 1325
pCNAPstatus
voiceSUPSInfo, 1352
pCOLPResp
unpack_voice_SLQSVoiceGetCOLP_t, 1219
voiceGetCOLPResp, 1326
pCOLPstatus
voiceSUPSInfo, 1352
pCOLRResp
unpack_voice_SLQSVoiceGetCOLR_t, 1220
voiceGetCOLRResp, 1327
pCOLRstatus
voiceSUPSInfo, 1352
PCSCFAddrPCO
unpack_wds_SLQSGetRuntimeSettings_t, 1250
PCSCFFQDNAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1250
PCSCFFQDNAddress, 730
fqdnAddr, 731
fqdnLen, 731
PCSCFFQDNAddressList, 731
numInstances, 731
pcsfFQDNAddress, 731
PCSCFIPv4ServerAddressList, 731
numInstances, 732
pscsfIPv4Addr, 732
pCSCFPortName
imsRegMgrConfigInfo, 321
SetRegMgrConfigReq, 856
pCSCFPortNameLen
SetRegMgrConfigReq, 856
pCSGID
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
pCUGIndex
voiceSUPSNotification, 1354
pCUGInfo
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
pCallBarPasswd
voiceSUPSInfo, 1352
pCallBarringPasswd
pack_voice_SLQSVoiceSetSUPSService_t, 703
voiceSetSUPSServiceReq, 1348
pCallEndReason
getDUNCallInfoResp, 266

pCallFWNum
voiceSUPSInfo, 1352
pCallFWTimerVal
voiceSUPSInfo, 1352
pCallForwardingNumber
pack_voice_SLQSVoiceSetSUPSService_t, 703
voiceSetSUPSServiceReq, 1348
pCallFwdInfo
voiceSUPSInfo, 1352
pCallFwdTypeAndPlan
pack_voice_SLQSVoiceSetSUPSService_t, 703
voiceSetSUPSServiceReq, 1348
pCallID
burstDTMFInfo, 142
pack_voice_SLQSVoiceManageCalls_t, 697
pack_voice_SLQSVoiceSendFlash_t, 698
pack_voice_SLQSVoiceStartContDTMF_t, 704
unpack_voice_SLQSVoiceBurstDTMF_t, 1203
unpack_voice_SLQSVoiceDialCall_t, 1203
unpack_voice_SLQSVoiceGetCallBarring_t, 1208
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1209
unpack_voice_SLQSVoiceGetCallWaiting_t, 1213
unpack_voice_SLQSVoiceGetCLIP_t, 1215
unpack_voice_SLQSVoiceGetCLIR_t, 1216
unpack_voice_SLQSVoiceGetCNAP_t, 1217
unpack_voice_SLQSVoiceGetCOLP_t, 1218
unpack_voice_SLQSVoiceGetCOLR_t, 1220
unpack_voice_SLQSVoiceSendFlash_t, 1222
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1223
unpack_voice_SLQSVoiceSetSUPSService_t, 1226
unpack_voice_SLQSVoiceStartContDTMF_t, 1227
voice_burstDTMFInfo, 1277
voiceCallResponseParams, 1311
voiceContDTMFinfo, 1311
voiceFlashInfo, 1313
voiceGetCallBarringResp, 1318
voiceGetCallFWResp, 1320
voiceGetCallWaitInfo, 1321
voiceGetCLIPResp, 1322
voiceGetCLIRResp, 1323
voiceGetCNAPResp, 1325
voiceGetCOLPResp, 1326
voiceGetCOLRResp, 1327
voiceManageCallsReq, 1335
voiceSetCallBarringPwdResp, 1342
voiceSetSUPSServiceResp, 1349
voiceSUPSInfo, 1352
pCallId
pack_voice_SLQSVoiceAnswerCall_t, 688
pack_voice_SLQSVoiceEndCall_t, 691
unpack_voice_SLQSOriginateUSSD_t, 1202
unpack_voice_SLQSVoiceAnswerCall_t, 1202
unpack_voice_SLQSVoiceEndCall_t, 1204
USSResp, 1264
voiceAnswerCall, 1304
pCallInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pCallPartySubAdd
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
pCallType
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
pCallWaitInd
voiceInfoRec, 1334
pCallbackAddr
cdmaMsgEncodingParams, 168
pCallbkAddr
cdmaMsgDecodingParams, 166
pCallbkAddrLength
cdmaMsgDecodingParams, 166
pCalledPartyInfo
voiceInfoRec, 1334
pCallerIDInfo
voiceInfoRec, 1334
pCallerNameInfo
voiceInfoRec, 1334
pCallingPartyInfo
voiceInfoRec, 1334
pCardResult
UIMAuthenticateResp, 992
UIMGetFileAttributesResp, 997
UIMReadTransparentResp, 1001
unpack_uim_ReadTransparent_t, 1194
unpack_uim_SLQSUIMAuthenticate_t, 1196
unpack_uim_SLQSUIMGetFileAttributes_t, 1199
pCardStatus
UIMGetCardStatusResp, 994
unpack_uim_GetCardStatus_t, 1194
unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind_t, 1200
pCcResultType
unpack_voice_SLQSOriginateUSSD_t, 1202
USSResp, 1264
pCellDataMask
pack_pds_ResetPDSData_t, 647
pCellDb
LocDelAssDataReq, 376
pack_loc_Delete_Assist_Data_t, 607
pChangeDuration
nasInitNetworkReg, 543
pack_nas_SLQSInitiateNetworkRegistration_t, 627
pChannelRate
getDUNCallInfoResp, 266
pChgDuration
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
pClkInfo
LocDelAssDataReq, 376
pack_loc_Delete_Assist_Data_t, 607
pCodecSTGain
GetAudioPathConfigResp, 256

SetAudioPathConfigReq, 840
pColorCode
nasGetHDRColorCodeResp, 532
unpack_nas_SLQSNasGetHDRColorCode_t, 1126
pConfidence
LocSetCradleMountReq, 386
pConfigAltitudeAssumed
LOCStartReq, 387
pack_loc_Start_t, 620
pConfigurationMask
pack_uim_SLQSUIMGetConfiguration_t, 679
UIMGetConfigurationReq, 995
pConnectNumInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
voiceInfoRec, 1334
pConnectionRetryTimers
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pConnectionStatus
getDUNCallInfoResp, 266
pContextId
swiPDPRuntimeSettingsResp, 933
pCrashInfo
CrashInfoParams, 190
pCrashString
CrashInfo, 189
pCreateProfileOut
unpack_wds_SLQSCreateProfile_t, 1241
pCsgId
nasNetworkReject, 548
pCurAMRConfig
unpack_voice_SLQSVoiceGetConfig_t, 1221
voiceGetConfigResp, 1331
pCurDataBearerTechnology
dataBearers, 209
pCurPrefVoiceSO
unpack_voice_SLQSVoiceGetConfig_t, 1221
voiceGetConfigResp, 1331
pCurProfile
pack_wds_SLQSCreateProfile_t, 715
pCurVoiceDomainPref
unpack_voice_SLQSVoiceGetConfig_t, 1221
voiceGetConfigResp, 1331
pCurVoicePrivacyPref
unpack_voice_SLQSVoiceGetConfig_t, 1222
voiceGetConfigResp, 1331
pCurrChannelRateInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1396
pCurrDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1396
pCurrImgInfo
CurrentImgList, 198
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1059
pCurrNetworkInfo
CurrDataSysStat, 197
pCurrPrefDataSysInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1396
pCurrTTYMode
unpack_voice_SLQSVoiceGetConfig_t, 1221
voiceGetConfigResp, 1331
pCurrentChannelRXRate
WdsConnectionRateElmnts, 1384
pCurrentChannelTXRate
WdsConnectionRateElmnts, 1384
pCurrentPersonality
HDRPersonalityInd, 299
HDRPersonalityResp, 299
unpack_nas_SLQSSwiGetHDRPersonality_t, 1133
pCurrentPrsnlty
HDRProtSubtypResp, 300
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1133
pCurrentmitigationLvl
TmdGetMitigationLvlResp, 957
pCustSettingInfo
DMSgetCustomFeatureV2, 228
getCustomFeatureV2, 261
pCustSettingList
DMSgetCustomFeatureV2, 228
getCustomFeatureV2, 261
pCustomSCP
pack_nas_SetCDMANetworkParameters_t, 623
pCwtMute
SetM2MAudioProfileReq, 852
SetM2MAVMuteReq, 853
pDDTMInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 635
pDHCPRelayEnabled
custFeaturesInfo, 205
custFeaturesSetting, 207
PDOP
loc_precisionDilution, 369
precisionDilution_s, 748
PDPType
unpack_wds_SLQSGetRuntimeSettings_t, 1250
pDRCParams
GetHRPDStatsResp, 269
unpack_nas_SLQSSwiGetHRPDStats_t, 1134
PDS_SRV
qaGobiApiCbk.h, 1558
PDSInjectTimeReference
qaGobiApiPds.h, 1774
PDSPosMethodStateReq, 734
pWifiState, 734
pXtraDataState, 734
pXtraTimeState, 735
PDSPositionData, 732
pAltitudeWrtEllipsoid, 733
pAltitudeWrtSealevel, 733
pHorizontalConfidence, 733
pHorizontalUncCircular, 733

pLatitude, 734
pLongitude, 734
pPositionSource, 734
pTimeStamp, 734
pTimeType, 734
pVerticalConfidence, 734
pVerticalUnc, 734
pDTMFTXGain
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 840
pDataBearer
QosEventInfo, 802
slqsWdsEventInfo, 886
unpack_wds_GetDataBearerTechnology_t, 1230
pDataBearerTech
getDUNCallInfoResp, 266
pDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1396
pDataCallStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1397
pDataMode
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pDataRate
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
swiQosFlow, 938
pDataSrc
voiceSUPSInfo, 1352
pDataStatusDetail
NetworkDebugResp, 572
unpack_nas_SLQSSwiNetworkDebug_t, 1136
pDataSystemStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1397
pDate
_SLQSOMADMSessionInfo, 91
pDateLength
_SLQSOMADMSessionInfo, 91
pDayltSavAdj
nasNetworkTime, 549
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1128
pDefaultPDNEnabled
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 721
pDeferTime
pack_swioma_SLQSOMADMSendSelection_t, 672
pDesription
QmiNas3GppNetworkInfo, 793
pDestAddr
cdmaMsgEncodingParams, 168
wcdmaMsgEncodingParams, 1360
pDestSMSContent
getDyingGaspCfg, 267
pack_dms_SLQSSwiSetDyingGaspCfg_t, 598
packgetDyingGaspCfg, 729
setDyingGaspCfg, 845
pDestSMSNum
getDyingGaspCfg, 267
pack_dms_SLQSSwiSetDyingGaspCfg_t, 598
packgetDyingGaspCfg, 729
setDyingGaspCfg, 845
pDetachAction
pack_nas_SLQSSwiPSDetach_t, 646
PSDetachReq, 764
pDevCrashStatus
CrashInfoParams, 190
pDeviceConfigDetail
NetworkDebugResp, 572
unpack_nas_SLQSSwiNetworkDebug_t, 1136
pDiagInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pDigitBuff
burstDTMFInfo, 142
voice_burstDTMFInfo, 1277
pDirNum
nasGet3GPP2SubscriptionInfoResp, 532
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1124
pDisableIMSI
custFeaturesInfo, 205
pDispInfo
voiceInfoRec, 1334
pDisplayMode
cdmaMsgDecodingParams, 166
pDomainList
qmiWdsRunTimeSettings, 800
pDormancyStatus
getDUNCallInfoResp, 266
slqsWdsEventInfo, 886
pDormancyStatusInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1397
pDualStandByPrefInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 635
pDuration
unpack_pds_GetXTRAValidity_t, 1147
pECIOThresList
ECIOThresh, 235
nas_ECIOThresh, 437
pECIOThresh
pack_nas_SLQSConfigSigInfo_t, 626
sigInfo, 865
pECMode
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pECTNum
voiceSUPSNotification, 1354

PER
nas_NetworkStatEVDO, 478
NetworkStatEVDO, 576
pESNString
serialNumbersInfo, 833
pEVDOPageMonPerChangeInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1397
pEarMute
SetM2MAudioProfileReq, 852
pEmerMode
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pEmergencyCategory
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
pEnableNotification
pack_wds_DHCPv4ClientLeaseChange_t, 705
WdsClientLeaseChange, 1382
pEnabled
pack_wds_SetMobileIPProfile_t, 714
pEnabledStatus
unpack_pds_GetPDSState_t, 1144
pEncodingAlphabet
cdmaMsgEncodingParams, 168
pEncryptData
pack_uim_ReadTransparent_t, 676
UIMReadTransparentReq, 1000
pEncryptedData
UIMReadTransparentResp, 1001
unpack_uim_ReadTransparent_t, 1194
pEncryptedPIN1
pack_uim_VerifyPin_t, 685
UIMPinResp, 998
UIMVerifyPinReq, 1012
unpack_uim_ChangePin_t, 1193
unpack_uim_SetPinProtection_t, 1195
unpack_uim_UnblockPin_t, 1201
unpack_uim_VerifyPin_t, 1201
pError
USSDNoWaitIndicationInfo, 1263
pErrorCodeStr
imsaRatStatusInfo, 313
pErrorMask
CATEventDataType, 159
pErrorRateInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pEspSpi
swiQosFilter, 935
pEtwsMessageInfo
SMSEventInfo_s, 906
pEtwsPlmnInfo
SMSEventInfo_s, 906
pExtDispInfo
voiceInfoRec, 1334
pExtDispRecInfo
voiceInfoRec, 1334
pExtErrCode
_GetProfileSettingOut, 65
UnPackGetProfileSettingOut, 1258
pExtErrorCode
CreateProfileOut, 193
ModifyProfileOut, 413
unpack_wds_SLQSModifyProfile_t, 1251
pFOTAUpdate
_SLQSOMADMSettings, 93
pFOTAdownload
_SLQSOMADMSettings, 93
pFailCause
unpack_voice_SLQSVoiceGetCallBarring_t, 1208
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1209
unpack_voice_SLQSVoiceGetCallWaiting_t, 1213
unpack_voice_SLQSVoiceGetCLIP_t, 1215
unpack_voice_SLQSVoiceGetCLIR_t, 1216
unpack_voice_SLQSVoiceGetCNAP_t, 1217
unpack_voice_SLQSVoiceGetCOLP_t, 1219
unpack_voice_SLQSVoiceGetCOLR_t, 1220
unpack_voice_SLQSVoiceManageCalls_t, 1222
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 1224
unpack_voice_SLQSVoiceSetSUPSService_t, 1226
voiceGetCallBarringResp, 1318
voiceGetCallFWResp, 1320
voiceGetCallWaitInfo, 1321
voiceGetCLIPResp, 1322
voiceGetCLIRResp, 1324
voiceGetCNAPResp, 1325
voiceGetCOLPResp, 1326
voiceGetCOLRResp, 1327
voiceManageCallsResp, 1336
voiceSetCallBarringPwdResp, 1342
voiceSetSUPSServiceResp, 1349
voiceSUPSInfo, 1352
pFailErrorCode
imsaPdpStatusInfo, 313
pFailureCause
USSDNoWaitIndicationInfo, 1263
pFailureReason
unpack_wds_SLQSStartDataSession_t, 1256
pFile
ERIFileparams, 238
pFileAttributes
UIMGetFileAttributesResp, 997
unpack_uim_SLQSUIMGetFileAttributes_t, 1199
pFileSize
ERIFileparams, 238
pFixId
QmiCbkLocPositionReportInd, 786
unpack_loc_PositionRpt_Ind_t, 1086
pFlag
RXPCMIIRFltr, 822

TXPCMIIRFltr, 965
pFlashPayLd
pack_voice_SLQSVoiceSendFlash_t, 698
voiceFlashInfo, 1313
pFlashType
pack_voice_SLQSVoiceSendFlash_t, 698
voiceFlashInfo, 1313
pFollowOnDC
slqssendasyncsmsparams_s, 877
sms_sendAsyncsmsParams, 897
pForbidden
QmiNas3GppNetworkInfo, 794
pForceOnDC
slqssendasyncsmsparams_s, 877
sms_sendAsyncsmsParams, 897
pForceRev0
pack_nas_SetCDMANetworkParameters_t, 624
pFwAutoCheck
_SLQSOMADMSettings, 93
_SLQSOMADMSettingsReqParams3, 95
pack_swioma_SLQSOMADMSetSettings_t, 673
pFwAutoSDM
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pFwAutoUpdate
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pFwAutodownload
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 85
pGCDumpString
CrashInfo, 189
pGERANInfo
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pGPRSGrantedQoS
qmiWdsRunTimeSettings, 800
pGPRSMinimumQoS
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pGPRSRequestedQos
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pGPSDataMask
pack_pds_ResetPDSData_t, 647
pGPSEnable
custFeaturesSetting, 207
pGPSLPM
custFeaturesInfo, 205
custFeaturesSetting, 207
pGPSSel
custFeaturesInfo, 205
custFeaturesSetting, 207
pGPSWeek
unpack_pds_GetXTRAValidity_t, 1147
pGPSWeekOffset
unpack_pds_GetXTRAValidity_t, 1147
pGSMBER
GetErrRateResp, 268
unpack_nas_SLQSGetErrorRate_t, 1107
pGSMCallBarringSysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pGSMCipherDomainSysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pGSMRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pGSMRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pGSMRSSIThreshList
GSMRSSIThresh, 292
nas_GSMRSSIThresh, 442
pGSMSSInfo
nasGetSigInfoResp, 535
pGSMSigInfo
nasSigInfo, 563
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1129
pGSMSrvStatusInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pGSMSysInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pGWAcqOrderPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pGWAddressV4
qmiWdsRunTimeSettings, 800
pGenerator
GetM2MAudioProfileReq, 274
SetM2MAudioProfileReq, 852
pGetCallFWExtInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1209
voiceGetCallFWResp, 1320
pGetCallFWInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 1209
voiceGetCallFWResp, 1320

pGetCustomInput
DMSgetCustomFeatureV2, 228
getCustomFeatureV2, 261
pGetDyingGaspCfg
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1058
pGetDyingGaspStatistics
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1058
pGetIndicationRegInfo
unpack_sms_SLQSGetIndicationRegister_t, 1180
pGetMsgWaitingInfoResp
unpack_sms_SLQSGetMessageWaiting_t, 1180
pGetTransLayerInfo
unpack_sms_SLQSGetTransLayerInfo_t, 1183
pGetTransNWRegInfo
unpack_sms_SLQSGetTransNWRegInfo_t, 1183
pGnssData
LocDelAssDataReq, 376
pack_loc_Delete_Assist_Data_t, 607
pGpsTime
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 786
unpack_loc_BestAvailPos_Ind_t, 1075
unpack_loc_PositionRpt_Ind_t, 1086
pGyroAcceptReady
QmiCbkLocSensorStreamingInd, 788
pGyroData
LocInjectSensorDataReq, 385
pGyroSamplesAccepted
QmiCbkLocInjectSensorDataInd, 780
pGyroTempAcceptReady
QmiCbkLocSensorStreamingInd, 788
pGyroTempData
LocInjectSensorDataReq, 385
pGyroTempSamplesAccepted
QmiCbkLocInjectSensorDataInd, 780
pGyroTimeSrc
LocInjectSensorDataReq, 385
pHA2002bis
pack_wds_SetMobileIPParameters_t, 712
pHAAuthenticator
pack_wds_SetMobileIPParameters_t, 712
pHASPI
pack_wds_SetMobileIPProfile_t, 714
pHDRECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pHDRECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pHDRECIOThreshList
HDRECIOThresh, 298
nas_HDRECIOThresh, 446
pHDRIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pHDRIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pHDRIOThreshList
HDRIOThresh, 298
nas_HDRIOThresh, 447
pHDRNewUATIAssInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pHDRPackErrRate
GetErrRateResp, 268
unpack_nas_SLQSGetErrorRate_t, 1107
pHDRRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pHDRRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pHDRRSSIThreshList
HDRRSSIThresh, 301
nas_HDRRSSIThresh, 447
pHDRSINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pHDRSINRThresList
HDRSINRThresh, 301
nas_HDRSINRThresh, 448
pHDRSINRThresh
pack_nas_SLQSConfigSigInfo_t, 626
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
sigInfo, 865
pHDRSINRThreshList
HDRSINRThreshold, 302
nas_HDRSINRThreshold, 449
pHDRSSInfo
nasGetSigInfoResp, 535
pHDRSessionCloseInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pHDRSigInfo
nasSigInfo, 563
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1129
pHDRSrvStatusInfo
nasGetSysInfoResp, 537
nasSysInfo, 568
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pHDRSysInfo
nasGetSysInfoResp, 537
nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pHaltSubscription
UIMGetConfigurationResp, 996
unpack_uim_SLQSUIMGetConfiguration_t, 1198
pHeading
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 786

unpack_loc_BestAvailPos_Ind_t, 1075
unpack_loc_PositionRpt_Ind_t, 1086
pHeadingUnc
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1075
unpack_loc_PositionRpt_Ind_t, 1087
pHomeSIDNID
nasGet3GPP2SubscriptionInfoResp, 532
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1124
pHorCirConf
QmiCbkLocBestAvailPosInd, 774
unpack_loc_BestAvailPos_Ind_t, 1075
pHorConfidence
LocInjectPositionReq, 383
QmiCbkLocPositionReportInd, 787
unpack_loc_PositionRpt_Ind_t, 1087
pHorEllpConf
QmiCbkLocBestAvailPosInd, 774
unpack_loc_BestAvailPos_Ind_t, 1075
pHorReliability
LocInjectPositionReq, 384
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1075
unpack_loc_PositionRpt_Ind_t, 1087
pHorUncCircular
LocInjectPositionReq, 384
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pHorUncEllipseOrientAzimuth
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pHorUncEllipseSemiMajor
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pHorUncEllipseSemiMinor
QmiCbkLocBestAvailPosInd, 774
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pHorizontalAccuracyLvl
LOCStartReq, 387
pack_loc_Start_t, 621
pHorizontalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 733
pHorizontalUncCircular
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 733
pHotSwapStatus
UIMGetCardStatusResp, 994
unpack_uim_GetCardStatus_t, 1194
pHwConfig
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 724
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1254
WdsDHCPv4Config, 1385
PI
calledPartyInfo, 146
callerIDInfo, 147
callFWExtInfo, 150
callingPartyInfo, 154
redirNumInfo, 808
voice_callFWExtInfo, 1281
PIFACEId
SetM2MAudioAVCFGReq, 850
pIMCNflag
qmiWdsRunTimeSettings, 800
pIMEIString
serialNumbersInfo, 833
pIMSDomain
GetIMSUserConfigParams, 271
imsUserConfigInfo, 323
SetIMSUserConfigReq, 846
pIMSDomainLen
GetIMSUserConfigParams, 271
SetIMSUserConfigReq, 846
pIMSTestMode
GetRegMgrConfigParams, 280
imsRegMgrConfigInfo, 321
SetRegMgrConfigReq, 856
pIOThresList
IOThresh, 328
nas_IOThresh, 453
pIOThresh
pack_nas_SLQSConfigSigInfo_t, 626
sigInfo, 865
pIPAddressV4
QmiWdsIpAddressInfo, 797
qmiWdsRunTimeSettings, 800
pIPAddressV6
QmiWdsIpAddressInfo, 797
pIPFamSupport
custFeaturesInfo, 205
pIPFamily
GetInstIDResp, 273
pIPFamilyPreference
qmiWdsRunTimeSettings, 800
pIPV6AddrInfo
qmiWdsRunTimeSettings, 800
pIPV6GWAddrInfo
qmiWdsRunTimeSettings, 800
pIPv4Addr
WdsDHCPv4ClientLeaseInd, 1384
pIPv4AddrInfo
QmiCbkLocGetServerInd, 778
unpack_loc_GetServer_Ind_t, 1080
pIPv4AddrPref

LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pIPv4Address
swiPDPRuntimeSettingsResp, 933
pIPv4DstAddr
swiQosFilter, 935
pIPv4GWAddress
swiPDPRuntimeSettingsResp, 933
pIPv4SrcAddr
swiQosFilter, 935
pIPv6AddPref
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pIPv6AddrInfo
QmiCbkLocGetServerInd, 778
unpack_loc_GetServer_Ind_t, 1080
pIPv6Address
swiPDPRuntimeSettingsResp, 933
pIPv6DstAddr
swiQosFilter, 935
pIPv6GWAddress
swiPDPRuntimeSettingsResp, 933
pIPv6Label
swiQosFilter, 935
pIPv6SrcAddr
swiQosFilter, 935
pIPv6TrafCls
swiQosFilter, 935
pIPv6prefixlen
QmiWdsIpAddressInfo, 797
pId
swiQosFilter, 935
pIds
swiQosIds, 939
pIgnoreHotSwapSwitch
pack_uim_SLQSUIMPowerUp_t, 681
UIMPowerUpReq, 999
pImCnFlag
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pImageList
pack_fms_SetImagesPreference_t, 606
unpack_fms_GetImagesPreference_t, 1070
pImeiSvnString
serialNumbersInfo, 833
pImgType
FirmwareUpdatStat, 246
pImsRegErrCode
IMSARegistrationStatus, 314
pImsRegStatus
IMSARegistrationStatus, 314
imsaRegStatusInfo, 315
pInUse
QmiNas3GppNetworkInfo, 794
pIndFieldsList
IMSASupportedFieldsResp, 318
pIndicationToken
pack_uim_ChangePin_t, 675
pack_uim_ReadTransparent_t, 676
pack_uim_SetPinProtection_t, 677
pack_uim_SLQSUIMAuthenticate_t, 678
pack_uim_SLQSUIMGetFileAttributes_t, 680
pack_uim_UnblockPin_t, 684
pack_uim_VerifyPin_t, 685
UIMAuthenticateReq, 991
UIMAuthenticateResp, 992
UIMChangePinReq, 992
UIMGetFileAttributesReq, 996
UIMGetFileAttributesResp, 997
UIMPinResp, 998
UIMReadTransparentReq, 1000
UIMReadTransparentResp, 1001
UIMSetPinProtectionReq, 1007
UIMUnblockPinReq, 1011
UIMVerifyPinReq, 1012
unpack_uim_ChangePin_t, 1193
unpack_uim_ReadTransparent_t, 1194
unpack_uim_SetPinProtection_t, 1195
unpack_uim_SLQSUIMAuthenticate_t, 1196
unpack_uim_SLQSUIMGetFileAttributes_t, 1199
unpack_uim_UnblockPin_t, 1201
unpack_uim_VerifyPin_t, 1201
pInfo
pack_voice_AnswerUSSD_t, 686
pack_voice_OriginateUSSD_t, 686
pInstanceID
GetInstIDResp, 273
pInstanceSize
QmiNasPerformNetworkScanResp, 794
pInstances
QmiNasPerformNetworkScanResp, 794
pInstancesSize
QmiNasGetRFBandInfoResp, 794
pIntermediateReportState
LOCStartReq, 388
pack_loc_Start_t, 621
pInterval
unpack_pds_GetPDSDefaults_t, 1144
unpack_pds_GetXTRAAutomaticDownload_t, 1146
pIpcpAckTimeout
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pIpcpCreqRetryCount
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pIsPcscfAddressNedded
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pIsUDHPresent

wcdmaLongMsgDecodingParams, 1357
pIsVoiceEnabled
custFeaturesInfo, 205
custFeaturesSetting, 207
pJitter
swiQosFlow, 938
pKeyReferenceID
pack_uim_ChangePin_t, 675
pack_uim_SetPinProtection_t, 677
pack_uim_UnblockPin_t, 684
pack_uim_VerifyPin_t, 685
UIMChangePinReq, 992
UIMSetPinProtectionReq, 1007
UIMUnblockPinReq, 1011
UIMVerifyPinReq, 1012
PLMN_LENGTH
qaGobiApiNas.h, 1723
PLMNData
nas_operatorPLMNList, 481
operatorPLMNList, 590
PLMNName
nas_operatorNameString, 479
operatorNameString, 588
PLMNNetName
nas_PLMNNetworkName, 487
PLMNNetworkName, 745
PLMNNetworkName, 744
numInstance, 745
PLMNNetName, 745
PLMNNetworkNameData, 745
codingScheme, 746
countryInitials, 746
longName, 746
longNameLen, 746
longNameSpareBits, 746
shortName, 746
shortNameLen, 746
shortNameSpareBits, 747
PLMNRecID
nas_OperatorPLMNData, 480
OperatorPLMNData, 589
pLTEAttachProfile
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 721
pLTEAttachProfileList
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 721
pLTEBandPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pLTECphyCa
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pLTEInfo
swiModemStatusResp, 929
unpack_nas_SLQSNasSwiModemStatus_t, 1131
pLTEInfoInterfreq
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pLTEInfoIntrafreq
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pLTEInfoNeighboringGSM
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pLTEInfoNeighboringWCDMA
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pLTERSRPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 631
setSignalStrengthInfo, 860
pLTERSRPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 860
pLTERSRPThreshList
LTERSRPThresh, 399
nas_LTERSRPThresh, 463
pLTERSRQDelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 860
pLTERSRQThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 860
pLTERSRQThreshList
LTERSRQThresh, 400
nas_LTERSRQThresh, 464
pLTERSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pLTERSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pLTERSSIThreshList
LTERSSIThresh, 400
nas_LTERSSIThresh, 464
pLTESNRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pLTESNRThresList
LTESNRThresh, 403
nas_LTESNRThresh, 467
pLTESNRThresh
pack_nas_SLQSConfigSigInfo_t, 626
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
sigInfo, 865
pLTESNRThreshList
LTESNRThreshold, 404
nas_LTESNRThreshold, 468
pLTESSInfo

nasGetSigInfoResp, 535
pLTESigInfo
nasSigInfo, 563
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1129
pLTESigRptCfg
pack_nas_SLQSConfigSigInfo_t, 626
sigInfo, 865
pLTESigRptConfig
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pLTESrvStatusInfo
nasGetSysInfoResp, 537
nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pLTESysInfo
nasGetSysInfoResp, 537
nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pLTEVoiceSupportSysInfo
nasGetSysInfoResp, 537
nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pLanguage
cdmaMsgDecodingParams, 166
pLastBearerTech
DataBearerTechExt, 212
pLastCallDataBearerTech
getDUNCallInfoResp, 266
pLastCallDataBearerTechnology
dataBearers, 209
pLastCallRXOKBytesCnt
getDUNCallInfoResp, 266
pLastCallTXOKBytesCnt
getDUNCallInfoResp, 266
pLatency
swiQosFlow, 938
pLatitude
LocInjectPositionReq, 384
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 734
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pLcpAckTimeout
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pLcpCreqRetryCount
LibpackProfile3GPP2, 343
LibPackprofile_3GPP2, 354
Profile3GPP2, 761
pLeapSeconds
QmiCbkLocPositionReportInd, 787
unpack_loc_PositionRpt_Ind_t, 1087
pLeaseState
WdsDHCPv4ClientLeaseInd, 1384
pLineCtrlInfo
voiceInfoRec, 1334
pLinktimer
pack_sms_SendSMS_t, 662
slqssendasyncsmsparams_s, 877
slqssendsmsparams_s, 878
sms_sendAsyncsmsParams, 897
pLogString
FirmwareUpdatStat, 246
pLogStringLen
FirmwareUpdatStat, 246
pLongitude
LocInjectPositionReq, 384
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 734
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pLoopbackMode
WDSSetLoopbackData, 1397
pLoopbackMultiplier
WDSSetLoopbackData, 1397
pLteBandCapability
BandCapabilityResp, 140
pLteEARFCN
nasSwiGetChannelLockResp, 564
nasSwiSetChannelLockReq, 566
pack_nas_SLQSNASSwiSetChannelLock_t, 638
unpack_nas_SLQSNASSwiGetChannelLock_t, 1130
pLteNasRelInfo
SwiOTAMsg_s, 930
pLtePCI
nasSwiGetChannelLockResp, 564
nasSwiSetChannelLockReq, 566
pack_nas_SLQSNASSwiSetChannelLock_t, 638
unpack_nas_SLQSNASSwiGetChannelLock_t, 1130
pLteQci
swiQosFlow, 938
pMCC
QmiNas3GppNetworkInfo, 794
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pMEIDString
serialNumbersInfo, 833
pMICGainSelect
GetAudioPathConfigResp, 257
pMIPStatusInd
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1397
pMNAAA
pack_wds_SetMobileIPProfile_t, 714
pMNC
QmiNas3GppNetworkInfo, 794

unpack_nas_GetHomeNetwork3GPP2_t, 1095
pMNCIncPCSDigStat
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
pMNHA
pack_wds_SetMobileIPProfile_t, 714
pMNRInfo
nasInitNetworkReg, 543
pack_nas_SLQSInitiateNetworkRegistration_t, 628
pMTMessageInfo
SMSEventInfo_s, 907
pMagneticDeviation
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pManString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pManagedRoamingInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pMaxAllowedPktSz
swiQosFlow, 938
pMaxChannelRXRate
WdsConnectionRateElmnts, 1384
pMaxChannelTXRate
WdsConnectionRateElmnts, 1384
pMaxStorageSizeReq
pack_sms_SLQSSmsGetMaxStorageSize_t, 669
pMaxStorageSizeResp
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1184
pMdmCallDurationActive
getDUNCallInfoResp, 266
pMeidString
_SLQSSwiGetSerialNoExtParams, 97
pMessage
cdmaMsgDecodingParams, 167
cdmaMsgEncodingParams, 168
pack_sms_SaveSMS_t, 661
pack_sms_SendSMS_t, 662
slqssendasyncsmsparams_s, 877
slqssendsmsparams_s, 878
sms_sendAsyncsmsParams, 897
wcdmaLongMsgDecodingParams, 1357
wcdmaMsgDecodingParams, 1359
pMessageID
cdmaMsgDecodingParams, 167
pMessageIndex
pack_sms_SLQSDeleteSMS_t, 663
unpack_sms_SaveSMS_t, 1178
pMessageMode
pack_sms_SLQSDeleteSMS_t, 663
pack_sms_SLQSGetSMS_t, 664
pack_sms_SLQSGetSMSList_t, 665
pack_sms_SLQSModifySMSStatus_t, 666
sms_maxStorageSizeReq, 891
smsMaxStorageSizeReq, 907
pMessageModeInfo
SMSEventInfo_s, 906
pMessageProtocol
unpack_sms_SLQSSmsGetMessageProtocol_t, 1184
pMessageSize
cdmaMsgEncodingParams, 168
pMessageTag
pack_sms_SLQSDeleteSMS_t, 663
pMicMute
SetM2MAudioProfileReq, 852
pMinBasedIMSI
nasGet3GPP2SubscriptionInfoResp, 532
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1124
pMinIntervalTime
LOCStartReq, 388
pack_loc_Start_t, 621
pMinPolicedPktSz
swiQosFlow, 939
pMinSessionExpiryTimer
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 325
SetIMSVoIPConfigReq, 849
pMitigationDevList
TmdGetMitigationDevListResp, 956
pMitigationDevListLen
TmdGetMitigationDevListResp, 956
pMncPcsDigitStatus
nasInitNetworkReg, 543
pack_nas_SLQSInitiateNetworkRegistration_t, 628
pMncPcsStatus
nasPLMNNameReq, 555
pack_nas_SLQSGetPLMNName_t, 627
pMode
pack_wds_SetMobileIPParameters_t, 712
pModePref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pModelString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pMtu
qmiWdsRunTimeSettings, 800
pNAI
pack_wds_SetMobileIPProfile_t, 714
pNAMNameInfo
nasGet3GPP2SubscriptionInfoResp, 532
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1124
pNID
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNITZInformation
nasOperatorNameResp, 550
unpack_nas_SLQSGetOperatorNameData_t, 1109

pNSEnable
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pNSSAudioCtrl
voiceInfoRec, 1334
pNSSRelease
voiceInfoRec, 1334
pNamID
pack_voice_SLQSVoiceGetConfig_t, 695
voiceGetConfigReq, 1329
pName
pack_wds_SetDefaultProfile_t, 710
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNameString
_SLQSSwiGetOSInfoParams, 96
_SLQSSwiSetOSInfoParams, 99
pNegoDnsSrvrPref
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pNeighborSetPilotPN
nas_NetworkStat1x, 476
NetworkStat1x, 574
pNetSelPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pNetworkInfo
_slqsNetworkScanInfo, 88
pNetworkInfoInstances
_slqsNetworkScanInfo, 88
pNetworkInfoLen
CurrDataSysStat, 197
pNetworkMode
pack_pds_SLQSGetAGPSConfig_t, 652
pack_pds_SLQSSetAGPSConfig_t, 655
pNetworkRejectInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pNetworkStat1x
NetworkDebugResp, 572
unpack_nas_SLQSSwiNetworkDebug_t, 1136
pNetworkStatEVDO
NetworkDebugResp, 573
unpack_nas_SLQSSwiNetworkDebug_t, 1136
pNetworkTimeInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pNewImsRegStatus
IMSARegistrationStatus, 314
pNewPwdData
voiceSUPSInfo, 1352
pNotifStore
_SLQSAVMSSetSettingsReqPara, 85
pNotificationStore
_SLQSAVMSSettings, 85
pNumSupUSBComps
USBCompParams, 1262
pNumberOfPhySlot
UIMGetSlotsStatusResp, 997
unpack_uim_SLQSUIMGetSlotsStatus_t, 1199
pNw2DescDisp
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNw2DescEnc
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNw2DescLen
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNw2MCC
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNw2MNC
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNw2Name
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pNxtHdrProto
swiQosFilter, 935
pOMADMEnabled
_SLQSAVMSSettings, 86
_SLQSOMADMSettings, 93
PORTNAM_LEN
qaGobiApiDcs.h, 1637
pOTASPStatus
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1207
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
voiceGetAllCallInfo, 1316
pObjectVer
NetworkDebugResp, 573
unpack_nas_SLQSSwiNetworkDebug_t, 1136
pOffLength
voiceDTMFEventInfo, 1312
pOnLength
voiceDTMFEventInfo, 1312
pOpaqueIdentifier
LocInjectSensorDataReq, 385
QmiCbkLocInjectSensorDataInd, 780
pOperation
unpack_pds_GetPDSDefaults_t, 1144
pOperatorNameString
nasOperatorNameResp, 550
unpack_nas_SLQSGetOperatorNameData_t, 1109
pOperatorPLMNList
nasOperatorNameResp, 550
unpack_nas_SLQSGetOperatorNameData_t, 1109
pOptList
wds_DHCPv4OptionList, 1371
WdsDHCPv4ClientLeaseInd, 1384
WdsDHCPv4OptionList, 1388
wdsDhcpv4OptionList, 1388
pOptVal
DHCPOption, 222
wds_DHCPOpt, 1369
pOptions
DHCPOptionList, 223
pPCMParams
SetM2MAudioAVCFGReq, 850

pPCSCFAddrPCO
qmiWdsRunTimeSettings, 800
pPCSCFFQDNAddrList
qmiWdsRunTimeSettings, 800
pPCSCFPort
GetRegMgrConfigParams, 280
pPCSDigitInfo
_slqsNetworkScanInfo, 88
pPCSDigitInstances
_slqsNetworkScanInfo, 88
pPCSInstance
unpack_nas_PerformNetworkScan_t, 1103
pPCSInstanceSize
unpack_nas_PerformNetworkScan_t, 1103
pPDNInactivTimeout
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPDNInactivTimeout3GPP2
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pPDPType
qmiWdsRunTimeSettings, 800
pPDPtype
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPDUMessage
wcdmaMsgEncodingParams, 1360
pPLMNNetworkName
nasOperatorNameResp, 550
unpack_nas_SLQSGetOperatorNameData_t, 1109
pPRLPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pPRLPreference
dmsCurrentPRLInfo, 226
pPRLVersion
dmsCurrentPRLInfo, 226
pPWD
_SLQSAVMSSettingsAPNInfo, 86
pPacketsCountRX
QosEventInfo, 802
slqsWdsEventInfo, 886
pPacketsCountTX
QosEventInfo, 802
slqsWdsEventInfo, 886
pPartNum
wcdmaLongMsgDecodingParams, 1358
pPass
pack_wds_SLQSStartDataSession_t, 725
pPassword
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
pack_wds_SetDefaultProfile_t, 710
Profile3GPP, 755
ssdatasession_params, 920
pPasswordSize
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPcscfAddrUsingDhcp
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPcscfAddrUsingPCO
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPdnType
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pPdpAccessConFlag
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPdpContext
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPdpDataCompType
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPdpHdrCompType
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPdpStatusConfig
IMSAIndRegisterInfo, 312
pPeroidInfo
_SLQSAVMSSetSettingsReqPara, 85
pPeroidsInfo
_SLQSAVMSSettings, 86
pPersonalityListLength
HDRPersonalityInd, 299
HDRPersonalityResp, 299
HDRProtSubtypResp, 300
unpack_nas_SLQSSwiGetHDRPersonality_t, 1133
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1133
pPersonalizationStatus
UIMGetConfigurationResp, 996
unpack_uim_SLQSUIMGetConfiguration_t, 1198
pPhoneCtxtURI
GetIMSSMSConfigParams, 270
imsSMSConfigInfo, 323
SetIMSSMSConfigReq, 845
pPhoneCtxtURILen
GetIMSSMSConfigParams, 270
SetIMSSMSConfigReq, 845
pPhyCaAggPcellInfo

nasGetLTECphyCaResp, 534
pPhyCaAggScellDlBw
nasGetLTECphyCaResp, 534
pPhyCaAggScellIndType
nasGetLTECphyCaResp, 534
pPhyCaAggScellIndex
nasGetLTECphyCaResp, 534
pPhyCaAggScellInfo
nasGetLTECphyCaResp, 534
pPilotSetData
GetHRPDStatsResp, 269
unpack_nas_SLQSSwiGetHRPDStats_t, 1134
pPilotSetInfo
nas_PilotSetData, 486
PilotSetData, 742
pPkgDescLength
_SLQSOMADMSessionInfo, 91
pPkgDescription
_SLQSOMADMSessionInfo, 91
pPkgName
_SLQSOMADMSessionInfo, 91
pPkgNameLength
_SLQSOMADMSessionInfo, 91
pPktErrRate
swiQosFlow, 939
pPlasmaIDString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pPlmnId
nasNetworkReject, 548
pPollingTimer
_SLQSAVMSSetSettingsReqPara, 85
_SLQSAVMSSettings, 86
pPositionSource
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 734
pPositionSrc
LocInjectPositionReq, 384
pPppSessCloseTimer1x
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pPppSessCloseTimerDO
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pPrDNSIPv4Address
swiPDPRuntimeSettingsResp, 933
pPrDNSIPv6Address
swiPDPRuntimeSettingsResp, 933
pPrPCSCFIPv4Address
swiPDPRuntimeSettingsResp, 933
pPrPCSCFIPv6Address
swiPDPRuntimeSettingsResp, 933
pPrecedence
swiQosFilter, 935
pPrecisionDilution
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pPrefNetwork
CurrDataSysStat, 197
pPrefVoiceDomain
pack_voice_SLQSVoiceSetConfig_t, 701
voiceSetConfigReq, 1344
pPrefVoicePrivacy
pack_voice_SLQSVoiceGetConfig_t, 695
voiceGetConfigReq, 1329
pPrefVoiceSO
pack_voice_SLQSVoiceGetConfig_t, 695
pack_voice_SLQSVoiceSetConfig_t, 701
voiceGetConfigReq, 1329
voiceSetConfigReq, 1344
pPrefVoiceSOStatus
unpack_voice_SLQSVoiceSetConfig_t, 1225
voiceSetConfigResp, 1345
pPreference
unpack_pds_GetXTRANetwork_t, 1146
pPreferred
QmiNas3GppNetworkInfo, 794
pPriCSCFPort
imsRegMgrConfigInfo, 321
SetRegMgrConfigReq, 856
pPriCSCFPortName
GetRegMgrConfigParams, 280
pPriCSCFPortNameLen
GetRegMgrConfigParams, 280
pPriDNSIPv4AddPref
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPriDNSIPv6addpref
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPriV6DnsAddress
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pPrimaryDNSV4
qmiWdsRunTimeSettings, 800
pPrimaryDNSV6
qmiWdsRunTimeSettings, 800
pPrimaryHA
pack_wds_SetMobileIPProfile_t, 714
pPrimaryID
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 755
pPrimaryV4DnsAddress
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pPriority
cdmaMsgDecodingParams, 167

cdmaMsgEncodingParams, 168
pPrivacy
cdmaMsgDecodingParams, 167
pProfSz
NWProfile, 580
pProfValues
NWProfile, 580
pProfileID
CreateProfileIn, 192
ModifyProfileIn, 413
qmiWdsRunTimeSettings, 800
unpack_wds_SLQSCreateProfile_t, 1241
pProfileId
pack_wds_SLQSCreateProfile_t, 715
pack_wds_SLQSModifyProfile_t, 720
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 723
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 724
WdsDHCPv4ClientLeaseInd, 1385
WdsDHCPv4Config, 1385
pProfileId3GPP
ssdatasession_params, 920
pProfileId3GPP2
ssdatasession_params, 920
swiQosFlow, 939
pProfileIndex
CreateProfileOut, 193
pProfileList
_slqs3GPPConfigItem, 83
pack_wds_SLQSSet3GPPConfigItem_t, 721
pProfileName
qmiWdsRunTimeSettings, 800
pProfileSettings
unpack_wds_SLQSGetProfileSettings_t, 1248
pProfileType
CreateProfileIn, 192
CreateProfileOut, 193
ModifyProfileIn, 413
pack_wds_SLQSCreateProfile_t, 715
pack_wds_SLQSModifyProfile_t, 720
pProfilename
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pProfilenameSize
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pProtoSubTypElmnt
HDRProtSubtypResp, 300
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1133
pProtocol
pack_nas_SetCDMANetworkParameters_t, 624
pProtocolSubtypeElement
HDRPersonalityInd, 299
HDRPersonalityResp, 299
unpack_nas_SLQSSwiGetHDRPersonality_t, 1133
pQFlowState
QosFlowInfo, 803
pQmiInterfaceInfo
_packetSrvStatus, 70
slqsSessionStateInfo, 879
slqsWdsEventInfo, 886
pQosClassID
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pRAT
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
pRATINstance
unpack_nas_PerformNetworkScan_t, 1103
pRATInfo
_slqsNetworkScanInfo, 88
pRATInstanceSize
unpack_nas_PerformNetworkScan_t, 1103
pRATInstances
_slqsNetworkScanInfo, 88
pRATStatus
imsaRatStatusInfo, 313
pRATType
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
PREFERRED_INDEX
qaNasPerformNetworkScan.h, 1964
pRFBandInfoElements
QmiNasGetRFBandInfoResp, 794
PRI_UPDATE_FAIL
qaGobiApiFms.h, 1686
PRIString
unpack_dms_GetFirmwareRevision_t, 1036
unpack_dms_GetFirmwareRevisions_t, 1037
PRLInd
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1114
PRLPTlv
NASQmiCbkNasSystemSelPrefInd, 560
PRLPref
NASPRLPreferenceTlv, 558
pRMAutoConnect
custFeaturesInfo, 205
pRSRPThresList
nas_RSRPThresh, 497
RSRPThresh, 816
pRSRPThresh
pack_nas_SLQSConfigSigInfo_t, 626
sigInfo, 865
pRSRQThresList
nas_RSRQThresh, 498
RSRQThresh, 817
pRSRQThresh
pack_nas_SLQSConfigSigInfo_t, 626
sigInfo, 865
pRSSIThresList
nas_RSSIThresh, 498

RSSIThresh, 818
pRSSIThresh
pack_nas_SLQSConfigSigInfo_t, 626
sigInfo, 865
pRTPRTCPInactTimer
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 326
SetIMSVoIPConfigReq, 849
pRXAGCList
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pRXAIG
RXAGCList, 819
pRXAVCAGCSwitch
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pRXAVCList
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pRXChain0Info
nasGetTxRxInfoResp, 539
unpack_nas_SLQSNasGetTxRxInfo_t, 1127
pRXChain1Info
nasGetTxRxInfoResp, 539
unpack_nas_SLQSNasGetTxRxInfo_t, 1127
pRXComprSlope
RXAGCList, 819
pRXComprThres
RXAGCList, 819
pRXDroppedCount
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pRXExpSlope
RXAGCList, 819
pRXExpThres
RXAGCList, 819
pRXOKBytesCount
getDUNCallInfoResp, 266
pRXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pRXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pRXPCMIIRFltr
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pRXPacketErrors
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pRXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pRXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pRXStaticGain
RXAGCList, 819
pRXTotalBytes
unpack_wds_GetByteTotals_t, 1229
WdsByteTotalsElmnts, 1382
pRadioInterface
nasNetworkTime, 549
unpack_nas_SLQSNasNetworkTimeCallBack_ind-
_t, 1128
pRankIndicatorInd
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
pRatHandoverStatusConfig
IMSAIndRegisterInfo, 312
pRawHorConfidence
LocInjectPositionReq, 384
pRawHorUncCircular
LocInjectPositionReq, 384
pReRegPeriod
pack_wds_SetMobileIPParameters_t, 712
pReRegTraffic
pack_wds_SetMobileIPParameters_t, 712
pReadAcknowledgementReq
cdmaMsgDecodingParams, 167
pReadResult
UIMReadTransparentResp, 1001
unpack_uim_ReadTransparent_t, 1194
pReason
voiceSUPSInfo, 1352
pReasonMask
unpack_dms_GetOfflineReason_t, 1041
pRecurrenceType
LOCStartReq, 388
pack_loc_Start_t, 621
pRedirNumInfo
voiceInfoRec, 1334
pRefData
FirmwareUpdatStat, 246
pRefString
FirmwareUpdatStat, 246
pRefStringLen
FirmwareUpdatStat, 246
pReferenceNum
wcdmaLongMsgDecodingParams, 1358
pRefreshEvent
UIMRefreshGetLastEventResp, 1004
unpack_uim_SLQSUIMRefreshGetLastEvent_t,
1200
pRegCallStatInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 889
sms_setIndicationReg, 898
pRegDTMFEvents
pack_voice_SLQSVoiceIndicationRegister_t, 696
voiceIndicationRegisterInfo, 1332
pRegInd
_getTransLayerInfoResp, 66
sms_getTransLayerInfo, 890
pRegMgrConfigEvents

imsCfgIndRegisterInfo, 320
pRegStatus
_getTransNWRegInfoResp, 67
sms_getTransNWRegInfo, 890
pRegStatusConfig
IMSAIndRegisterInfo, 312
pRegStatusErrorCode
imsaRegStatusInfo, 315
pRegTransLayerInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 889
sms_setIndicationReg, 898
pRegTransNWRegInfoEvt
_getIndicationRegResp, 64
_setIndicationRegReq, 81
sms_getIndicationReg, 889
sms_setIndicationReg, 898
pRegVoicePrivacyEvents
pack_voice_SLQSVoiceIndicationRegister_t, 696
voiceIndicationRegisterInfo, 1332
pRejectReason
pack_swioma_SLQSOMADMSendSelection_t, 672
pRelValidity
cdmaMsgEncodingParams, 168
pRelativeValidity
cdmaMsgDecodingParams, 167
pRemainingRetries
UIMDepersonalizationResp, 993
UIMPinResp, 998
unpack_uim_ChangePin_t, 1193
unpack_uim_SetPinProtection_t, 1195
unpack_uim_SLQSUIMDepersonalization_t, 1197
unpack_uim_UnblockPin_t, 1201
unpack_uim_VerifyPin_t, 1201
pRemotePartyName
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pRemotePartyNum
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pReportChannelRate
getDUNCallInfoReq, 263
pack_wds_SLQSGetDUNCallInfo_t, 717
pReportConnStatus
getDUNCallInfoReq, 263
pack_wds_SLQSGetDUNCallInfo_t, 717
pReportDataBearerTech
getDUNCallInfoReq, 263
pack_wds_SLQSGetDUNCallInfo_t, 717
pReportDormStatus
getDUNCallInfoReq, 263
pack_wds_SLQSGetDUNCallInfo_t, 717
pReqFieldsList
IMSASupportedFieldsResp, 318
pReqMitigationLvl
TmdGetMitigationLvlResp, 957
pReqSettings
pack_wds_SLQSGetRuntimeSettings_t, 719
pRequestOptionList
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 724
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1254
WdsDHCPv4Config, 1385
pRequestedTag
pack_sms_SLQSGetSMSList_t, 665
pRespData
USSDRespFNetwork, 1263
pRespFieldsList
IMSASupportedFieldsResp, 318
pRetryCount
_SLQSOMADMSessionInfo, 91
pRetryInterval
pack_wds_SetMobileIPParameters_t, 712
pRetryLimit
pack_wds_SetMobileIPParameters_t, 712
pRetryMessage
slqssendasyncsmsparams_s, 877
sms_sendAsyncsmsParams, 897
pRetryMessageId
slqssendasyncsmsparams_s, 877
sms_sendAsyncsmsParams, 897
pRevInUse
CDMASysInfo, 174
nas_CDMASysInfo, 425
pRevInUseValid
CDMASysInfo, 174
nas_CDMASysInfo, 425
pRevTunneling
pack_wds_SetMobileIPProfile_t, 714
pRingBackTimer
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 325
SetIMSVoIPConfigReq, 849
pRingingTimer
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 326
SetIMSVoIPConfigReq, 849
pRoamPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 110
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pRoamTimer
pack_voice_SLQSVoiceGetConfig_t, 695
voiceGetConfigReq, 1329
pRoamTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 1222
voiceGetConfigResp, 1331
pRoamTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 701
voiceSetConfigReq, 1344
pRoamTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 1225
voiceSetConfigResp, 1345

pRoaming
pack_nas_SetCDMANetworkParameters_t, 624
QmiNas3GppNetworkInfo, 794
pRscp
nasSigInfo, 563
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1129
pRxFilter
swiQosModifyReq, 940
swiQosReq, 941
pRxFlow
swiQosGranted, 939
swiQosModifyReq, 940
swiQosReq, 941
pRxQFilter
QosFlowInfo, 803
pRxQFlowGranted
QosFlowInfo, 803
PSDetachReq, 763
pDetachAction, 764
PSDomain
unpack_nas_GetServingNetwork_t, 1100
pSID
unpack_nas_GetHomeNetwork3GPP2_t, 1095
pSIPConfigEvents
imsCfgIndRegisterInfo, 320
pSIPLocalPort
GetSIPConfigResp, 281
imsSIPConfigInfo, 322
SetSIPConfigReq, 862
pSMSAttemptedFlag
getDyingGaspStatistics, 267
packgetDyingGaspStatistics, 730
pSMSCAddress
pack_sms_SetSMSCAddress_t, 662
unpack_sms_GetSMSCAddress_t, 1177
pSMSCAddressInfo
SMSEventInfo_s, 907
pSMSCType
pack_sms_SetSMSCAddress_t, 662
unpack_sms_GetSMSCAddress_t, 1178
pSMSConfigEvents
imsCfgIndRegisterInfo, 320
pSMSFormat
GetIMSSMSConfigParams, 270
imsSMSConfigInfo, 323
SetIMSSMSConfigReq, 845
pSMSOnIMSInfo
SMSEventInfo_s, 907
pSMSOverIPNwInd
GetIMSSMSConfigParams, 270
imsSMSConfigInfo, 323
SetIMSSMSConfigReq, 845
pSMSSupport
custFeaturesInfo, 205
pSMSSvcRAT
imsaSvcStatusInfo, 319
pSMSSvcStatus
imsaSvcStatusInfo, 319
pSPC
pack_nas_SetCDMANetworkParameters_t, 624
pack_wds_SetMobileIPParameters_t, 712
pSV
BdsSVInfo, 141
loc_BdsSVInfo, 363
loc_SVInfo, 373
SVInfo, 924
pSVInfo
LocDelAssDataReq, 377
pack_loc_Delete_Assist_Data_t, 607
pSWVerString
_SLQSSwiGetHostDevInfoParams, 96
_SLQSSwiSetHostDevInfoParams, 98
pSarRFState
unpack_sar_SLQSGetRfSarState_t, 1177
pSatelliteInfo
gnssSvInfoNotification, 284
unpack_loc_GnssSvInfo_Ind_t, 1081
pScAddr
wcdmaLongMsgDecodingParams, 1358
wcdmaMsgDecodingParams, 1359
pScAddrLength
wcdmaLongMsgDecodingParams, 1358
wcdmaMsgDecodingParams, 1359
pScanResult
_slqsNetworkScanInfo, 88
unpack_nas_PerformNetworkScan_t, 1103
pSccRxInfo
LteSccRxInfoResp, 400
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1135
pScrAmrEnable
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 326
SetIMSVoIPConfigReq, 849
pScrAmrWbEnable
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 326
SetIMSVoIPConfigReq, 849
pSeDNSIPv4Address
swiPDPRuntimeSettingsResp, 933
pSeDNSIPv6Address
swiPDPRuntimeSettingsResp, 933
pSePCSCFIPv4Address
swiPDPRuntimeSettingsResp, 933
pSePCSCFIPv6Address
swiPDPRuntimeSettingsResp, 933
pSecDNSIPv4AddPref
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pSecDNSIPv6addpref
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pSecV6DnsAddress
LibpackProfile3GPP2, 344

LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pSecondaryDNSV4
qmiWdsRunTimeSettings, 800
pSecondaryDNSV6
qmiWdsRunTimeSettings, 800
pSecondaryFlag
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pSecondaryHA
pack_wds_SetMobileIPProfile_t, 714
pSecondaryV4DnsAddress
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pSectorID
nas_NetworkStatEVDO, 478
NetworkStatEVDO, 576
pSendSmsParams
pack_sms_SLQSSendAsyncSMS_t, 666
pSenderAddr
cdmaMsgDecodingParams, 167
wcdmaLongMsgDecodingParams, 1358
wcdmaMsgDecodingParams, 1359
pSenderAddrLength
cdmaMsgDecodingParams, 167
wcdmaLongMsgDecodingParams, 1358
wcdmaMsgDecodingParams, 1359
pSensorDataUsage
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pServerAddrList
qmiWdsRunTimeSettings, 800
pServerAddrTypeMask
LocGetServerReq, 380
pServerAddress
pack_pds_SLQSSetAGPSConfig_t, 655
unpack_pds_SLQSGetAGPSConfig_t, 1150
pServerPort
pack_pds_SLQSSetAGPSConfig_t, 655
unpack_pds_SLQSGetAGPSConfig_t, 1150
pServerURL
pack_pds_SLQSSetAGPSConfig_t, 655
unpack_pds_SLQSGetAGPSConfig_t, 1150
pServerURLLength
pack_pds_SLQSSetAGPSConfig_t, 655
unpack_pds_SLQSGetAGPSConfig_t, 1150
pServiceClass
pack_voice_SLQSVoiceSetSUPSService_t, 703
voiceSetSUPSServiceReq, 1348
pServiceOption
slqssendasyncsmsparams_s, 877
sms_sendAsyncsmsParams, 897
pServiceStatusConfig
IMSAIndRegisterInfo, 312
pServingSystemInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pSessionExpiryTimer
GetIMSVoIPConfigResp, 273
imsVoIPConfigInfo, 326
SetIMSVoIPConfigReq, 849
pSessionIDv4
GetSessionIDResp, 280
pSessionIDv6
GetSessionIDResp, 280
pSessionState
_SLQSOMADMSessionInfo, 91
pSessionType
_SLQSOMADMSessionInfo, 91
pSetIndicationRegReq
pack_sms_SLQSSetIndicationRegister_t, 667
pSetRoutesReq
pack_sms_SLQSSmsSetRoutes_t, 669
pSettingResp
GetIMSSMSConfigParams, 270
GetIMSUserConfigParams, 271
GetIMSVoIPConfigResp, 273
GetRegMgrConfigParams, 280
GetSIPConfigResp, 281
SetIMSSMSConfigResp, 846
SetIMSUserConfigResp, 847
SetIMSVoIPConfigResp, 850
SetRegMgrConfigResp, 856
SetSIPConfigResp, 863
pSeverity
_SLQSOMADMSessionInfo, 91
pSigCompEnabled
GetSIPConfigResp, 281
imsSIPConfigInfo, 322
SetSIPConfigReq, 862
pSigIndReq
pack_nas_SLQSSetSignalStrengthsCallback_t, 640
pSignalInfo
voiceInfoRec, 1334
pSignalStrengthInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pSmsOnIms
slqssendasyncsmsparams_s, 877
slqssendsmsparams_s, 878
sms_sendAsyncsmsParams, 897
pSmsServiceRat
IMSAServiceStatus, 317
pSmsServiceStatus
IMSAServiceStatus, 317
pSmsStorage
unpack_sms_SLQSSwiGetSMSStorage_t, 1185
pSource
_SLQSOMADMSessionInfo, 91
pSourceIP
LibPackTFTIDParams, 358

TFTIDParams, 953
pSourceLength
_SLQSOMADMSessionInfo, 91
pSpeedHorizontal
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pSpeedUnc
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pSpeedVertical
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pSpeedVerticalUnc
QmiCbkLocBestAvailPosInd, 775
unpack_loc_BestAvailPos_Ind_t, 1076
pSrcRAT
imsaRatStatusInfo, 313
pSrvDomainPref
_sysSelectPrefInfo, 104
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 645
unpack_nas_SLQSGetSysSelectionPref_t, 1123
pSrvOpt
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pSrvRegRestriction
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 645
pSrvcProviderName
nasOperatorNameResp, 550
unpack_nas_SLQSGetOperatorNameData_t, 1109
pStage0Val
RXPCMIIRFltr, 822
TXPCMIIRFltr, 965
pStage1Val
RXPCMIIRFltr, 822
TXPCMIIRFltr, 965
pStage2Val
RXPCMIIRFltr, 822
TXPCMIIRFltr, 965
pStage3Val
RXPCMIIRFltr, 822
TXPCMIIRFltr, 965
pStage4Val
RXPCMIIRFltr, 822
TXPCMIIRFltr, 965
pStageCnt
RXPCMIIRFltr, 822
TXPCMIIRFltr, 965
pStatMask
pack_wds_GetPacketStatistics_t, 707
WdsPktStatisticsReq, 1393
pStatus
_SLQSOMADMSessionInfo, 91
pSubnetMaskV4
qmiWdsRunTimeSettings, 800
pSubscribeTimer
GetSIPConfigResp, 281
imsSIPConfigInfo, 322
SetSIPConfigReq, 862
pSubscriptionInfoInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pSupUSBComps
USBCompParams, 1262
pSupportedMsgList
IMSASupportedMsgInfo, 318
pSuppressSysInfoInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pSuppsNotifEvents
pack_voice_SLQSVoiceIndicationRegister_t, 696
voiceIndicationRegisterInfo, 1332
pSvUsedforFix
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pSvcClass
pack_voice_SLQSVoiceGetCallBarring_t, 692
pack_voice_SLQSVoiceGetCallForwardingStatus-
_t, 693
pack_voice_SLQSVoiceGetCallWaiting_t, 694
unpack_voice_SLQSVoiceGetCallBarring_t, 1208
unpack_voice_SLQSVoiceGetCallWaiting_t, 1213
voiceGetCallBarringReq, 1316
voiceGetCallBarringResp, 1318
voiceGetCallFWReq, 1318
voiceGetCallWaitInfo, 1321
voiceSUPSInfo, 1352
pSvcType
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
pSwiGetResetInd
dmsIndicationRegisterReq, 229
pSysInfoInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pSysInfoNoChange
nasSysInfo, 569
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pSystemSelectionInd
nasIndicationRegisterReq, 542
pack_nas_SLQSNasIndicationRegisterExt_t, 636
pTCPDstPort
swiQosFilter, 935
pTCPSrcPort
swiQosFilter, 935
pTDSCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 632

setSignalStrengthInfo, 861
pTDSCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pTDSCDMAECIOThreshList
nas_TDSCDMAECIOThresh, 512
TDSCDMAECIOThresh, 945
pTDSCDMARSCPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pTDSCDMARSCPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pTDSCDMARSCPThreshList
nas_TDSCDMARSCPThresh, 512
TDSCDMARSCPThresh, 946
pTDSCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pTDSCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pTDSCDMARSSIThreshList
nas_TDSCDMARSSIThresh, 513
TDSCDMARSSIThresh, 946
pTDSCDMASINRCONFThresh
sigInfo, 865
pTDSCDMASINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pTDSCDMASINRThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pTDSCDMASINRThreshList
nas_TDSCDMASINRThresh, 514
TDSCDMASINRThresh, 949
pTDSCDMASigInfoExt
nasGetSigInfoResp, 535
nasSigInfo, 563
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1129
pTDSCDMASigInfoRscp
nasGetSigInfoResp, 535
pTFTID1Params
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pTFTID2Params
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pTTYConfigStatus
unpack_voice_SLQSVoiceSetConfig_t, 1225
voiceSetConfigResp, 1345
pTTYMode
pack_voice_SLQSVoiceGetConfig_t, 695
pack_voice_SLQSVoiceSetConfig_t, 701
voiceGetConfigReq, 1329
voiceSetConfigReq, 1344
pTXAGCList
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pTXAIG
TXAGCList, 963
pTXAVCSwitch
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pTXComprSlope
TXAGCList, 963
pTXComprThres
TXAGCList, 963
pTXDroppedCount
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pTXExpSlope
TXAGCList, 963
pTXExpThres
TXAGCList, 963
pTXGain
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pTXInfo
nasGetTxRxInfoResp, 539
unpack_nas_SLQSNasGetTxRxInfo_t, 1127
pTXOKBytesCount
getDUNCallInfoResp, 266
pTXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pTXOkBytesCount
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pTXPCMIIRFltr
GetAudioPathConfigResp, 257
SetAudioPathConfigReq, 841
pTXPacketErrors
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pTXPacketOverflows
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pTXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 1237
WdsPktStatisticsElmnts, 1392
pTXStaticGain
TXAGCList, 963
pTXTotalBytes
unpack_wds_GetByteTotals_t, 1229
WdsByteTotalsElmnts, 1382
pTdsBandCapability
BandCapabilityResp, 140
pTdscdmaBandPref
_sysSelectPrefParams, 111
pack_nas_SLQSSetSysSelectionPref_t, 646
pTech
pack_wds_SLQSStartDataSession_t, 725

pTechnology
qmiWdsRunTimeSettings, 800
ssdatasession_params, 920
pTechnologyMask
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pTextMsg
cdmaMsgDecodingParams, 167
cdmaMsgEncodingParams, 169
wcdmaLongMsgDecodingParams, 1358
wcdmaMsgDecodingParams, 1359
wcdmaMsgEncodingParams, 1360
pTextMsgLength
cdmaMsgDecodingParams, 167
wcdmaLongMsgDecodingParams, 1358
wcdmaMsgDecodingParams, 1359
pTgtRAT
imsaRatStatusInfo, 313
pTime
_SLQSOMADMSessionInfo, 91
SwiOTAMsg_s, 930
pTimeLength
_SLQSOMADMSessionInfo, 91
pTimeSrc
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pTimeStamp
getDyingGaspStatistics, 268
pack_pds_SLQSPDSInjectPositionData_t, 654
packgetDyingGaspStatistics, 730
PDSPositionData, 734
pTimeType
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 734
pTimeUnc
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pTimeZone
nasNetworkTime, 549
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1128
pTimeout
unpack_pds_GetPDSDefaults_t, 1144
pTimer
NasSwiIndReg, 565
pack_nas_SLQSNasSwiIndicationRegister_t, 637
pTimerSIPReg
GetSIPConfigResp, 281
imsSIPConfigInfo, 322
SetSIPConfigReq, 862
pTimerT1
GetSIPConfigResp, 281
imsSIPConfigInfo, 322
SetSIPConfigReq, 862
pTimerT2
GetSIPConfigResp, 281
imsSIPConfigInfo, 322
SetSIPConfigReq, 862
pTimerTf
GetSIPConfigResp, 281
imsSIPConfigInfo, 322
SetSIPConfigReq, 862
pTimerVal
pack_voice_SLQSVoiceSetSUPSService_t, 703
voiceSetSUPSServiceReq, 1348
pTimestampAge
LocInjectPositionReq, 384
pTimestampUtc
LocInjectPositionReq, 384
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pTokenBucket
swiQosFlow, 939
pTos
swiQosFilter, 935
pTotalBytesRX
QosEventInfo, 802
slqsWdsEventInfo, 887
pTotalBytesTX
QosEventInfo, 802
slqsWdsEventInfo, 887
pTotalNum
wcdmaLongMsgDecodingParams, 1358
pTrackingStatus
unpack_pds_GetPDSState_t, 1144
pTrafficClass
swiQosFlow, 939
pTranDstPort
swiQosFilter, 935
pTranSrcPort
swiQosFilter, 935
pTransLayerInfo
_getTransLayerInfoResp, 66
_transLayerInfoNotification, 112
sms_getTransLayerInfo, 890
pTransferRouteMTMessageInfo
SMSEventInfo_s, 907
pTransferStatInd
getDUNCallInfoReq, 263
pack_wds_SLQSGetDUNCallInfo_t, 717
pack_wds_SLQSWdsSetEventReport_t, 727
wdsSetEventReportReq, 1397
pTransferStatusReport
sms_setRoutesReq, 899
smsSetRoutesReq, 913
pTrueIMSI
nasGet3GPP2SubscriptionInfoResp, 532

unpack_nas_SLQSNasGet3GPP2Subscription_t, 1124
pTxFilter
swiQosModifyReq, 940
swiQosReq, 941
pTxFlow
swiQosGranted, 939
swiQosModifyReq, 940
swiQosReq, 941
pTxQFilter
QosFlowInfo, 803
pTxQFlowGranted
QosFlowInfo, 803
pTypeCode
USSDRespFNetwork, 1263
pUATI
GetHRPDStatsResp, 269
unpack_nas_SLQSSwiGetHRPDStats_t, 1134
pUDPDstPort
swiQosFilter, 935
pUDPSrcPort
swiQosFilter, 935
pUMTSCellID
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pUMTSGrantedQoS
qmiWdsRunTimeSettings, 800
pUMTSInfo
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pUMTSMinQoS
LibpackProfile3GPP, 338
LibPackprofile_3GPP, 349
Profile3GPP, 756
pUMTSMinQosSigInd
LibpackProfile3GPP, 339
LibPackprofile_3GPP, 350
Profile3GPP, 756
pUMTSReqQoS
LibpackProfile3GPP, 339
LibPackprofile_3GPP, 350
Profile3GPP, 756
pUMTSReqQoSSigInd
LibpackProfile3GPP, 339
LibPackprofile_3GPP, 350
Profile3GPP, 756
pURL
QmiCbkLocGetServerInd, 778
unpack_loc_GetServer_Ind_t, 1080
pUSBComp
USBCompConfig, 1260
USBCompParams, 1262
pUSSDData
USSDNoWaitIndicationInfo, 1263
pUSSDInfo
unpack_voice_SLQSOriginateUSSD_t, 1202
USSResp, 1265
pUSSInfo
voiceSUPSInfo, 1352
pUTSvcRAT
imsaSvcStatusInfo, 319
pUTSvcStatus
imsaSvcStatusInfo, 319
pUUSInFo
pack_voice_SLQSVoiceDialCall_t, 691
voiceCallRequestParams, 1310
pUUSInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
pUimSlotsStatus
UIMGetSlotsStatusResp, 997
unpack_uim_SLQSUIMGetSlotsStatus_t, 1199
pUname
_SLQSAVMSSettingsAPNInfo, 86
pUpdateCompleteStatus
_SLQSOMADMSessionInfo, 91
pUser
pack_wds_SLQSStartDataSession_t, 725
pUserAcknowledgementReq
cdmaMsgDecodingParams, 167
pUserConfigEvents
imsCfgIndRegisterInfo, 320
pUserData
slqssendasyncsmsparams_s, 877
sms_sendAsyncsmsParams, 897
pUserId
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pUserIdSize
LibpackProfile3GPP2, 344
LibPackprofile_3GPP2, 355
Profile3GPP2, 761
pUsername
LibpackProfile3GPP, 339
LibPackprofile_3GPP, 350
pack_wds_SetDefaultProfile_t, 710
Profile3GPP, 756
qmiWdsRunTimeSettings, 800
ssdatasession_params, 921
pUsernameSize
LibpackProfile3GPP, 339
LibPackprofile_3GPP, 350
Profile3GPP, 756
pUtServiceRat
IMSAServiceStatus, 317
pUtServiceStatus
IMSAServiceStatus, 317
pV4sessionId
WdsByteTotals, 1381
WdsConnectionRate, 1383
WdsPktStatisticsResp, 1394
pV6sessionId
WdsByteTotals, 1381

WdsConnectionRate, 1383
WdsPktStatisticsResp, 1394
pVOIPSvcRAT
imsaSvcStatusInfo, 319
pVOIPSvcStatus
imsaSvcStatusInfo, 319
pVTSvcRAT
imsaSvcStatusInfo, 319
pVTSvcStatus
imsaSvcStatusInfo, 319
pVerboseFailReasonType
unpack_wds_SLQSStartDataSession_t, 1256
pVerboseFailureReason
unpack_wds_SLQSStartDataSession_t, 1256
pVersionString
_SLQSSwiGetOSInfoParams, 97
_SLQSSwiSetOSInfoParams, 99
pVertConfidence
LocInjectPositionReq, 384
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pVertReliability
LocInjectPositionReq, 384
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pVertUnc
LocInjectPositionReq, 384
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocPositionReportInd, 787
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_PositionRpt_Ind_t, 1087
pVerticalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 734
pVerticalUnc
pack_pds_SLQSPDSInjectPositionData_t, 654
PDSPositionData, 734
pVoIPConfigEvents
imsCfgIndRegisterInfo, 320
pVoiceDomainPref
pack_voice_SLQSVoiceGetConfig_t, 695
voiceGetConfigReq, 1329
pVoiceDomainPrefStatus
unpack_voice_SLQSVoiceSetConfig_t, 1225
voiceSetConfigResp, 1345
pVoicePrivacy
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1207
unpack_voice_SLQSVoiceGetCallInfo_t, 1212
voiceCallInfoResp, 1308
voiceGetAllCallInfo, 1316
pVoipServiceRat
IMSAServiceStatus, 317
pVoipServiceStatus
IMSAServiceStatus, 317
pVolume
SetM2MAudioProfileReq, 852
pVsServiceRat
IMSAServiceStatus, 317
pVsServiceStatus
IMSAServiceStatus, 317
pVtServiceRat
IMSAServiceStatus, 317
pVtServiceStatus
IMSAServiceStatus, 317
pWCDMABER
GetErrRateResp, 268
unpack_nas_SLQSGetErrorRate_t, 1107
pWCDMACallBarringSysInfo
nasGetSysInfoResp, 538
nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pWCDMACipherDomainSysInfo
nasGetSysInfoResp, 538
nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pWCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pWCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pWCDMAECIOThreshList
nas_WCDMAECIOThresh, 523
WCDMAECIOThresh, 1355
pWCDMAInfoLTENeighborCell
nasCellLocationInfoResp, 530
unpack_nas_SLQSNasGetCellLocationInfo_t, 1125
pWCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pWCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 632
setSignalStrengthInfo, 861
pWCDMARSSIThreshList
nas_WCDMARSSIThresh, 525
WCDMARSSIThresh, 1361
pWCDMASSInfo
nasGetSigInfoResp, 535
pWCDMASigInfo
nasSigInfo, 563
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1129
pWCDMASrvStatusInfo
nasGetSysInfoResp, 538
nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pWCDMASysInfo
nasGetSysInfoResp, 538

nasSysInfo, 569
unpack_nas_SLQSGetSysInfo_t, 1119
unpack_nas_SLQSSysInfoCallback_ind_t, 1139
pWcdmaUARFCN
nasSwiGetChannelLockResp, 564
nasSwiSetChannelLockReq, 566
pack_nas_SLQSNASSwiSetChannelLock_t, 638
unpack_nas_SLQSNASSwiGetChannelLock_t, 1130
pWifiState
pack_pds_SLQSSetPositionMethodState_t, 656
PDSPosMethodStateReq, 734
pXid
QmiCbkLocBestAvailPosInd, 775
unpack_loc_BestAvailPos_Ind_t, 1076
pXtraDataState
pack_pds_SLQSSetPositionMethodState_t, 656
PDSPosMethodStateReq, 734
pXtraTimeState
pack_pds_SLQSSetPositionMethodState_t, 656
PDSPosMethodStateReq, 735
pack_dms_ActivateAutomatic
dms.h, 1412
pack_dms_ActivateAutomatic_t, 590
actCode, 590
pack_dms_GetActivationState
dms.h, 1412
pack_dms_GetBandCapability
dms.h, 1412
pack_dms_GetCrashAction
dms.h, 1413
pack_dms_GetCustFeature
dms.h, 1413
pack_dms_GetCustFeaturesV2
dms.h, 1413
pack_dms_GetCustFeaturesV2_t, 590
cust_id, 591
list_type, 591
Tlvresult, 591
pack_dms_GetDeviceCap
dms.h, 1414
pack_dms_GetDeviceCapabilities
dms.h, 1414
pack_dms_GetDeviceHardwareRev
dms.h, 1414
pack_dms_GetDeviceMfr
dms.h, 1415
pack_dms_GetDeviceSerialNumbers
dms.h, 1415
pack_dms_GetFSN
dms.h, 1417
pack_dms_GetFirmwareInfo
dms.h, 1416
pack_dms_GetFirmwareRevision
dms.h, 1416
pack_dms_GetFirmwareRevisions
dms.h, 1416
pack_dms_GetHardwareRevision
dms.h, 1417
pack_dms_GetIMSI
dms.h, 1417
pack_dms_GetManufacturer
dms.h, 1418
pack_dms_GetModelID
dms.h, 1418
pack_dms_GetNetworkTime
dms.h, 1418
pack_dms_GetOfflineReason
dms.h, 1419
pack_dms_GetPRLVersion
dms.h, 1419
pack_dms_GetPower
dms.h, 1419
pack_dms_GetSerialNumbers
dms.h, 1420
pack_dms_GetUSBComp
dms.h, 1420
pack_dms_GetVoiceNumber
dms.h, 1420
pack_dms_ResetToFactoryDefaults
dms.h, 1421
pack_dms_ResetToFactoryDefaults_t, 591
spc, 591
pack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1424
pack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1424
pack_dms_SLQSDmsSwiIndicationRegister_t, 597
resetInfoInd, 598
pack_dms_SLQSGetBandCapability
dms.h, 1425
pack_dms_SLQSGetERIFile
dms.h, 1425
pack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1426
pack_dms_SLQSSwiGetCrashInfo
dms.h, 1426
pack_dms_SLQSSwiGetCrashInfo_t, 598
clear, 598
pack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1426
pack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1427
pack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1427
pack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1427
pack_dms_SLQSSwiGetHostDevInfo
dms.h, 1428
pack_dms_SLQSSwiGetOSInfo
dms.h, 1428
pack_dms_SLQSSwiGetSerialNoExt
dms.h, 1428
pack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1429
pack_dms_SLQSSwiSetDyingGaspCfg_t, 598

pDestSMSContent, 598
pDestSMSNum, 598
pack_dms_SLQSSwiSetHostDevInfo
dms.h, 1429
pack_dms_SLQSSwiSetHostDevInfo_t, 599
manString, 599
modelString, 599
plasmaIDString, 599
swVerString, 599
pack_dms_SLQSSwiSetOSInfo
dms.h, 1429
pack_dms_SLQSSwiSetOSInfo_t, 599
nameString, 600
versionString, 600
pack_dms_SLQSUIMGetState
dms.h, 1430
pack_dms_SetActivationStatusCallback
dms.h, 1421
pack_dms_SetActivationStatusCallback_t, 591
activationState, 592
pack_dms_SetCrashAction
dms.h, 1422
pack_dms_SetCrashAction_t, 592
crashAction, 592
pack_dms_SetCustFeature
dms.h, 1422
pack_dms_SetCustFeature_t, 592
DHCPRelayEnabled, 594
DisableIMSI, 594
GPSLPM, 594
GPSSel, 594
GpsEnable, 594
IPFamSupport, 594
IsVoiceEnabled, 594
RMAutoConnect, 594
SMSSupport, 594
pack_dms_SetCustFeaturesV2
dms.h, 1422
pack_dms_SetCustFeaturesV2_t, 594
cust_id, 595
cust_value, 595
Tlvresult, 595
value_length, 595
pack_dms_SetEventReport
dms.h, 1423
pack_dms_SetEventReport_t, 595
mode, 595
pack_dms_SetFirmwarePreference
dms.h, 1423
pack_dms_SetPower
dms.h, 1423
pack_dms_SetPower_t, 596
mode, 596
Tlvresult, 596
pack_dms_SetUSBComp
dms.h, 1424
pack_dms_SetUSBComp_t, 596
Tlvresult, 597
USBComp, 597
pack_dms_UIMChangePIN
dms.h, 1430
pack_dms_UIMChangePIN_t, 600
id, 600
newValue, 600
oldValue, 600
pack_dms_UIMGetControlKeyStatus
dms.h, 1430
pack_dms_UIMGetControlKeyStatus_t, 601
facility, 601
pack_dms_UIMGetICCID
dms.h, 1431
pack_dms_UIMGetICCID_t, 601
Tlvresult, 601
pack_dms_UIMGetPINStatus
dms.h, 1431
pack_dms_UIMSetControlKeyProtection
dms.h, 1432
pack_dms_UIMSetControlKeyProtection_t, 601
facility, 602
facilityCk, 602
facilityState, 602
pack_dms_UIMSetPINProtection
dms.h, 1432
pack_dms_UIMSetPINProtection_t, 602
bEnable, 603
id, 603
value, 603
pack_dms_UIMUnblockControlKey
dms.h, 1432
pack_dms_UIMUnblockControlKey_t, 603
facility, 603
facilityCk, 603
pack_dms_UIMUnblockPIN
dms.h, 1433
pack_dms_UIMUnblockPIN_t, 603
id, 604
newPin, 604
pukValue, 604
pack_dms_UIMVerifyPIN
dms.h, 1433
pack_dms_UIMVerifyPIN_t, 604
id, 604
value, 604
pack_dms_ValidateSPC
dms.h, 1433
pack_fms_GetImagesPreference
fms.h, 1458
pack_fms_GetImagesPreference_t, 605
Tlvresult, 605
pack_fms_GetStoredImages
fms.h, 1458
pack_fms_GetStoredImages_t, 605
Tlvresult, 605
pack_fms_SetImagesPreference
fms.h, 1458
pack_fms_SetImagesPreference_t, 605

bForceDownload, 606
imageListSize, 606
modemindex, 606
pImageList, 606
Tlvresult, 606
pack_loc_Delete_Assist_Data_t, 606
pBdsSVInfo, 607
pCellDb, 607
pClkInfo, 607
pGnssData, 607
pSVInfo, 607
Tlvresult, 607
pack_loc_DeleteAssistData
loc.h, 1465
pack_loc_EventRegister
loc.h, 1465
pack_loc_EventRegister_t, 607
eventRegister, 609
Tlvresult, 609
pack_loc_SLQSLOCGetBestAvailPos
loc.h, 1466
pack_loc_SLQSLOCGetBestAvailPos_t, 610
Tlvresult, 610
xid, 610
pack_loc_SLQSLOCGetServer
loc.h, 1466
pack_loc_SLQSLOCGetServer_t, 611
has_serverAddrTypeMask, 611
serverAddrTypeMask, 611
serverType, 611
pack_loc_SLQSLOCInjectPosition
loc.h, 1467
pack_loc_SLQSLOCInjectPosition_t, 611
altitudeSrcInfo, 615
altitudeWrtEllipsoid, 615
altitudeWrtMeanSeaLevel, 615
has_altitudeSrcInfo, 615
has_altitudeWrtEllipsoid, 615
has_horConfidence, 615
has_horReliability, 615
has_horUncCircular, 615
has_latitude, 615
has_longitude, 615
has_positionSrc, 615
has_rawHorConfidence, 615
has_rawHorUncCircular, 615
has_timestampAge, 615
has_timestampUtc, 615
has_vertConfidence, 615
has_vertRelicability, 615
has_vertUnc, 616
horConfidence, 616
horReliability, 616
horUncCircular, 616
latitude, 616
longitude, 616
positionSrc, 616
rawHorConfidence, 616
rawHorUncCircular, 616
timestampAge, 616
timestampUtc, 616
vertConfidence, 616
vertReliability, 616
vertUnc, 616
pack_loc_SLQSLOCInjectSensorData
loc.h, 1467
pack_loc_SLQSLOCInjectSensorData_t, 616
accelTemp, 617
acceleroData, 617
acceleroTimeSrc, 617
gyroData, 617
gyroTemp, 617
gyroTimeSrc, 618
has_accelTemp, 618
has_acceleroTimeSrc, 618
has_accleroData, 618
has_gyroData, 618
has_gyroTemp, 618
has_gyroTimeSrc, 618
has_opaqueId, 618
opaqueId, 618
pack_loc_SLQSLOCInjectUTCTime
loc.h, 1468
pack_loc_SLQSLOCInjectUTCTime_t, 618
timeMsec, 618
timeUncMsec, 618
pack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1468
pack_loc_SLQSLOCSetCradleMountConfig_t, 618
confidence, 619
has_confidence, 619
state, 619
pack_loc_SetExtPowerState
loc.h, 1465
pack_loc_SetExtPowerState_t, 609
extPowerState, 609
Tlvresult, 609
pack_loc_SetOperationMode
loc.h, 1466
pack_loc_SetOperationMode_t, 609
mode, 610
Tlvresult, 610
pack_loc_Start
loc.h, 1468
pack_loc_Start_t, 619
pApplicationInfo, 620
pConfigAltitudeAssumed, 620
pHorizontalAccuracyLvl, 621
pIntermediateReportState, 621
pMinIntervalTime, 621
pRecurrenceType, 621
SessionId, 621
Tlvresult, 621
pack_loc_Stop
loc.h, 1469
pack_loc_Stop_t, 621

SessionId, 621
TlvResult, 621
pack_nas_GetACCOLC
nas.h, 1485
pack_nas_GetANAAAAuthenticationStatus
nas.h, 1485
pack_nas_GetCDMANetworkParameters
nas.h, 1485
pack_nas_GetHomeNetwork
nas.h, 1486
pack_nas_GetHomeNetwork3GPP2
nas.h, 1486
pack_nas_GetNetworkPreference
nas.h, 1486
pack_nas_GetRFInfo
nas.h, 1486
pack_nas_GetServingNetwork
nas.h, 1487
pack_nas_GetServingNetworkCapabilities
nas.h, 1487
pack_nas_GetSignalStrengths
nas.h, 1487
pack_nas_InitiateDomainAttach
nas.h, 1488
pack_nas_InitiateDomainAttach_t, 621
action, 622
pack_nas_PerformNetworkScan
nas.h, 1488
pack_nas_SLQSConfigSigInfo
nas.h, 1490
pack_nas_SLQSConfigSigInfo_t, 625
pECIOThresh, 626
pIOThresh, 626
pLTESNRThresh, 626
pLTESigRptCfg, 626
pRSRPThresh, 626
pRSRQThresh, 626
pRSSIThresh, 626
pack_nas_SLQSGetErrorRate
nas.h, 1490
pack_nas_SLQSGetNetworkTime
nas.h, 1491
pack_nas_SLQSGetOperatorNameData
nas.h, 1491
pack_nas_SLQSGetPLMNName
nas.h, 1492
pack_nas_SLQSGetPLMNName_t, 626
mcc, 627
mnc, 627
pMncPcsStatus, 627
pack_nas_SLQSGetServingSystem
nas.h, 1492
pack_nas_SLQSGetSignalStrength
nas.h, 1492
pack_nas_SLQSGetSysInfo
nas.h, 1493
pack_nas_SLQSGetSysSelectionPref
nas.h, 1493
pack_nas_SLQSInitiateNetworkRegistration
nas.h, 1493
pack_nas_SLQSInitiateNetworkRegistration_t, 627
pChangeDuration, 627
pMNRInfo, 628
pMncPcsDigitStatus, 628
regAction, 628
pack_nas_SLQSNASSwiGetChannelLock
nas.h, 1496
pack_nas_SLQSNASSwiSetChannelLock
nas.h, 1497
pack_nas_SLQSNASSwiSetChannelLock_t, 637
pack_nas_SLQSNasConfigSigInfo2
nas.h, 1494
pack_nas_SLQSNasConfigSigInfo2_t, 628
pHDRIODelta, 631
pHDRIOThresh, 631
pLTESigRptConfig, 632
pack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1494
pack_nas_SLQSNasGet3GPP2Subscription_t, 632
namID, 633
pack_nas_SLQSNasGetCellLocationInfo
nas.h, 1494
pack_nas_SLQSNasGetHDRColorCode
nas.h, 1495
pack_nas_SLQSNasGetSigInfo
nas.h, 1495
pack_nas_SLQSNasGetTxRxInfo
nas.h, 1495
pack_nas_SLQSNasGetTxRxInfo_t, 633
radio_if, 633
pack_nas_SLQSNasIndicationRegisterExt
nas.h, 1496
pack_nas_SLQSNasIndicationRegisterExt_t, 633
pDDTMInd, 635
pDualStandByPrefInd, 635
pErrorRateInd, 636
pHDRSessionCloseInd, 636
pLTECphyCa, 636
pManagedRoamingInd, 636
pNetworkRejectInd, 636
pNetworkTimeInd, 636
pServingSystemInd, 636
pSignalStrengthInd, 636
pSubscriptionInfoInd, 636
pSuppressSysInfoInd, 636
pSysInfoInd, 636
pSystemSelectionInd, 636
pack_nas_SLQSNasSwiIndicationRegister
nas.h, 1496
pack_nas_SLQSNasSwiIndicationRegister_t, 636
gsmUmtsDl, 637
gsmUmtsUl, 637
lteEmmDl, 637
lteEmmUl, 637
lteEsmDl, 637
lteEsmUl, 637

pRankIndicatorInd, 637
pTimer, 637
pack_nas_SLQSNasSwiModemStatus
nas.h, 1497
pack_nas_SLQSSetBandPreference
nas.h, 1497
pack_nas_SLQSSetBandPreference_t, 638
bandPref, 639
pack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1498
pack_nas_SLQSSetSignalStrengthsCallback_t, 640
bEnable, 640
pSigIndReq, 640
pack_nas_SLQSSetSysSelectionPref
nas.h, 1498
pack_nas_SLQSSetSysSelectionPref_t, 640
pAcqOrderPref, 645
pBandPref, 645
pCSGID, 645
pChgDuration, 645
pEmerMode, 645
pGWAcqOrderPref, 645
pLTEBandPref, 645
pModePref, 645
pNetSelPref, 645
pPRLPref, 645
pRAT, 645
pRoamPref, 645
pSrvDomainPref, 645
pSrvRegRestriction, 645
pTdscdmaBandPref, 646
pack_nas_SLQSSwiGetHDRPersonality
nas.h, 1498
pack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1499
pack_nas_SLQSSwiGetHRPDStats
nas.h, 1499
pack_nas_SLQSSwiGetLteCQI
nas.h, 1499
pack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1500
pack_nas_SLQSSwiNetworkDebug
nas.h, 1500
pack_nas_SLQSSwiPSDetach
nas.h, 1500
pack_nas_SLQSSwiPSDetach_t, 646
pDetachAction, 646
pack_nas_SetACCOLC
nas.h, 1488
pack_nas_SetACCOLC_t, 622
accolc, 622
spc, 622
pack_nas_SetCDMANetworkParameters
nas.h, 1489
pack_nas_SetCDMANetworkParameters_t, 622
pApplication, 623
pBroadcast, 623
pCustomSCP, 623
pForceRev0, 624
pProtocol, 624
pRoaming, 624
pSPC, 624
pack_nas_SetLURejectCallback
nas.h, 1489
pack_nas_SetNetworkPreference
nas.h, 1489
pack_nas_SetNetworkPreference_t, 624
Duration, 624
TechnologyPref, 624
Tlvresult, 624
pack_nas_SetRFInfoCallback
nas.h, 1490
pack_nas_SlqsGetLTECphyCAInfo
nas.h, 1491
pack_pds_ForceXTRADownload
pds.h, 1522
pack_pds_GetPDSDefaults
pds.h, 1523
pack_pds_GetPDSState
pds.h, 1523
pack_pds_GetPortAutomaticTracking
pds.h, 1523
pack_pds_GetServiceAutomaticTracking
pds.h, 1524
pack_pds_GetXTRAAutomaticDownload
pds.h, 1524
pack_pds_GetXTRANetwork
pds.h, 1524
pack_pds_GetXTRAValidity
pds.h, 1525
pack_pds_PDSInjectTimeReference
pds.h, 1525
pack_pds_PDSInjectTimeReference_t, 646
systemDiscontinuities, 646
systemTime, 646
pack_pds_ResetPDSData
pds.h, 1525
pack_pds_ResetPDSData_t, 647
pCellDataMask, 647
pGPSDataMask, 647
pack_pds_SLQSGetAGPSConfig
pds.h, 1528
pack_pds_SLQSGetAGPSConfig_t, 651
pNetworkMode, 652
pack_pds_SLQSGetGPSStateInfo
pds.h, 1529
pack_pds_SLQSPDSDeterminePosition
pds.h, 1529
pack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1529
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 652
forceFlag, 652
timeBase, 652
timeMsec, 652
timeUncMsec, 652

pack_pds_SLQSPDSInjectPositionData
pds.h, 1530
pack_pds_SLQSPDSInjectPositionData_t, 652
pAltitudeWrtEllipsoid, 654
pAltitudeWrtSealevel, 654
pHorizontalConfidence, 654
pHorizontalUncCircular, 654
pLatitude, 654
pLongitude, 654
pPositionSource, 654
pTimeStamp, 654
pTimeType, 654
pVerticalConfidence, 654
pVerticalUnc, 654
pack_pds_SLQSSetAGPSConfig
pds.h, 1530
pack_pds_SLQSSetAGPSConfig_t, 654
pNetworkMode, 655
pServerAddress, 655
pServerPort, 655
pServerURL, 655
pack_pds_SLQSSetPositionMethodState
pds.h, 1531
pack_pds_SLQSSetPositionMethodState_t, 655
pWifiState, 656
pXtraDataState, 656
pXtraTimeState, 656
pack_pds_SetEventReportCallback
pds.h, 1526
pack_pds_SetEventReportCallback_t, 647
posDataNmea, 648
rptPosData, 648
pack_pds_SetPDSDefaults
pds.h, 1526
pack_pds_SetPDSDefaults_t, 648
accuracy, 649
interval, 649
operation, 649
timeout, 649
pack_pds_SetPDSState
pds.h, 1527
pack_pds_SetPDSState_t, 649
enable, 649
pack_pds_SetPortAutomaticTracking
pds.h, 1527
pack_pds_SetPortAutomaticTracking_t, 649
bAuto, 650
pack_pds_SetServiceAutomaticTracking
pds.h, 1527
pack_pds_SetServiceAutomaticTracking_t, 650
bAuto, 650
pack_pds_SetXTRAAutomaticDownload
pds.h, 1528
pack_pds_SetXTRAAutomaticDownload_t, 650
bEnabled, 651
interval, 651
pack_pds_SetXTRANetwork
pds.h, 1528
pack_pds_SetXTRANetwork_t, 651
preference, 651
pack_pds_StartPDSTrackingSessionExt
pds.h, 1531
pack_pds_StartPDSTrackingSessionExt_t, 656
fixAccuracy, 657
fixCount, 657
fixInterval, 657
fixTimeout, 657
sessionControl, 657
sessionOperation, 657
sessionServerOption, 657
sessionType, 657
pack_pds_StopPDSTrackingSession
pds.h, 1531
pack_qmi_t, 657
msgid, 658
svc, 658
timeout, 658
xid, 658
pack_qos_SLQSQosGetNetworkStatus
qos.h, 1974
pack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 1974
pack_qos_SLQSQosSwiReadApnExtraParams_t, 658
apnId, 658
pack_qos_SLQSQosSwiReadDataStats
qos.h, 1975
pack_qos_SLQSQosSwiReadDataStats_t, 658
apnId, 659
pack_qos_SLQSSetQosEventCallback
qos.h, 1976
pack_qos_SLQSSetQosEventCallback_t, 659
enable, 659
pack_rms_GetSMSWake
rms.h, 1981
pack_rms_SetSMSWake
rms.h, 1981
pack_rms_SetSMSWake_t, 659
enabled, 659
wake_mask, 660
pack_sar_SLQSGetRfSarState
sar.h, 1982
pack_sar_SLQSSetRfSarState
sar.h, 1983
pack_sar_SLQSSetRfSarState_t, 660
RfSarState, 660
pack_sms_GetSMSCAddress
sms.h, 1989
pack_sms_SLQSDeleteSMS
sms.h, 1991
pack_sms_SLQSDeleteSMS_t, 663
pMessageIndex, 663
pMessageMode, 663
pMessageTag, 663
storageType, 663
pack_sms_SLQSGetIndicationRegister
sms.h, 1991

pack_sms_SLQSGetMessageWaiting
sms.h, 1992
pack_sms_SLQSGetSMS
sms.h, 1992
pack_sms_SLQSGetSMS_t, 663
messageIndex, 664
pMessageMode, 664
storageType, 664
pack_sms_SLQSGetSMSList
sms.h, 1993
pack_sms_SLQSGetSMSList_t, 664
pMessageMode, 665
pRequestedTag, 665
storageType, 665
pack_sms_SLQSGetSmsBroadcastConfig
sms.h, 1992
pack_sms_SLQSGetSmsBroadcastConfig_t, 664
mode, 664
pack_sms_SLQSGetTransLayerInfo
sms.h, 1993
pack_sms_SLQSGetTransNWRegInfo
sms.h, 1994
pack_sms_SLQSModifySMSStatus
sms.h, 1994
pack_sms_SLQSModifySMSStatus_t, 665
messageIndex, 666
messageTag, 666
pMessageMode, 666
storageType, 666
pack_sms_SLQSSendAsyncSMS
sms.h, 1994
pack_sms_SLQSSendAsyncSMS_t, 666
pSendSmsParams, 666
pack_sms_SLQSSetIndicationRegister
sms.h, 1995
pack_sms_SLQSSetIndicationRegister_t, 666
pSetIndicationRegReq, 667
pack_sms_SLQSSetSmsBroadcastActivation
sms.h, 1995
pack_sms_SLQSSetSmsBroadcastActivation_t, 667
broadcastActivate, 667
mode, 667
pack_sms_SLQSSetSmsBroadcastConfig
sms.h, 1995
pack_sms_SLQSSetSmsBroadcastConfig_t, 667
mode, 668
pBroadcastConfig, 668
pCDMABroadcastConfig, 668
pack_sms_SLQSSetSmsStorage
sms.h, 1996
pack_sms_SLQSSetSmsStorage_t, 668
smsStorage, 668
pack_sms_SLQSSmsGetMaxStorageSize
sms.h, 1996
pack_sms_SLQSSmsGetMaxStorageSize_t, 668
pMaxStorageSizeReq, 669
pack_sms_SLQSSmsGetMessageProtocol
sms.h, 1996
pack_sms_SLQSSmsSetRoutes
sms.h, 1997
pack_sms_SLQSSmsSetRoutes_t, 669
pSetRoutesReq, 669
pack_sms_SLQSSwiGetSMSStorage
sms.h, 1997
pack_sms_SaveSMS
sms.h, 1989
pack_sms_SaveSMS_t, 660
messageFormat, 661
messageSize, 661
pMessage, 661
storageType, 661
pack_sms_SendSMS
sms.h, 1990
pack_sms_SendSMS_t, 661
messageFormat, 662
messageSize, 662
pLinktimer, 662
pMessage, 662
pack_sms_SetNewSMSCallback
sms.h, 1990
pack_sms_SetNewSMSCallback_t, 662
status, 662
pack_sms_SetSMSCAddress
sms.h, 1991
pack_sms_SetSMSCAddress_t, 662
pSMSCAddress, 662
pSMSCType, 662
pack_swiloc_SwiLocGetAutoStart
swiloc.h, 2008
pack_swiloc_SwiLocSetAutoStart
swiloc.h, 2008
pack_swiloc_SwiLocSetAutoStart_t, 669
fix_rate, 670
fix_type, 670
function, 670
max_dist, 671
max_time, 671
set_fix_rate, 671
set_fix_type, 671
set_function, 671
set_max_dist, 671
set_max_time, 671
pack_swioma_SLQSOMADMAlertCallback
swioma.h, 2010
pack_swioma_SLQSOMADMCancelSession
swioma.h, 2011
pack_swioma_SLQSOMADMCancelSession_t, 671
sessionType, 671
pack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 2011
pack_swioma_SLQSOMADMGetSessionInfo_t, 671
SessionType, 672
pack_swioma_SLQSOMADMGetSettings
swioma.h, 2012
pack_swioma_SLQSOMADMSendSelection
swioma.h, 2012

pack_swioma_SLQSOMADMSendSelection_t, 672
pDeferTime, 672
pRejectReason, 672
selection, 672
pack_swioma_SLQSOMADMSetSettings
swioma.h, 2013
pack_swioma_SLQSOMADMSetSettings_t, 672
FOTAdownload, 673
pAutosdm, 673
pFwAutoCheck, 673
pack_swioma_SLQSOMADMStartSession
swioma.h, 2014
pack_swioma_SLQSOMADMStartSession_t, 673
sessionType, 674
pack_uim_ChangePin
uim.h, 2021
pack_uim_ChangePin_t, 674
changePIN, 675
EncryptedPIN1, 675
pIndicationToken, 675
pKeyReferenceID, 675
sessionInfo, 675
Tlvresult, 675
pack_uim_GetCardStatus
uim.h, 2022
pack_uim_ReadTransparent
uim.h, 2022
pack_uim_ReadTransparent_t, 675
fileIndex, 676
pEncryptData, 676
pIndicationToken, 676
readTransparent, 676
sessionInfo, 676
Tlvresult, 676
pack_uim_SLQSUIMAuthenticate
uim.h, 2023
pack_uim_SLQSUIMAuthenticate_t, 677
authData, 678
pIndicationToken, 678
sessionInfo, 678
pack_uim_SLQSUIMDepersonalization
uim.h, 2023
pack_uim_SLQSUIMDepersonalization_t, 678
depersonilisationInfo, 678
pack_uim_SLQSUIMEventRegister
uim.h, 2023
pack_uim_SLQSUIMEventRegister_t, 678
eventMask, 678
pack_uim_SLQSUIMGetConfiguration
uim.h, 2024
pack_uim_SLQSUIMGetConfiguration_t, 678
pConfigurationMask, 679
pack_uim_SLQSUIMGetFileAttributes
uim.h, 2024
pack_uim_SLQSUIMGetFileAttributes_t, 679
fileIndex, 680
pIndicationToken, 680
sessionInfo, 680
pack_uim_SLQSUIMGetSlotsStatus
uim.h, 2025
pack_uim_SLQSUIMPowerDown
uim.h, 2025
pack_uim_SLQSUIMPowerDown_t, 680
slot, 680
pack_uim_SLQSUIMPowerUp
uim.h, 2025
pack_uim_SLQSUIMPowerUp_t, 680
pIgnoreHotSwapSwitch, 681
slot, 681
pack_uim_SLQSUIMRefreshComplete
uim.h, 2026
pack_uim_SLQSUIMRefreshComplete_t, 681
refreshComplete, 681
sessionInfo, 681
pack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 2026
pack_uim_SLQSUIMRefreshGetLastEvent_t, 681
sessionInfo, 682
pack_uim_SLQSUIMRefreshOK
uim.h, 2026
pack_uim_SLQSUIMRefreshOK_t, 682
OKtoRefresh, 682
sessionInfo, 682
pack_uim_SLQSUIMRefreshRegister
uim.h, 2027
pack_uim_SLQSUIMRefreshRegister_t, 682
regRefresh, 683
sessionInfo, 683
pack_uim_SLQSUIMReset
uim.h, 2027
pack_uim_SLQSUIMSwitchSlot
uim.h, 2027
pack_uim_SLQSUIMSwitchSlot_t, 683
bLogicalSlot, 683
ulPhysicalSlot, 683
pack_uim_SetPinProtection
uim.h, 2022
pack_uim_SetPinProtection_t, 676
EncryptedPIN1, 677
pIndicationToken, 677
pKeyReferenceID, 677
pinProtection, 677
sessionInfo, 677
Tlvresult, 677
pack_uim_UnblockPin
uim.h, 2028
pack_uim_UnblockPin_t, 684
EncryptedPIN1, 684
pIndicationToken, 684
pKeyReferenceID, 684
pinProtection, 684
sessionInfo, 684
Tlvresult, 684
pack_uim_VerifyPin
uim.h, 2028
pack_uim_VerifyPin_t, 685

pEncryptedPIN1, 685
pIndicationToken, 685
pKeyReferenceID, 685
sessionInfo, 686
Tlvresult, 686
verifyPIN, 686
pack_voice_AnswerUSSD
voice.h, 2041
pack_voice_AnswerUSSD_t, 686
pInfo, 686
pack_voice_CancelUSSD
voice.h, 2041
pack_voice_OriginateUSSD
voice.h, 2042
pack_voice_OriginateUSSD_t, 686
pInfo, 686
pack_voice_SLQSOriginateUSSD
voice.h, 2042
pack_voice_SLQSOriginateUSSD_t, 686
ussDCS, 687
ussData, 687
ussLen, 687
pack_voice_SLQSVoiceALSSelectLine
voice.h, 2042
pack_voice_SLQSVoiceALSSelectLine_t, 687
lineValue, 687
pack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 2043
pack_voice_SLQSVoiceALSSetLineSwitching_t, 687
switchOption, 688
pack_voice_SLQSVoiceAnswerCall
voice.h, 2043
pack_voice_SLQSVoiceAnswerCall_t, 688
pCallId, 688
pack_voice_SLQSVoiceBindSubscription
voice.h, 2044
pack_voice_SLQSVoiceBindSubscription_t, 688
subsType, 689
pack_voice_SLQSVoiceBurstDTMF
voice.h, 2044
pack_voice_SLQSVoiceBurstDTMF_t, 689
pack_voice_SLQSVoiceDialCall
voice.h, 2044
pack_voice_SLQSVoiceDialCall_t, 689
callNumber, 691
pCLIRType, 691
pCUGInfo, 691
pCallPartySubAdd, 691
pCallType, 691
pEmergencyCategory, 691
pSvcType, 691
pUUSInFo, 691
pack_voice_SLQSVoiceEndCall
voice.h, 2045
pack_voice_SLQSVoiceEndCall_t, 691
pCallId, 691
pack_voice_SLQSVoiceGetAllCallInfo
voice.h, 2045
pack_voice_SLQSVoiceGetCLIP
voice.h, 2047
pack_voice_SLQSVoiceGetCLIR
voice.h, 2047
pack_voice_SLQSVoiceGetCNAP
voice.h, 2048
pack_voice_SLQSVoiceGetCOLP
voice.h, 2048
pack_voice_SLQSVoiceGetCOLR
voice.h, 2048
pack_voice_SLQSVoiceGetCallBarring
voice.h, 2045
pack_voice_SLQSVoiceGetCallBarring_t, 691
pSvcClass, 692
reason, 692
pack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 2046
pack_voice_SLQSVoiceGetCallForwardingStatus_t, 692
pSvcClass, 693
Reason, 693
pack_voice_SLQSVoiceGetCallInfo
voice.h, 2046
pack_voice_SLQSVoiceGetCallInfo_t, 693
callID, 693
pack_voice_SLQSVoiceGetCallWaiting
voice.h, 2047
pack_voice_SLQSVoiceGetCallWaiting_t, 693
pSvcClass, 694
pack_voice_SLQSVoiceGetConfig
voice.h, 2049
pack_voice_SLQSVoiceGetConfig_t, 694
pAMRStatus, 695
pAirTimer, 695
pAutoAnswer, 695
pNamID, 695
pPrefVoicePrivacy, 695
pPrefVoiceSO, 695
pRoamTimer, 695
pTTYMode, 695
pVoiceDomainPref, 695
pack_voice_SLQSVoiceIndicationRegister
voice.h, 2049
pack_voice_SLQSVoiceIndicationRegister_t, 696
pRegDTMFEvents, 696
pRegVoicePrivacyEvents, 696
pSuppsNotifEvents, 696
pack_voice_SLQSVoiceManageCalls
voice.h, 2049
pack_voice_SLQSVoiceManageCalls_t, 696
pCallID, 697
SUPSType, 697
pack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 2050
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 697
pack_voice_SLQSVoiceSendFlash
voice.h, 2050
pack_voice_SLQSVoiceSendFlash_t, 698
pCallID, 698

pFlashPayLd, 698
pFlashType, 698
pack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 2050
pack_voice_SLQSVoiceSetCallBarringPassword_t, 698
newPasswd, 699
newPasswdAgain, 699
oldPasswd, 699
Reason, 699
pack_voice_SLQSVoiceSetConfig
voice.h, 2051
pack_voice_SLQSVoiceSetConfig_t, 699
pAirTimerConfig, 701
pAutoAnswer, 701
pPrefVoiceDomain, 701
pPrefVoiceSO, 701
pRoamTimerConfig, 701
pTTYMode, 701
pack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 2051
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 701
privacyPref, 701
pack_voice_SLQSVoiceSetSUPSService
voice.h, 2052
pack_voice_SLQSVoiceSetSUPSService_t, 701
pCallBarringPasswd, 703
pCallForwardingNumber, 703
pServiceClass, 703
pTimerVal, 703
reason, 703
voiceSvc, 703
pack_voice_SLQSVoiceStartContDTMF
voice.h, 2052
pack_voice_SLQSVoiceStartContDTMF_t, 703
DTMFdigit, 704
pCallID, 704
pack_voice_SLQSVoiceStopContDTMF
voice.h, 2052
pack_voice_SLQSVoiceStopContDTMF_t, 704
callID, 704
pack_wds_DHCPv4ClientLeaseChange
wds.h, 2070
pack_wds_DHCPv4ClientLeaseChange_t, 704
pEnableNotification, 705
pack_wds_GetAutoconnect
wds.h, 2070
pack_wds_GetByteTotals
wds.h, 2071
pack_wds_GetConnectionRate
wds.h, 2071
pack_wds_GetDataBearerTechnology
wds.h, 2072
pack_wds_GetDefaultProfile
wds.h, 2072
pack_wds_GetDefaultProfile_t, 705
profiletype, 705
pack_wds_GetDefaultProfileNum
wds.h, 2072
pack_wds_GetDefaultProfileNum_t, 705
family, 705
type, 705
pack_wds_GetDormancyState
wds.h, 2073
pack_wds_GetDormancyState_t, 705
pack_wds_GetLastMobileIPError
wds.h, 2073
pack_wds_GetLastMobileIPError_t, 706
pack_wds_GetMobileIP
wds.h, 2074
pack_wds_GetMobileIP_t, 706
pack_wds_GetMobileIPProfile
wds.h, 2074
pack_wds_GetMobileIPProfile_t, 706
index, 706
pack_wds_GetPacketStatistics
wds.h, 2075
pack_wds_GetPacketStatistics_t, 706
pStatMask, 707
pack_wds_GetPacketStatus
wds.h, 2075
pack_wds_GetPacketStatus_t, 707
statmask, 707
pack_wds_GetSessionDuration
wds.h, 2075
pack_wds_GetSessionDuration_t, 707
pack_wds_GetSessionState
wds.h, 2076
pack_wds_RMSetTransferStatistics
wds.h, 2076
pack_wds_RMSetTransferStatistics_t, 707
RmTrasnferStaticsReq, 708
pack_wds_SLQSCreateProfile
wds.h, 2079
pack_wds_SLQSCreateProfile_t, 714
pCurProfile, 715
pProfileId, 715
pProfileType, 715
pack_wds_SLQSDeleteProfile
wds.h, 2080
pack_wds_SLQSDeleteProfile_t, 715
profileIndex, 715
profileType, 715
pack_wds_SLQSGet3GPPConfigItem
wds.h, 2080
pack_wds_SLQSGetCurrDataSystemStat
wds.h, 2080
pack_wds_SLQSGetCurrDataSystemStat_t, 715
pack_wds_SLQSGetCurrentChannelRate
wds.h, 2081
pack_wds_SLQSGetDUNCallInfo
wds.h, 2082
pack_wds_SLQSGetDUNCallInfo_t, 716
Mask, 717
pReportChannelRate, 717
pReportConnStatus, 717
pReportDataBearerTech, 717

pReportDormStatus, 717
pTransferStatInd, 717
pack_wds_SLQSGetDataBearerTechnology
wds.h, 2081
pack_wds_SLQSGetDataBearerTechnology_t, 716
pack_wds_SLQSGetProfileSettings
wds.h, 2082
pack_wds_SLQSGetProfileSettings_t, 717
ProfileId, 718
ProfileType, 718
pack_wds_SLQSGetRuntimeSettings
wds.h, 2083
pack_wds_SLQSGetRuntimeSettings_t, 718
pReqSettings, 719
pack_wds_SLQSModifyProfile
wds.h, 2083
pack_wds_SLQSModifyProfile_t, 719
curProfile, 720
pProfileId, 720
pProfileType, 720
pack_wds_SLQSResetPacketStatics
wds.h, 2083
pack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 2085
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 723
pProfileId, 723
pack_wds_SLQSSGetLoopback
wds.h, 2085
pack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 2086
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 723
pHwConfig, 724
pProfileId, 724
pack_wds_SLQSSSetLoopback
wds.h, 2086
pack_wds_SLQSSSetLoopback_t, 724
loopbackMode, 724
loopbackMultiplier, 724
pack_wds_SLQSSet3GPPConfigItem
wds.h, 2084
pack_wds_SLQSSet3GPPConfigItem_t, 720
p3gppRelease, 721
pProfileList, 721
pack_wds_SLQSSetIPFamilyPreference
wds.h, 2084
pack_wds_SLQSSetIPFamilyPreference_t, 722
IPFamilyPreference, 722
pack_wds_SLQSSetWdsEventCallback
wds.h, 2085
pack_wds_SLQSSetWdsEventCallback_t, 722
currentDataBearer, 722
dataBearer, 723
dataBearerTechExt, 723
dataSystemStatus, 723
dormancyStatus, 723
interval, 723
mobileIP, 723
transferStats, 723
pack_wds_SLQSStartDataSession
wds.h, 2087
pack_wds_SLQSStartDataSession_t, 724
pAuth, 725
pPass, 725
pTech, 725
pUser, 725
pprofileid3gpp, 725
pprofileid3gpp2, 725
pack_wds_SLQSStopDataSession
wds.h, 2087
pack_wds_SLQSStopDataSession_t, 725
psid, 726
pack_wds_SLQSWdsGoActive
wds.h, 2087
pack_wds_SLQSWdsGoDormant
wds.h, 2088
pack_wds_SLQSWdsSetEventReport
wds.h, 2088
pack_wds_SLQSWdsSetEventReport_t, 726
pCurrChannelRateInd, 727
pCurrDataBearerTechInd, 727
pCurrPrefDataSysInd, 727
pDataBearerTechInd, 727
pDataCallStatusChangeInd, 727
pDataSystemStatusChangeInd, 727
pDormancyStatusInd, 727
pMIPStatusInd, 727
pTransferStatInd, 727
pack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 2089
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 728
contextId, 728
contextType, 728
pack_wds_SetAutoconnect
wds.h, 2077
pack_wds_SetAutoconnect_t, 708
acroamsetting, 708
acsetting, 708
pack_wds_SetDefaultProfile
wds.h, 2077
pack_wds_SetDefaultProfile_t, 708
authentication, 710
ipAddress, 710
pApnname, 710
pName, 710
pPassword, 710
pUsername, 710
pdpType, 710
primaryDNS, 710
profileType, 710
secondaryDNS, 710
pack_wds_SetDefaultProfileNum
wds.h, 2077
pack_wds_SetDefaultProfileNum_t, 710
family, 711
index, 711
type, 711

pack_wds_SetMobileIP
wds.h, 2078
pack_wds_SetMobileIP_t, 711
mode, 711
pack_wds_SetMobileIPParameters
wds.h, 2078
pack_wds_SetMobileIPParameters_t, 711
pHA2002bis, 712
pHAAuthenticator, 712
pMode, 712
pReRegPeriod, 712
pReRegTraffic, 712
pRetryInterval, 712
pRetryLimit, 712
pSPC, 712
pack_wds_SetMobileIPProfile
wds.h, 2079
pack_wds_SetMobileIPProfile_t, 712
index, 714
pAAASPI, 714
pAddress, 714
pEnabled, 714
pHASPI, 714
pMNAAA, 714
pMNHA, 714
pNAI, 714
pPrimaryHA, 714
pRevTunneling, 714
pSecondaryHA, 714
spc, 714
PackCreateProfileOut, 728
ExtErrorCode, 729
ProfileIndex, 729
ProfileType, 729
package_name
omaDmFotaTlv, 585
omaDmFotaTlvExt, 587
unpack_omaDmFotaTlv_t, 1142
PackageID
_SwiAvmsEventInd, 100
packageSize
omaDmFotaTlvExt, 587
packageid_str
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
packetSrvStatus
qaGobiApiCbk.h, 1562
packetZone
CDMASysInfo, 174
nas_CDMASysInfo, 425
packetZoneValid
CDMASysInfo, 174
nas_CDMASysInfo, 425
packgetDyingGaspCfg, 729
pDestSMSContent, 729
pDestSMSNum, 729
packgetDyingGaspStatistics, 729
pSMSAttemptedFlag, 730
pTimeStamp, 730
path
fileInfo, 244
uim_fileInfo, 979
pathLen
fileInfo, 244
uim_fileInfo, 979
pbAuto
unpack_pds_GetPortAutomaticTracking_t, 1145
unpack_pds_GetServiceAutomaticTracking_t, 1145
pbEnabled
unpack_pds_GetXTRAAutomaticDownload_t, 1146
pbIMSRegistered
imsaRegStatusInfo, 315
pbPlatform
unpack_dms_GetOfflineReason_t, 1041
pci
cellParams, 176
ltePCI, 398
nas_cellParams, 426
nas_ltePCI, 462
nas_PhyCaAggPcellInfo, 482
nas_PhyCaAggScellIndType, 483
nas_PhyCaAggScellInfo, 486
nas_umtsLTENbrCell, 520
NASPhyCaAggPcellInfo, 551
NASPhyCaAggScellIndType, 553
NASPhyCaAggScellInfo, 554
PhyCaAggPcellInfo, 738
PhyCaAggScellIndType, 740
PhyCaAggScellInfo, 742
umtsLTENbrCell, 1015
pcsDigit
PlmnID, 744
pcsfFQDNAddress
PCSCFFQDNAddressList, 731
wds_PCSCFFQDNAddressList, 1376
pdpType
pack_wds_SetDefaultProfile_t, 710
pdptype
unpack_wds_GetDefaultProfile_t, 1232
pds.h, 1520
pack_pds_ForceXTRADownload, 1522
pack_pds_GetPDSDefaults, 1523
pack_pds_GetPDSState, 1523
pack_pds_GetPortAutomaticTracking, 1523
pack_pds_GetServiceAutomaticTracking, 1524
pack_pds_GetXTRAAutomaticDownload, 1524
pack_pds_GetXTRANetwork, 1524
pack_pds_GetXTRAValidity, 1525
pack_pds_PDSInjectTimeReference, 1525
pack_pds_ResetPDSData, 1525
pack_pds_SLQSGetAGPSConfig, 1528
pack_pds_SLQSGetGPSStateInfo, 1529
pack_pds_SLQSPDSDeterminePosition, 1529

pack_pds_SLQSPDSInjectAbsoluteTimeReference, 1529
pack_pds_SLQSPDSInjectPositionData, 1530
pack_pds_SLQSSetAGPSConfig, 1530
pack_pds_SLQSSetPositionMethodState, 1531
pack_pds_SetEventReportCallback, 1526
pack_pds_SetPDSDefaults, 1526
pack_pds_SetPDSState, 1527
pack_pds_SetPortAutomaticTracking, 1527
pack_pds_SetServiceAutomaticTracking, 1527
pack_pds_SetXTRAAutomaticDownload, 1528
pack_pds_SetXTRANetwork, 1528
pack_pds_StartPDSTrackingSessionExt, 1531
pack_pds_StopPDSTrackingSession, 1531
unpack_pds_ForceXTRADownload, 1532
unpack_pds_GetPDSDefaults, 1532
unpack_pds_GetPDSState, 1532
unpack_pds_GetPortAutomaticTracking, 1533
unpack_pds_GetServiceAutomaticTracking, 1533
unpack_pds_GetXTRAAutomaticDownload, 1533
unpack_pds_GetXTRANetwork, 1534
unpack_pds_GetXTRAValidity, 1534
unpack_pds_PDSInjectTimeReference, 1535
unpack_pds_ResetPDSData, 1535
unpack_pds_SLQSGetAGPSConfig, 1538
unpack_pds_SLQSGetGPSStateInfo, 1538
unpack_pds_SLQSPDSDeterminePosition, 1539
unpack_pds_SLQSPDSInjectAbsoluteTime-Reference, 1539
unpack_pds_SLQSPDSInjectPositionData, 1539
unpack_pds_SLQSSetAGPSConfig, 1540
unpack_pds_SLQSSetPositionMethodState, 1540
unpack_pds_SetEventReport_Ind, 1535
unpack_pds_SetEventReportCallback, 1536
unpack_pds_SetPDSDefaults, 1536
unpack_pds_SetPDSState, 1536
unpack_pds_SetPdsState_Ind, 1536
unpack_pds_SetPortAutomaticTracking, 1537
unpack_pds_SetServiceAutomaticTracking, 1537
unpack_pds_SetXTRAAutomaticDownload, 1537
unpack_pds_SetXTRANetwork, 1538
unpack_pds_StartPDSTrackingSessionExt, 1540
unpack_pds_StopPDSTrackingSession, 1541
peakRate
tokenBucket, 958
unpack_qos_tokenBucket_t, 1173
peakThroughputClass
GPRSQoS, 285
GPRSRequestedQoS, 286
LibPackGPRSRequestedQoS, 333
wds_GPRSQoS, 1373
peerNumberInfo, 735
callID, 736
numLen, 736
numPI, 736
numPlan, 736
numSI, 736
numType, 736
number, 736
PerformNetworkScan
qaGobiApiNas.h, 1746
PersistentTechPref
unpack_nas_GetNetworkPreference_t, 1097
persoFeature
appStats, 124
appStatus, 127
uim_appStatus, 968
persoRetries
appStats, 124
appStatus, 127
uim_appStatus, 968
persoState
appStats, 124
appStatus, 127
uim_appStatus, 968
persoUnblockRetries
appStats, 124
appStatus, 127
uim_appStatus, 968
personalizationStatus, 736
feature, 737
numFeatures, 737
unblockLeft, 737
verifyLeft, 737
pfailureCause
unpack_voice_SLQSOriginateUSSD_t, 1202
USSResp, 1264
phase
nas_rxInfo, 499
rxInfo, 820
PhyCaAggPcellInfo, 737
dl_bw_value, 738
freq, 738
iLTEbandValue, 738
NasGetLTECphyCaInfo, 533
pci, 738
TlvPresent, 738
PhyCaAggScellDlBw, 738
dl_bw_value, 738
NasGetLTECphyCaInfo, 533
TlvPresent, 738
PhyCaAggScellIndType, 739
freq, 740
NasGetLTECphyCaInfo, 533
pci, 740
scell_state, 740
TlvPresent, 740
PhyCaAggScellIndex, 739
NasGetLTECphyCaInfo, 533
scell_idx, 739
TlvPresent, 739
PhyCaAggScellInfo, 740
dl_bw_value, 742
freq, 742
iLTEbandValue, 742
NasGetLTECphyCaInfo, 533

pci, 742
scell_state, 742
TlvPresent, 742
PhysicalLayer
nas_protocolSubtypeElement, 491
protocolSubtypeElement, 763
PilotEnergy
nas_NetworkStatEVDO, 478
NetworkStatEVDO, 576
PilotPN
nas_PilotSetParams, 487
PilotSetParams, 743
PilotSetData, 742
NumPilots, 742
pPilotSetInfo, 742
PilotSetParams, 743
PilotPN, 743
PilotStrength, 743
PilotType, 743
PilotStrength
nas_PilotSetParams, 487
PilotSetParams, 743
PilotType
nas_PilotSetParams, 487
PilotSetParams, 743
pin1Len
encryptedPIN1, 237
uim_encryptedPIN1, 975
pin1Retries
appStats, 124
appStatus, 127
uim_appStatus, 968
pin1State
appStats, 124
appStatus, 127
uim_appStatus, 968
pin1Val
encryptedPIN1, 237
uim_encryptedPIN1, 975
pin2Retries
appStats, 124
appStatus, 127
uim_appStatus, 968
pin2State
appStats, 124
appStatus, 127
uim_appStatus, 968
pinID
changeUIMPIN, 177
setPINProtection, 855
uim_changeUIMPIN, 973
uim_setPINProtection, 986
uim_unblockUIMPIN, 989
uim_verifyUIMPIN, 990
unblockUIMPIN, 1022
verifyUIMPIN, 1266
pinLen
changeUIMPIN, 177
uim_changeUIMPIN, 973
uim_verifyUIMPIN, 990
verifyUIMPIN, 1266
pinLength
setPINProtection, 855
uim_setPINProtection, 986
pinOperation
setPINProtection, 855
uim_setPINProtection, 986
pinProtection
pack_uim_SetPinProtection_t, 677
pack_uim_UnblockPin_t, 684
UIMSetPinProtectionReq, 1007
pinVal
changeUIMPIN, 177
uim_changeUIMPIN, 973
uim_verifyUIMPIN, 990
verifyUIMPIN, 1266
pinValue
setPINProtection, 855
uim_setPINProtection, 986
PkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 1964
PkQmiNasPerformNetworkScan
qaNasPerformNetworkScan.h, 1965
PkgDescLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
PkgDescription
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
PkgName
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
PkgNameLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
pkgver
CurrentImgList, 198
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1059
PktErrRate
unpack_qos_swiQosFlow_t, 1173
pktErrRate, 743
exponent, 743
multiplier, 743
PktStatElmntsV4
WdsPktStatisticsResp, 1394
PktStatElmntsV6
WdsPktStatisticsResp, 1394
plasmaIDString
pack_dms_SLQSSwiSetHostDevInfo_t, 599
unpack_dms_SLQSSwiGetHostDevInfo_t, 1061
plmn
GERANInfo, 252
LTEInfoIntrafreq, 395
nas_GERANInfo, 440
nas_LTEInfoIntrafreq, 460
nas_UMTSInfo, 519

UMTSInfo, 1013
PlmnID, 744
mcc, 744
mnc, 744
pcsDigit, 744
TlvPresent, 744
polarityIncluded
lineCtrlInfo, 361
Port, 747
port, 747
range, 747
port
IPv4Info, 329
IPv6Info, 332
loc_IPv4Info, 367
loc_IPv6Info, 367
Port, 747
unpack_qos_Port_t, 1157
posDataNmea
pack_pds_SetEventReportCallback_t, 648
posSrc
unpack_pds_SetEventReport_Ind_t, 1149
Position Determination Service (PDS), 43
PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 1149
positionSrc
pack_loc_SLQSLOCInjectPosition_t, 616
pprofileid3gpp
pack_wds_SLQSStartDataSession_t, 725
pprofileid3gpp2
pack_wds_SLQSStartDataSession_t, 725
prDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
prDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
prPCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
prPCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
Precedence
unpack_qos_swiQosFilter_t, 1168
precedenceClass
GPRSQoS, 285
GPRSRequestedQoS, 286
LibPackGPRSRequestedQoS, 333
wds_GPRSQoS, 1373
precisionDilution
qaGobiApiCbk.h, 1563
precisionDilution_s, 747
HDOP, 748
PDOP, 748
VDOP, 748
PrefImageList, 748
listEntries, 748
listSize, 748
prefNetwork
unpack_wds_SLQSGetCurrDataSystemStat_t, 1243
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
prefVoiceSO, 748
evrcCapability, 750
homeOrigVoiceSO, 750
homePageVoiceSO, 750
namID, 750
roamOrigVoiceSO, 750
preference
pack_pds_SetXTRANetwork_t, 651
Preferred
nas_QmiNas3GppNetworkInfo, 493
SlqsNas3GppNetworkInfo, 874
prefixLen
IPv6Addr, 329
unpack_qos_IPv6Addr_t, 1156
presentationInd
ECTNum, 236
remotePartyNum, 811
voice_remotePartyNum, 1300
priChA
CDMAChannel, 162
nas_CDMAChannel, 419
priChB
CDMAChannel, 162
nas_CDMAChannel, 419
priSize
unpack_dms_GetFirmwareRevisions_t, 1037
pridns
unpack_wds_GetDefaultProfile_t, 1232
pridnsv6
unpack_wds_GetDefaultProfile_t, 1232
primaryDNS
pack_wds_SetDefaultProfile_t, 710
PrimaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1250
PrimaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1250
primaryHA
unpack_wds_GetMobileIPProfile_t, 1236
privacyPref
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 701
voiceSetPrefPrivacy, 1346
priver
CurrentImgList, 198
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1059
priversion_str
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
Profile
GetAudioPathConfigReq, 255
GetAudioProfileResp, 258
GetAudioVolTLBConfigReq, 259

GetM2MAudioProfileResp, 275
GetM2MAudioVolumeReq, 276
GetM2MAVMuteReq, 276
GetM2MSpkrGainReq, 278
SetAudioPathConfigReq, 841
SetAudioProfileReq, 842
SetAudioVolTLBConfigReq, 843
SetM2MAudioAVCFGReq, 850
SetM2MAudioProfileReq, 852
SetM2MAudioVolumeReq, 853
SetM2MAVMuteReq, 854
SetM2MSpkrGainReq, 854
Profile3GPP, 750
pAPNClass, 755
pAPNDisabledFlag, 755
pAPNName, 755
pAPNnameSize, 755
pAddrAllocPref, 755
pAuthenticationPref, 755
pGPRSMinimumQoS, 755
pGPRSRequestedQos, 755
pIPv4AddrPref, 755
pIPv6AddPref, 755
pImCnFlag, 755
pPDNInactivTimeout, 755
pPDPtype, 755
pPassword, 755
pPasswordSize, 755
pPcscfAddrUsingDhcp, 755
pPcscfAddrUsingPCO, 755
pPdpAccessConFlag, 755
pPdpContext, 755
pPdpDataCompType, 755
pPdpHdrCompType, 755
pPriDNSIPv4AddPref, 755
pPriDNSIPv6addpref, 755
pPrimaryID, 755
pProfilename, 756
pProfilenameSize, 756
pQosClassID, 756
pSecDNSIPv4AddPref, 756
pSecDNSIPv6addpref, 756
pSecondaryFlag, 756
pTFTID1Params, 756
pTFTID2Params, 756
pUMTSMinQoS, 756
pUMTSMinQosSigInd, 756
pUMTSReqQoS, 756
pUMTSReqQoSSigInd, 756
pUsername, 756
pUsernameSize, 756
Profile3GPP2, 756
pAPNClass3GPP2, 760
pAPNEnabled3GPP2, 760
pAllowLinger, 760
pApnString, 760
pApnStringSize, 760
pAppPriority, 760
pAppType, 760
pAuthPassword, 760
pAuthPasswordSize, 760
pAuthProtocol, 760
pAuthRetryCount, 761
pAuthTimeout, 761
pDataMode, 761
pDataRate, 761
pIpcpAckTimeout, 761
pIpcpCreqRetryCount, 761
pIsPcscfAddressNedded, 761
pLcpAckTimeout, 761
pLcpCreqRetryCount, 761
pNegoDnsSrvrPref, 761
pPDNInactivTimeout3GPP2, 761
pPdnType, 761
pPppSessCloseTimer1x, 761
pPppSessCloseTimerDO, 761
pPriV6DnsAddress, 761
pPrimaryV4DnsAddress, 761
pRATType, 761
pSecV6DnsAddress, 761
pSecondaryV4DnsAddress, 761
pUserId, 761
pUserIdSize, 761
ProfileID
_GetProfileSettingIn, 64
unpack_wds_SLQSGetRuntimeSettings_t, 1250
ProfileId
pack_wds_SLQSGetProfileSettings_t, 718
profileId
wds_DHCPProfileIdTlv, 1369
wds_DHCPv4ProfileId, 1371
WdsDHCPv4ProfileId, 1389
wdsDhcpv4ProfileId, 1389
ProfileId3GPP2
unpack_qos_swiQosFlow_t, 1173
ProfileIdTlv
unpack_wds_DHCPv4ClientLease_ind_t, 1228
ProfileIdentifier, 761
profileIndex, 762
profileType, 762
ProfileIndex
PackCreateProfileOut, 729
profileIndex
pack_wds_SLQSDeleteProfile_t, 715
ProfileIdentifier, 762
SLQSDeleteProfileParams, 872
wds_ProfileIdentifier, 1377
profileList
unpack_wds_SLQSGet3GPPConfigItem_t, 1243
ProfileName
unpack_wds_SLQSGetRuntimeSettings_t, 1250
ProfileType
_GetProfileSettingIn, 64
pack_wds_SLQSGetProfileSettings_t, 718
PackCreateProfileOut, 729
unpack_wds_SLQSGetProfileSettings_t, 1248

profileType
pack_wds_SetDefaultProfile_t, 710
pack_wds_SLQSDeleteProfile_t, 715
ProfileIdentifier, 762
SLQSDeleteProfileParams, 872
wds_DHCPProfileIdTlv, 1369
wds_DHCPv4ProfileId, 1371
wds_ProfileIdentifier, 1377
WdsDHCPv4ProfileId, 1389
wdsDhcpv4ProfileId, 1389
profiletype
pack_wds_GetDefaultProfile_t, 705
Protocol
unpack_nas_GetCDMANetworkParameters_t, 1093
protocolSubtypeElement, 762
AccessMac, 763
AuthProt, 763
ControlMac, 763
EncryptProt, 763
ForwardMac, 763
IdleState, 763
KeyExchange, 763
MultDisc, 763
PhysicalLayer, 763
ReverseMac, 763
SecProt, 763
VirtStream, 763
ProvisionStatus
CLIPResp, 179
CLIRResp, 180
CNAPResp, 182
COLPResp, 183
COLRResp, 183
voice_CLIPResp, 1285
voice_CLIRResp, 1286
voice_CNAPResp, 1286
voice_COLPResp, 1287
voice_COLRResp, 1288
psAttachState
nas_servSystem, 504
NASServingSystemInfo, 562
ServingSystemInfo, 835
servSystem, 837
psBarStatus
CallBarringSysInfo, 143
callBarStatus, 144
nas_CallBarringSysInfo, 417
nas_callBarStatus, 418
psState
CommInfo, 185
nas_CommInfo, 428
psc
nas_UMTSInfo, 519
nas_wcdmaCellInfo, 523
nas_WCDMASysInfo, 528
UMTSInfo, 1014
wcdmaCellInfo, 1354
WCDMASysInfo, 1364
pscValid
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
pscsfIPv4Addr
PCSCFIPv4ServerAddressList, 732
wds_PCSCFIPv4ServerAddressList, 1376
psetting
unpack_wds_GetAutoconnect_t, 1228
psid
pack_wds_SLQSStopDataSession_t, 726
unpack_wds_SLQSStartDataSession_t, 1256
puk1Retries
appStats, 124
appStatus, 127
uim_appStatus, 968
puk2Retries
appStats, 124
appStatus, 127
uim_appStatus, 968
pukLen
uim_unblockUIMPIN, 989
unblockUIMPIN, 1022
pukVal
uim_unblockUIMPIN, 990
unblockUIMPIN, 1022
pukValue
pack_dms_UIMUnblockPIN_t, 604
pv4sessionId
WdsIpAddressInfoReq, 1391
pv6sessionId
WdsIpAddressInfoReq, 1391
pwrDenialTime
lineCtrlInfo, 361

QMI_LITE_WDS_CURRENT_CALL_DB_MASK
wds.h, 2070
QMI_LITE_WDS_LAST_CALL_DB_MASK
wds.h, 2070
QMI_SAR_RF_STATE_1
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_2
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_3
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_4
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_5
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_6
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_7
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_8
qaGobiApiSar.h, 1792
QMI_SAR_RF_STATE_DEFAULT
qaGobiApiSar.h, 1792
QMI_WDS_CURRENT_CALL_DB_MASK
qaGobiApiWds.h, 1927

QMI_WDS_LAST_CALL_DB_MASK
qaGobiApiWds.h, 1927
QCI
LibPackQosClassID, 356
QosClassID, 801
QCWWAN2kConnect
qaGobiApiDcs.h, 1637
QCWWAN2kEnumerateDevices
qaGobiApiDcs.h, 1637
QCWWAN2kGetConnectedDeviceID
qaGobiApiDcs.h, 1638
QCWWANConnect
qaGobiApiDcs.h, 1638
QCWWANDisconnect
qaGobiApiDcs.h, 1639
QCWWANEnumerateDevices
qaGobiApiDcs.h, 1639
QFlowState
unpack_qos_QosFlowInfo_t, 1159
QLIC
DeviceConfigDetail, 222
nas_DeviceConfigDetail, 435
qaCbkCatEventReportInd.h
eTLV_CBK_ALPHA_IDENTIFIER, 1542
eTLV_CBK_DISPLAY_TEXT, 1542
eTLV_CBK_END_PROACTIVE_SESSION, 1542
eTLV_CBK_GET_IN_KEY, 1542
eTLV_CBK_GET_INPUT, 1542
eTLV_CBK_LANGUAGE_NOTIFICATION, 1542
eTLV_CBK_REFRESH, 1542
eTLV_CBK_SELECT_ITEM, 1542
eTLV_CBK_SETUP_EVENT_LIST, 1542
eTLV_CBK_SETUP_IDLE_MODE_TEXT, 1542
eTLV_CBK_SETUP_MENU, 1542
eTLV_END_PROACTIVE_SESSION_LENGTH, 1542
eTLV_REFRESH_LENGTH, 1542
eTLV_SETUP_EVENT_LIST_LENGTH, 1542
qaCbkSwiOmaDmEventReportInd.h
eTLV_IND_OMA_DM_CONFIG, 1543
eTLV_IND_OMA_DM_FOTA, 1543
eTLV_IND_OMA_DM_NOT, 1543
qaGobiApiCbk.h
DEVICE_STATE_BOOT, 1598
DEVICE_STATE_DISCONNECTED, 1598
DEVICE_STATE_READY, 1598
DEVICE_STATE_SERIAL_ONLY, 1598
eQA_QMI_SVC_NA, 1598
eQA_QMI_SVC_NAS, 1598
eQA_QMI_SVC_WDS, 1598
SMS_EVENT_ETWS, 1598
SMS_EVENT_ETWS_PLMN, 1598
SMS_EVENT_MESSAGE_MODE, 1598
SMS_EVENT_MT_MESSAGE, 1598
SMS_EVENT_SMS_ON_IMS, 1598
SMS_EVENT_SMSC_ADDRESS, 1598
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE, 1598
SYS_SRV_DOMAIN_CAMPED, 1598
SYS_SRV_DOMAIN_CS_ONLY, 1598
SYS_SRV_DOMAIN_CS_PS, 1598
SYS_SRV_DOMAIN_NO_SRV, 1598
SYS_SRV_DOMAIN_PS_ONLY, 1598
qaGobiApiFms.h
eGOBI_DEV_SERIES_9X07_9X50, 1687
eGOBI_DEV_SERIES_9X15, 1687
eGOBI_DEV_SERIES_9X30, 1687
eGOBI_DEV_SERIES_G3K, 1687
eGOBI_DEV_SERIES_NON_GOBI, 1687
eGOBI_DEV_SERIES_SIERRA_GOBI, 1687
eGOBI_DEV_SERIES_UNKNOWN, 1687
eGOBI_IMG_CAR_3, 1688
eGOBI_IMG_CAR_AERIS, 1688
eGOBI_IMG_CAR_ALLTEL, 1687
eGOBI_IMG_CAR_AMX_TELCEL, 1688
eGOBI_IMG_CAR_ATT, 1687
eGOBI_IMG_CAR_BELL, 1687
eGOBI_IMG_CAR_BHARTI, 1688
eGOBI_IMG_CAR_BRASIL_VIVO, 1688
eGOBI_IMG_CAR_CHINA_MOBILE, 1688
eGOBI_IMG_CAR_CHINA_TELECOM, 1687
eGOBI_IMG_CAR_CHINA_UNICOM, 1687
eGOBI_IMG_CAR_EMOBILE, 1688
eGOBI_IMG_CAR_FACTORY, 1687
eGOBI_IMG_CAR_GENERIC, 1687
eGOBI_IMG_CAR_GENERIC_CDMA, 1687
eGOBI_IMG_CAR_IUSACELL, 1687
eGOBI_IMG_CAR_KDDI, 1687
eGOBI_IMG_CAR_KT_FREETEL, 1688
eGOBI_IMG_CAR_LEAP, 1687
eGOBI_IMG_CAR_METROPCS, 1687
eGOBI_IMG_CAR_NETCOM, 1688
eGOBI_IMG_CAR_NORF, 1687
eGOBI_IMG_CAR_NTT_DOCOMO, 1688
eGOBI_IMG_CAR_O2, 1688
eGOBI_IMG_CAR_OMH, 1687
eGOBI_IMG_CAR_ORANGE, 1688
eGOBI_IMG_CAR_RELIANCE1, 1687
eGOBI_IMG_CAR_RELIANCE2, 1688
eGOBI_IMG_CAR_ROGERS, 1688
eGOBI_IMG_CAR_SFR, 1688
eGOBI_IMG_CAR_SINGTEL_OPTUS, 1688
eGOBI_IMG_CAR_SK_TELCOM1, 1687
eGOBI_IMG_CAR_SK_TELCOM2, 1688
eGOBI_IMG_CAR_SOFTBANK, 1688
eGOBI_IMG_CAR_SPRINT, 1687
eGOBI_IMG_CAR_SWISSCOM, 1688
eGOBI_IMG_CAR_TATA, 1687
eGOBI_IMG_CAR_TELCOM_ITALIA, 1688
eGOBI_IMG_CAR_TELCOM_NZ, 1687
eGOBI_IMG_CAR_TELEFONICA, 1688
eGOBI_IMG_CAR_TELENOR, 1688
eGOBI_IMG_CAR_TELIASONERA, 1688
eGOBI_IMG_CAR_TELSTRA1, 1687
eGOBI_IMG_CAR_TELSTRA2, 1688
eGOBI_IMG_CAR_TELUS, 1687

eGOBI_IMG_CAR_TMOBILE, 1687
eGOBI_IMG_CAR_US, 1687
eGOBI_IMG_CAR_VERIZON, 1687
eGOBI_IMG_CAR_VODAFONE, 1687
eGOBI_IMG_GPS_ASSISTED, 1688
eGOBI_IMG_GPS_NO_XTRA, 1688
eGOBI_IMG_GPS_NONE, 1688
eGOBI_IMG_GPS_STAND_ALONE, 1688
eGOBI_IMG_REG_ASIA, 1688
eGOBI_IMG_REG_AUS, 1688
eGOBI_IMG_REG_EU, 1688
eGOBI_IMG_REG_GLOBAL, 1688
eGOBI_IMG_REG_LA, 1688
eGOBI_IMG_REG_NA, 1688
eGOBI_IMG_TECH_CDMA, 1689
eGOBI_IMG_TECH_UMTS, 1689
eGobi_DEV_SERIES_MC83, 1687
qaGobiApiNas.h
eNAS_LTE_CPHY_CA_BW_NRB_100, 1736
eNAS_LTE_CPHY_CA_BW_NRB_15, 1736
eNAS_LTE_CPHY_CA_BW_NRB_25, 1736
eNAS_LTE_CPHY_CA_BW_NRB_50, 1736
eNAS_LTE_CPHY_CA_BW_NRB_6, 1736
eNAS_LTE_CPHY_CA_BW_NRB_75, 1736
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1736
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1736
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1736
eNAS_RADIO_IF_GSM, 1735
eNAS_RADIO_IF_LTE, 1735
eNAS_RADIO_IF_TDSCDMA, 1735
eNAS_RADIO_IF_UMTS, 1735
eSYS_SRV_DOMAIN_CAMPED, 1735
eSYS_SRV_DOMAIN_CS_ONLY, 1735
eSYS_SRV_DOMAIN_CS_PS, 1735
eSYS_SRV_DOMAIN_NO_SRV, 1735
eSYS_SRV_DOMAIN_PS_ONLY, 1735
eSYS_SRV_DOMAIN_UNKNOWN, 1735
qaGobiApiPds.h
eSetServiceAutomaticTrackingDisable, 1769
eSetServiceAutomaticTrackingEnable, 1769
qaGobiApiSar.h
QMI_SAR_RF_STATE_1, 1792
QMI_SAR_RF_STATE_2, 1792
QMI_SAR_RF_STATE_3, 1792
QMI_SAR_RF_STATE_4, 1792
QMI_SAR_RF_STATE_5, 1792
QMI_SAR_RF_STATE_6, 1792
QMI_SAR_RF_STATE_7, 1792
QMI_SAR_RF_STATE_8, 1792
QMI_SAR_RF_STATE_DEFAULT, 1792
qaGobiApiVoice.h
VOICE_SUPS_SRV_CLASS_DATA, 1903
VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC, 1903
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC, 1903
VOICE_SUPS_SRV_CLASS_FAX, 1903
VOICE_SUPS_SRV_CLASS_NONE, 1903
VOICE_SUPS_SRV_CLASS_PACKETACCESS, 1903
VOICE_SUPS_SRV_CLASS_PADACCESS, 1903
VOICE_SUPS_SRV_CLASS_SMS, 1903
VOICE_SUPS_SRV_CLASS_VOICE, 1903
qaGobiApiWds.h
QMI_WDS_CURRENT_CALL_DB_MASK, 1927
QMI_WDS_LAST_CALL_DB_MASK, 1927
qaNasGetRFBandInfo.h
eTLV_RF_BAND_INFO, 1963
qaNasPerformNetworkScan.h
eTLV_3GPP_NETWORK_INFO, 1965
qaCbkCatEventReportInd.h, 1541
UpkQmiCbkCatEventReportInd, 1543
qaCbkSwiOmaDmEventReportInd.h, 1543
UpkQmiCbkSwiOmaDmEventReportInd, 1544
UpkQmiCbkSwiOmaDmEventReportIndExt, 1544
qaGobiApiAudio.h, 1544
SLQSGetAudioPathConfig, 1544
SLQSGetAudioProfile, 1545
SLQSGetAudioVolTLBConfig, 1545
SLQSSetAudioPathConfig, 1546
SLQSSetAudioProfile, 1546
SLQSSetAudioVolTLBConfig, 1547
qaGobiApiCat.h, 1547
CATSendEnvelopeCommand, 1548
CATSendTerminalResponse, 1548
qaGobiApiCbk.h, 1549
accelAcceptReady, 1559
accelTempAcceptReady, 1559
CBK_ENABLE_EVENT, 1557
CBK_NOCHANGE, 1557
DEREGISTER_EVENT, 1557
DEREGISTER_SRV, 1557
device_state_enum, 1598
eDevState, 1560
eQaQMIService, 1598
eSMSEventType, 1560
EVENT_MASK_CARD, 1557
FIRST_INSTANCE, 1557
gpsTime, 1560
gyroAcceptReady, 1560
gyroTempAcceptReady, 1561
INVALID_INSTACNE, 1558
IPV4, 1558
IPV4V6, 1558
IPV6, 1558
iSLQSSetDUNCallInfoCallback, 1599
iSLQSSetSignalStrengthsCallback, 1599
iSLQSSetWdsFirstInstEventCallback, 1599
iSLQSSetWdsSecondInstEventCallback, 1599
iSLQSSetWdsThirdInstEventCallback, 1599
iSLQSSetWdsXferStatsFirstInstCallback, 1599
iSLQSSetWdsXferStatsSecondInstCallback, 1599

iSetCATEventCallback, 1599
iSetSignalStrengthCallback, 1599
LteNasReleaseInfo, 1561
MAX_NO_OF_CALLS, 1558
MAX_NO_OF_FILES, 1558
MAX_NO_OF_SLOTS, 1558
MAX_PATH_LENGTH, 1558
MAXUSSDLENGTH, 1558
modemTempNotification, 1562
NAS_SRV, 1558
NUM_OF_SET, 1558
PDS_SRV, 1558
packetSrvStatus, 1562
precisionDilution, 1563
REGISTER_EVENT, 1559
REGISTER_SRV, 1559
ResetInfoNotification, 1564
SECOND_INSTANCE, 1559
SLQSNasNetworkRejectCallback, 1614
SLQSNasNetworkTimeCallBack, 1614
SLQSNasSigInfo2CallBack, 1614
SLQSNasSigInfoCallBack, 1615
SLQSNasSwiOTAMessageCallback, 1615
SLQSNasSysInfoCallBack, 1616
SLQSNasTimerCallback, 1616
SLQSSetBandPreferenceCbk, 1616
SLQSSetDHCPv4ClientLeaseStatusCallback, 1617
SLQSSetDUNCallInfoCallback, 1618
SLQSSetDataSystemStatusCallback, 1617
SLQSSetIMSAPdpStatusCallback, 1618
SLQSSetIMSARatStatusCallback, 1618
SLQSSetIMSARegStatusCallback, 1619
SLQSSetIMSASvcStatusCallback, 1619
SLQSSetIMSSMSConfigCallback, 1620
SLQSSetIMSUserConfigCallback, 1620
SLQSSetIMSVoIPConfigCallback, 1621
SLQSSetLocInjectPositionCallback, 1621
SLQSSetLocInjectUTCTimeCallback, 1621
SLQSSetModemTempCallback, 1621
SLQSSetPacketSrvStatusCallback, 1622
SLQSSetQosEventCallback, 1622
SLQSSetQosNWStatusCallback, 1623
SLQSSetQosPriEventCallback, 1623
SLQSSetQosStatusCallback, 1623
SLQSSetRegMgrConfigCallback, 1624
SLQSSetSDKTerminatedCallback, 1624
SLQSSetSIPConfigCallback, 1626
SLQSSetSMSEventCallback, 1627
SLQSSetServingSystemCallback, 1625
SLQSSetSessionStateCallback, 1625
SLQSSetSignalStrengthsCallback, 1626
SLQSSetSwiGetResetInfoCallback, 1627
SLQSSetSwiHDRPersCallback, 1627
SLQSSetSysSelectionPrefCallBack, 1628
SLQSSetTransLayerInfoCallback, 1628
SLQSSetTransNWRegInfoCallback, 1628
SLQSSetWdsEventCallback, 1629
SLQSSetWdsTransferStatisticCallback, 1630
SLQSSwiAVMSEventReportCallback, 1630
SLQSTmdMitigationLvlRptCallback, 1630
SLQSUIMSetRefreshCallBack, 1631
SLQSUIMSetStatusChangeCallBack, 1631
SLQSVoiceInfoRecCallback, 1631
SLQSVoiceSetAllCallStatusCallBack, 1632
SLQSVoiceSetDTMFEventCallBack, 1632
SLQSVoiceSetOTASPStatusCallBack, 1633
SLQSVoiceSetPrivacyChangeCallBack, 1633
SLQSVoiceSetSUPSCallBack, 1634
SLQSVoiceSetSUPSNotificationCallback, 1634
SLQSWmsAsyncRawSendCallBack, 1634
SLQSWmsMemoryFullCallBack, 1635
SLQSWmsMessageWaitingCallBack, 1635
SMSAsyncRawSend, 1565
SMSCAddressInfo, 1566
SMSEtwsMessageInfo, 1566
SMSEtwsPlmnInfo, 1567
SMSEventInfo, 1567
SMSEventType, 1598
SMSMTMessageInfo, 1568
SMSMessageModeInfo, 1568
SMSOnIMSInfo, 1568
SMSTransferRouteMTMessageInfo, 1568
sensorDataUsage, 1564
sessionInformation, 1565
sessionInformationExt, 1565
SetActivationStatusCallback, 1599
SetCATEventCallback, 1599
SetDataCapabilitiesCallback, 1600
SetDeviceStateChangeCbk, 1601
SetFwDldCompletionCbk, 1601
SetGPSCallback, 1602
SetLURejectCallback, 1606
SetLocBestAvailPosCallback, 1602
SetLocCradleMountCallback, 1602
SetLocDeleteAssistDataCallback, 1603
SetLocEngineStateCallback, 1603
SetLocEventMaskNMEACallback, 1603
SetLocEventPositionCallback, 1603
SetLocEventTimeSyncCallback, 1604
SetLocGetServerCallback, 1604
SetLocGnssSvInfoCallback, 1604
SetLocInjectSensorDataCallback, 1605
SetLocInjectTimeCallback, 1605
SetLocOpModeCallback, 1605
SetLocSensorStreamingCallback, 1605
SetLocSetExtPowerConfigCallback, 1606
SetMobileIPStatusCallback, 1606
SetNMEACallback, 1608
SetNasLTECphyCaIndCallback, 1607
SetNetChangeCbk, 1607
SetNewSMSCallback, 1608
SetOMADMStateCallback, 1609
SetPDSStateCallback, 1609
SetPowerCallback, 1609
SetRFInfoCallback, 1610

SetRMTransferStatisticsCallback, 1610
SetRankIndicatorCallback, 1610
SetRoamingIndicatorCallback, 1610
SetSLQSOMADMAlertCallback, 1612
SetSLQSOMADMAlertCallbackExt, 1612
SetSignalStrengthCallback, 1611
SetUSSDNoWaitIndicationCallback, 1613
SetUSSDNotificationCallback, 1613
SetUSSDReleaseCallback, 1613
SetUimSlotStatusChangeCallback, 1612
svUsedforFix, 1569
SwiAvmsEventInd, 1569
SwiOTAMsg, 1569
tFNASwiLTECphyCaInfo, 1570
tFNASwiOTAMsg, 1571
tFNActivationStatus, 1570
tFNAllCallStatus, 1570
tFNAsyncRawSend, 1571
tFNBandPreference, 1571
tFNBestAvailPos, 1572
tFNCATEvent, 1573
tFNCbkUimSlotStatusChangeInd, 1573
tFNDHCPv4ClientLeaseStatus, 1575
tFNDTMFEvent, 1575
tFNDUNCallInfo, 1575
tFNDataCapabilities, 1573
tFNDataSysStatus, 1574
tFNDelAssistData, 1574
tFNDeviceStateChange, 1574
tFNEventNmea, 1575
tFNEventPosition, 1575
tFNFwDldCompletion, 1576
tFNGetServer, 1576
tFNGnssSvInfo, 1576
tFNHDRPersonaity, 1576
tFNImsRegMgrConfig, 1577
tFNImsSIPConfig, 1578
tFNImsSMSConfig, 1578
tFNImsUserConfig, 1578
tFNImsVoIPConfig, 1578
tFNImsaPdpStatus, 1577
tFNImsaRatStatus, 1577
tFNImsaRegStatus, 1577
tFNImsaSvcStatus, 1577
tFNInfoRec, 1578
tFNInjectPosition, 1579
tFNInjectSensorData, 1579
tFNInjectTimeStatus, 1579
tFNInjectUTCTime, 1579
tFNLUReject, 1580
tFNMemoryFull, 1582
tFNMessageWaiting, 1582
tFNMitiLvlRpt, 1582
tFNMobileIPStatus, 1582
tFNModemTempInfo, 1582
tFNNasTimer, 1583
tFNNet, 1583
tFNNetworkReject, 1583
tFNNetworkTime, 1583
tFNNewGPS, 1584
tFNNewNMEA, 1584
tFNNewRMTransferStatistics, 1584
tFNNewSMS, 1585
tFNOMADMState, 1585
tFNOTASPStatus, 1587
tFNOpMode, 1586
tFNPDSState, 1587
tFNPacketSrvState, 1587
tFNPower, 1588
tFNPrivacyChange, 1588
tFNQosNWStatus, 1588
tFNQosPriEvent, 1589
tFNQosStatus, 1589
tFNRFInfo, 1590
tFNRankIndicator, 1590
tFNResetInfo, 1590
tFNRoamingIndicator, 1591
tFNSDKTerminated, 1591
tFNSLQSOMADMAlert, 1593
tFNSLQSQOSEvent, 1593
tFNSLQSSessionState, 1593
tFNSLQSSignalStrengths, 1594
tFNSLQSWDSEvent, 1594
tFNSMSEvents, 1594
tFNSUPSInfo, 1594
tFNSUPSNotification, 1594
tFNSensorStreaming, 1591
tFNServingSystem, 1591
tFNSetCradleMount, 1592
tFNSetEngineState, 1592
tFNSetEventTimeSync, 1592
tFNSetExtPowerConfig, 1592
tFNSigInfo, 1592
tFNSignalStrength, 1593
tFNSwiAvmEventReport, 1595
tFNSysInfo, 1595
tFNSysSelectionPref, 1595
tFNUIMRefresh, 1596
tFNUIMStatusChangeInfo, 1596
tFNUSSDNoWaitIndication, 1596
tFNUSSDNotification, 1596
tFNUSSDRelease, 1597
tFNtransLayerInfo, 1595
tFNtransNWRegInfo, 1595
THIRD_INSTANCE, 1559
transLayerNotification, 1597
transNWRegInfoNotification, 1597
USSD_DCS_8BIT, 1559
USSD_DCS_ASCII, 1559
USSD_DCS_UCS2, 1559
VOICE_SRV, 1559
WDS_SRV, 1559
qaGobiApiDcs.h, 1636
LEN, 1637
PORTNAM_LEN, 1637
QCWWAN2kConnect, 1637

QCWWAN2kEnumerateDevices, 1637
QCWWAN2kGetConnectedDeviceID, 1638
QCWWANConnect, 1638
QCWWANDisconnect, 1639
QCWWANEnumerateDevices, 1639
SLQSGetDeviceMode, 1640
SLQSGetNetStatistic, 1640
SLQSGetUsbPortNames, 1641
SLQSKillSDKProcess, 1641
SLQSSetLoggingMask, 1642
SLQSStart, 1642
SLQSStart_AVAgent, 1643
SLQSStartSrv, 1643
SetSDKImagePath, 1640
qaGobiApiDms.h, 1644
ActivateAutomatic, 1653
custFeaturesInfo, 1646
custFeaturesSetting, 1648
dmsCurrentPRLInfo, 1649
ERIFileparams, 1650
GetActivationState, 1654
GetDeviceCapabilities, 1654
GetFirmwareRevision, 1655
GetFirmwareRevisions, 1656
GetHardwareRevision, 1656
GetIMSI, 1657
GetManufacturer, 1657
GetModelID, 1658
GetNetworkTime, 1658
GetOfflineReason, 1659
GetPRLVersion, 1660
GetPower, 1660
GetSerialNumbers, 1660
GetVoiceNumber, 1661
MAX_CUST_ID_LEN, 1646
MAX_FSN_LENGTH, 1646
ResetToFactoryDefaults, 1662
SLQSDmsSwiGetResetInfo, 1663
SLQSDmsSwiIndicationRegister, 1663
SLQSGetBandCapabilities, 1663
SLQSGetBandCapability, 1663
SLQSGetCurrentPRLInfo, 1665
SLQSGetCustFeatures, 1665
SLQSGetCustFeaturesV2, 1666
SLQSGetERIFile, 1666
SLQSGetSerialNumbers, 1666
SLQSSetCustFeatures, 1667
SLQSSetCustFeaturesV2, 1667
SLQSSwiClearDyingGaspStatistics, 1668
SLQSSwiGetCrashAction, 1668
SLQSSwiGetCrashInfo, 1668
SLQSSwiGetDyingGaspCfg, 1669
SLQSSwiGetDyingGaspStatistics, 1669
SLQSSwiGetFSN, 1669
SLQSSwiGetFwUpdateStatus, 1670
SLQSSwiGetHostDevInfo, 1670
SLQSSwiGetHostDevInfoParams, 1651
SLQSSwiGetOSInfo, 1671
SLQSSwiGetOSInfoParams, 1651
SLQSSwiGetSerialNoExt, 1671
SLQSSwiGetSerialNoExtParams, 1652
SLQSSwiGetUSBComp, 1671
SLQSSwiSetCrashAction, 1672
SLQSSwiSetDyingGaspCfg, 1672
SLQSSwiSetHostDevInfo, 1673
SLQSSwiSetHostDevInfoParams, 1652
SLQSSwiSetOSInfo, 1673
SLQSSwiSetOSInfoParams, 1653
SLQSSwiSetUSBComp, 1673
SLQSUIMGetState, 1674
serialNumbersInfo, 1650
SetPower, 1662
UIMChangePIN, 1674
UIMGetControlKeyStatus, 1675
UIMGetICCID, 1676
UIMGetPINStatus, 1677
UIMSetControlKeyProtection, 1678
UIMSetPINProtection, 1678
UIMUnblockControlKey, 1679
UIMUnblockPIN, 1680
UIMVerifyPIN, 1681
ValidateSPC, 1681
qaGobiApiFms.h, 1682
BUILD_ID_LEN, 1686
DEVICE_OFFLINE, 1686
DEVICE_RESET, 1686
DEVICE_SHUTDOWN, 1686
DeleteStoredImage, 1689
DownloadToSlot, 1689
eGetDeviceSeries, 1690
eGobiDeviceSeries, 1687
eGobiImageCarrier, 1687
eGobiImageGPS, 1688
eGobiImageRegion, 1688
eGobiImageTech, 1688
GetImageStore, 1691
GetImagesPreference, 1690
GetStoredImages, 1691
IMG_ID_LEN, 1686
IMGDETAILS_LEN, 1686
PRI_UPDATE_FAIL, 1686
SLQSDownloadFirmwareToSlot, 1692
SLQSGetBootVersionNumber, 1693
SLQSGetFirmwareInfo, 1694
SLQSGetImageInfo, 1694
SLQSGetImageInfo_9x15, 1695
SLQSGetImageInfoMC77xx, 1695
SLQSGetImageInfoMC83xx, 1696
SLQSGetValidFwPriCombinations, 1696
SLQSIsSpkgFormatRequired, 1697
SLQSSetCrashStateCheckIgnore, 1697
SLQSSetSIMBasedImageSwitching, 1698
SLQSSetSpkgFormatRequired, 1698
SLQSSwiGetAllCarrierImages, 1698
SLQSSwiGetFirmwareCurr, 1699
SLQSUpgradeFirmware9x15, 1699

SetImagesPreference, 1691
UNIQUE_ID_LEN, 1686
upgrade_mc77xx_fw, 1700
UpgradeFirmware2k, 1700
qaGobiApiIms.h, 1701
SLQSGetIMSSMSConfig, 1702
SLQSGetIMSUserConfig, 1702
SLQSGetIMSVoIPConfig, 1703
SLQSGetRegMgrConfig, 1703
SLQSGetSIPConfig, 1704
SLQSImsConfigIndicationRegister, 1704
SLQSSetIMSSMSConfig, 1705
SLQSSetIMSUserConfig, 1705
SLQSSetIMSVoIPConfig, 1706
SLQSSetRegMgrConfig, 1706
SLQSSetSIPConfig, 1707
qaGobiApiImsa.h, 1707
SLQSGetIMSARegStatus, 1708
SLQSGetIMSAServiceStatus, 1708
SLQSGetIMSASupportedFields, 1709
SLQSGetIMSASupportedMsg, 1709
SLQSRegisterIMSAIndication, 1710
qaGobiApiLoc.h, 1710
SLQSLOCDelAssData, 1711
SLQSLOCEventRegister, 1712
SLQSLOCGetBestAvailPos, 1712
SLQSLOCGetServer, 1713
SLQSLOCInjectPosition, 1713
SLQSLOCInjectSensorData, 1714
SLQSLOCInjectUTCTime, 1714
SLQSLOCSetCradleMountConfig, 1714
SLQSLOCSetExtPowerState, 1715
SLQSLOCSetOpMode, 1715
SLQSLOCStart, 1716
SLQSLOCStop, 1716
SwiLocGetAutoStart, 1717
SwiLocSetAutoStart, 1717
qaGobiApiNas.h, 1718
eSYS_SRV_DOMAIN, 1735
GetACCOLC, 1736
GetANAAAAuthenticationStatus, 1736
GetCDMANetworkParameters, 1737
GetHomeNetwork, 1738
GetHomeNetwork3GPP2, 1739
GetNetworkPreference, 1741
GetRFInfo, 1742
GetServingNetwork, 1742
GetServingNetworkCapabilities, 1743
GetSignalStrengths, 1744
IMSI_M_S1_LENGTH, 1723
IMSI_M_S2_LENGTH, 1723
InitiateDomainAttach, 1745
InitiateNetworkRegistration, 1745
MAX_PILOT_SETS, 1723
NAM_NAME_LENGTH, 1723
PLMN_LENGTH, 1723
PerformNetworkScan, 1746
SLQSConfigSigInfo, 1749
SLQSGetErrorRate, 1750
SLQSGetNetworkTime, 1750
SLQSGetOperatorNameData, 1750
SLQSGetPLMNName, 1751
SLQSGetServingSystem, 1751
SLQSGetSignalStrength, 1752
SLQSGetSysSelectionPref, 1752
SLQSInitiateNetworkRegistration, 1753
SLQSNASGetLTECPHYCaInfo, 1755
SLQSNASSwiGetChannelLock, 1758
SLQSNASSwiSetChannelLock, 1760
SLQSNasConfigSigInfo2, 1753
SLQSNasGet3GPP2Subscription, 1754
SLQSNasGetCellLocationInfo, 1754
SLQSNasGetHDRColorCode, 1755
SLQSNasGetSigInfo, 1755
SLQSNasGetSysInfo, 1756
SLQSNasGetTxRxInfo, 1756
SLQSNasIndicationRegister, 1757
SLQSNasIndicationRegisterExt, 1758
SLQSNasIndicationRegisterLTECphyCa, 1758
SLQSNasSwiIndicationRegister, 1759
SLQSNasSwiModemStatus, 1759
SLQSPerformNetworkScan, 1760
SLQSSetBandPreference, 1760
SLQSSetSysSelectionPref, 1762
SLQSSwiGetHDRPersonality, 1762
SLQSSwiGetHDRProtSubtype, 1763
SLQSSwiGetHRPDStats, 1763
SLQSSwiGetLteCQI, 1763
SLQSSwiGetLteSccRxInfo, 1764
SLQSSwiNetworkDebug, 1764
SLQSSwiPSDetach, 1765
SetACCOLC, 1746
SetCDMANetworkParameters, 1747
SetNetworkPreference, 1748
SlqsNas3GppNetworkRAT, 1723
slqsNetworkScanInfo, 1724
sysSelectPrefInfo, 1724
sysSelectPrefParams, 1729
UATISIZE, 1723
qaGobiApiOmadm.h, 1765
OMADMCancelSession, 1765
OMADMGetPendingNIA, 1766
OMADMGetSessionInfo, 1766
OMADMStartSession, 1768
qaGobiApiPds.h, 1768
DEFAULTBYTEVALUE, 1769
DEFAULTLONGVALUE, 1769
DEFAULTWORDVALUE, 1769
ForceXTRADownload, 1770
GetPDSDefaults, 1770
GetPDSState, 1771
GetPortAutomaticTracking, 1771
GetServiceAutomaticTracking, 1772
GetXTRAAutomaticDownload, 1772
GetXTRANetwork, 1773
GetXTRAValidity, 1773

PDSInjectTimeReference, 1774
ResetPDSData, 1774
SLQSGetAGPSConfig, 1778
SLQSGetGPSStateInfo, 1779
SLQSPDSDeterminePosition, 1779
SLQSPDSInjectAbsoluteTimeReference, 1780
SLQSPDSInjectPositionData, 1780
SLQSSetAGPSConfig, 1781
SLQSSetPositionMethodState, 1781
SetPDSDefaults, 1775
SetPDSState, 1776
SetPortAutomaticTracking, 1776
SetServiceAutomaticTracking, 1777
SetXTRAAutomaticDownload, 1777
SetXTRANetwork, 1778
StartPDSTrackingSessionExt, 1782
StopPDSTrackingSession, 1783
qaGobiApiQos.h, 1783
SLQSQosGetFlowStatus, 1784
SLQSQosGetGranted, 1784
SLQSQosGetNWProf, 1785
SLQSQosGetNetworkStatus, 1785
SLQSQosModify, 1786
SLQSQosRel, 1786
SLQSQosReq, 1787
SLQSQosReset, 1787
SLQSQosResume, 1788
SLQSQosSuspend, 1788
SLQSQosSwiReadApnExtraParams, 1788
SLQSQosSwiReadDataStats, 1789
qaGobiApiRms.h, 1789
GetSMSWake, 1790
SetSMSWake, 1790
qaGobiApiSar.h, 1791
eQMISARRFState, 1791
SLQSGetRfSarState, 1792
SLQSSetRfSarState, 1793
qaGobiApiSms.h, 1793
ABSOLUTE_VALIDITY, 1795
CONFIG_LEN, 1795
getIndicationRegResp, 1795
GetSMSCAddress, 1799
getTransLayerInfoResp, 1796
getTransNWRegInfoResp, 1796
MAX_SMS_ROUTES, 1795
NUM_OF_SET, 1795
qaQmi3GPP2BroadcastCfgInfo, 1797
qaQmi3GPPBroadcastCfgInfo, 1797
SLQSCDMADecodeMTTextMsg, 1801
SLQSCDMAEncodeMOTextMsg, 1802
SLQSDeleteSMS, 1802
SLQSGetIndicationRegister, 1803
SLQSGetMessageWaiting, 1804
SLQSGetSMS, 1804
SLQSGetSMSList, 1806
SLQSGetSmsBroadcastConfig, 1805
SLQSGetTransLayerInfo, 1807
SLQSGetTransNWRegInfo, 1807
SLQSModifySMSStatus, 1807
SLQSSendAsyncSMS, 1808
SLQSSendLongSMS, 1808
SLQSSendSMS, 1809
SLQSSetIndicationRegister, 1810
SLQSSetSmsBroadcastActivation, 1810
SLQSSetSmsBroadcastConfig, 1811
SLQSSetSmsStorage, 1811
SLQSSmsGetMaxStorageSize, 1812
SLQSSmsGetMessageProtocol, 1812
SLQSSmsSetRoutes, 1813
SLQSSwiGetSMSStorage, 1813
SLQSWCDMADecodeLongTextMsg, 1814
SLQSWCDMADecodeMTTextMsg, 1814
SLQSWCDMAEncodeMOTextMsg, 1814
SaveSMS, 1799
SendSMS, 1800
setIndicationRegReq, 1798
SetSMSCAddress, 1801
TIME_DATE_BUF, 1795
TIME_STAMP_BUF, 1795
transLayerInfo, 1798
qaGobiApiSwi.h, 1815
SLQSGetPidof, 1815
SLQSGetSdkVersion, 1816
SLQSSendRawQMI, 1816
qaGobiApiSwiAudio.h, 1816
SLQSGetM2MAVMute, 1818
SLQSGetM2MAudioProfile, 1817
SLQSGetM2MAudioVolume, 1818
SLQSGetM2MSpkrGain, 1818
SLQSSetM2MAVMute, 1821
SLQSSetM2MAudioAVCFG, 1819
SLQSSetM2MAudioLPBK, 1819
SLQSSetM2MAudioNVDef, 1820
SLQSSetM2MAudioProfile, 1820
SLQSSetM2MAudioVolume, 1820
SLQSSetM2MSpkrGain, 1821
qaGobiApiSwiAvms.h, 1822
QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 1824
QmiSwiAvmsEventReportConfigResp, 1825
QmiSwiAvmsEventReportConnectionRequestResp, 1825
QmiSwiAvmsEventReportDataSessionStatusResp, 1826
QmiSwiAvmsEventReportHTTPStatusResp, 1826
QmiSwiAvmsEventReportNotificationResp, 1827
QmiSwiAvmsEventReportPackageIDResp, 1828
QmiSwiAvmsEventReportRegStatusResp, 1828
QmiSwiAvmsEventReportSessionTypeResp, 1828
QmiSwiAvmsEventReportWAMSParamChangeResp, 1829
QmiSwiAvmsSetSettingsAPNInfo, 1829
QmiSwiAvmsSetSettingsConnectionRetryTimers, 1829
QmiSwiAvmsSetSettingsPeriodInfo, 1829
SLQSAVMSGetSessionInfo, 1831

SLQSAVMSGetSettings, 1831
SLQSAVMSSelection, 1831
SLQSAVMSSetEventReport, 1832
SLQSAVMSSetSettings, 1832
SLQSAVMSSetSettingsIgnoreAutoReboot, 1833
SLQSAVMSSetSettingsReq, 1829
SLQSAVMSSettings, 1830
SLQSAVMSSettingsAPNInfo, 1830
SLQSAVMSSettingsConnectionRetryTimers, 1830
SLQSAVMSSettingsPeriodsInfo, 1830
SLQSAVMSStartSession, 1833
SLQSAVMSStopSession, 1834
qaGobiApiSwiOmadms.h, 1834
SLQSOMADMCancelSession, 1840
SLQSOMADMGetSessionInfo, 1840
SLQSOMADMGetSettings, 1841
SLQSOMADMGetSettings2, 1841
SLQSOMADMSendSelection, 1842
SLQSOMADMSendSelection2, 1842
SLQSOMADMSessionInfo, 1835
SLQSOMADMSetSettings, 1843
SLQSOMADMSetSettings2, 1844
SLQSOMADMSetSettings3, 1844
SLQSOMADMSettings, 1837
SLQSOMADMSettingsReqParams, 1838
SLQSOMADMSettingsReqParams3, 1839
SLQSOMADMStartSession, 1844
SLQSOMADMStartSession2, 1845
qaGobiApiTableBandClasses.h, 1845
qaGobiApiTableCallControlReturnReasons.h, 1848
qaGobiApiTableCallEndReasons.h, 1849
qaGobiApiTableCarrierCodes.h, 1865
qaGobiApiTableCodingScheme.h, 1867
qaGobiApiTableGpsCapabilityCodes.h, 1870
qaGobiApiTablePowerModes.h, 1871
qaGobiApiTableRadioInterfaces.h, 1871
qaGobiApiTableRegionCodes.h, 1872
qaGobiApiTableServiceOptions.h, 1872
qaGobiApiTableSupServiceInfoClasses.h, 1875
qaGobiApiTableSwiAudio.h, 1875
qaGobiApiTableSwiOMADMSessionStatus.h, 1876
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 1877
qaGobiApiTableVoiceCallEndReasons.h, 1878
qaGobiApiTmd.h, 1885
SLQSTmdDeRegNotMitigationLvl, 1885
SLQSTmdGetMitigationDevList, 1886
SLQSTmdGetMitigationLvl, 1886
SLQSTmdRegNotMitigationLvl, 1886
qaGobiApiUim.h, 1887
MAX_ICCID_LENGTH, 1889
MAX_NO_OF_SLOTS, 1889
MAX_PATH_LENGTH, 1889
MAX_PUK_LENGTH, 1889
MAX_SLOTS_STATUS, 1889
SLQSUIMAuthenticate, 1889
SLQSUIMChangePin, 1890
SLQSUIMDepersonalization, 1890
SLQSUIMEventRegister, 1891
SLQSUIMGetCardStatus, 1891
SLQSUIMGetConfiguration, 1892
SLQSUIMGetFileAttributes, 1892
SLQSUIMGetSlotsStatus, 1893
SLQSUIMPowerDown, 1893
SLQSUIMPowerUp, 1894
SLQSUIMReadTransparent, 1894
SLQSUIMRefreshComplete, 1895
SLQSUIMRefreshGetLastEvent, 1895
SLQSUIMRefreshOK, 1896
SLQSUIMRefreshRegister, 1896
SLQSUIMReset, 1897
SLQSUIMSetPinProtection, 1897
SLQSUIMSwitchSlot, 1898
SLQSUIMUnblockPin, 1898
SLQSUIMVerifyPin, 1899
qaGobiApiVoice.h, 1899
AnswerUSSD, 1903
CancelUSSD, 1903
MAX_CALL_NO_LEN, 1902
MAX_NO_OF_CALLS, 1902
MAXUSSDLENGTH, 1902
OriginateUSSD, 1904
PASSWORD_LENGTH, 1903
SLQSOriginateUSSD, 1904
SLQSVoiceALSSelectLine, 1905
SLQSVoiceALSSetLineSwitching, 1905
SLQSVoiceAnswerCall, 1906
SLQSVoiceBindSubscription, 1906
SLQSVoiceBurstDTMF, 1907
SLQSVoiceDialCall, 1907
SLQSVoiceEndCall, 1908
SLQSVoiceGetAllCallInfo, 1908
SLQSVoiceGetCLIP, 1911
SLQSVoiceGetCLIR, 1912
SLQSVoiceGetCNAP, 1912
SLQSVoiceGetCOLP, 1913
SLQSVoiceGetCOLR, 1913
SLQSVoiceGetCallBarring, 1909
SLQSVoiceGetCallForwardingStatus, 1909
SLQSVoiceGetCallInfo, 1910
SLQSVoiceGetCallWaiting, 1910
SLQSVoiceGetConfig, 1914
SLQSVoiceIndicationRegister, 1914
SLQSVoiceManageCalls, 1915
SLQSVoiceOrigUSSDNoWait, 1915
SLQSVoiceSendFlash, 1916
SLQSVoiceSetCallBarringPassword, 1916
SLQSVoiceSetConfig, 1917
SLQSVoiceSetPreferredPrivacy, 1917
SLQSVoiceSetSUPSService, 1918
SLQSVoiceStartContDTMF, 1919
SLQSVoiceStopContDTMF, 1919
serviceClassInformation, 1903
qaGobiApiWds.h, 1920
GetAutoconnect, 1927
GetByteTotals, 1927

GetConnectionRate, 1928
GetDataBearerTechnology, 1929
GetDefaultProfile, 1929
GetDefaultProfileLTE, 1931
GetDefaultProfileNum, 1933
GetDormancyState, 1933
GetIPAddressLTE, 1934
GetLastMobileIPError, 1934
GetMobileIP, 1935
GetMobileIPProfile, 1935
GetPacketStatistics, 1937
GetPacketStatus, 1937
GetProfileSettingIn, 1923
GetProfileSettingOut, 1924
GetSessionDuration, 1938
GetSessionState, 1938
iGetByteTotals, 1939
iGetConnectionRate, 1939
iGetPacketStatistics, 1939
iSLQSMISetIPFamilyPreference, 1939
qmiDataBearerMasks, 1927
QmiProfileInfo, 1924
QmiWDSDataBearerTechnology, 1925
QmiWDSDataBearers, 1924
RMSetTransferStatistics, 1939
SLQSAutoConnect, 1948
SLQSCreateProfile, 1948
SLQSDeleteProfile, 1949
SLQSGet3GPPConfigItem, 1950
SLQSGetByteTotals, 1950
SLQSGetConnectionRate, 1950
SLQSGetCurrDataSystemStat, 1951
SLQSGetCurrentChannelRate, 1951
SLQSGetDUNCallInfo, 1953
SLQSGetDataBearerTechnology, 1952
SLQSGetDataBearerTechnologyExt, 1952
SLQSGetPacketStatistics, 1953
SLQSGetProfile, 1954
SLQSGetProfileSettings, 1955
SLQSGetRuntimeSettings, 1956
SLQSGetSessionState, 1956
SLQSModifyProfile, 1957
SLQSResetPacketStatics, 1957
SLQSSGetDHCPv4ClientConfig, 1959
SLQSSGetLoopback, 1960
SLQSSSetDHCPv4ClientConfig, 1960
SLQSSSetLoopback, 1960
SLQSSet3GPPConfigItem, 1957
SLQSSetProfile, 1958
SLQSStartStopDataSession, 1961
SLQSWdsGoActive, 1961
SLQSWdsGoDormant, 1962
SLQSWdsSetEventReport, 1962
SLQSWdsSwiPDPRuntimeSettings, 1962
SetActiveMobileIPProfile, 1940
SetAutoconnect, 1940
SetDefaultProfile, 1940
SetDefaultProfileLTE, 1942
SetDefaultProfileLTEV2, 1943
SetDefaultProfileNum, 1945
SetMobileIP, 1945
SetMobileIPParameters, 1946
SetMobileIPProfile, 1947
slqs3GPPConfigItem, 1925
WDS_IsGobiDevice, 1963
qaNasGetRFBandInfo.h, 1963
PkQmiNasGetRFBandInfo, 1964
UpkQmiNasGetRFBandInfo, 1964
qaNasPerformNetworkScan.h, 1964
FORBIDDEN_INDEX, 1964
INDEX_ZERO, 1964
PREFERRED_INDEX, 1964
PkQmiNasPerformNetworkScan, 1965
ROAMING_INDEX, 1964
UpkQmiNasPerformNetworkScan, 1965
qaQmi3GPP2BroadcastCfgInfo
qaGobiApiSms.h, 1797
qaQmi3GPPBroadcastCfgInfo
qaGobiApiSms.h, 1797
qaQmi3Gpp2TimeZone, 764
daylightSavings, 764
leapSeconds, 764
localTimeOffset, 764
qaQmiInterfaceInfo, 765
qaQmiinstanceid, 765
qaQmisvctype, 765
v4sessionId, 765
v6sessionId, 765
qaQmiServingSystemParam, 765
BasestationID, 768
BasestationLatitude, 768
BasestationLongitude, 768
CDMA_P_Rev, 768
CDMASystemInfoExt, 768
CallBarStatus, 768
CellID, 768
concSvcInfo, 768
CurrentPLMN, 768
DTMInd, 769
DataSrvCapabilities, 769
defaultRoamInd, 769
DetailedSvcInfo, 769
Gpp2TimeZone, 769
GppNetworkDSTAdjustment, 769
GppTimeZone, 769
hdrPersonality, 769
Lac, 769
NetworkID, 769
PRLInd, 769
roamIndicatorVal, 769
RoamingIndicatorList, 769
ServingSystem, 769
SystemID, 769
trackAreaCode, 769
qaQmiinstanceid
qaQmiInterfaceInfo, 765

qaQmisvctype
qaQmiInterfaceInfo, 765
qm_wds_ds_profile_extended_err_codes
qmerrno.h, 1971
qmerrno.h
eQCWWAN_ERR_API_MUTEX_TIMEOUT, 1968
eQCWWAN_ERR_BUFFER_SZ, 1967
eQCWWAN_ERR_CANCEL_OP, 1968
eQCWWAN_ERR_DRIVER, 1968
eQCWWAN_ERR_ENUM_BEGIN, 1967
eQCWWAN_ERR_ENUM_END, 1968
eQCWWAN_ERR_FILE_COPY, 1967
eQCWWAN_ERR_FILE_OPEN, 1967
eQCWWAN_ERR_GENERAL, 1967
eQCWWAN_ERR_INTERNAL, 1967
eQCWWAN_ERR_INVALID_ARG, 1967
eQCWWAN_ERR_INVALID_DEVID, 1967
eQCWWAN_ERR_INVALID_FILE, 1967
eQCWWAN_ERR_INVALID_QMI_RSP, 1967
eQCWWAN_ERR_INVALID_XID, 1968
eQCWWAN_ERR_MALFORMED_QMI_RSP, 1967
eQCWWAN_ERR_MEMORY, 1967
eQCWWAN_ERR_MULTIPLE_DEVICES, 1968
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED, 1968
eQCWWAN_ERR_NO_CANCELABLE_OP, 1968
eQCWWAN_ERR_NO_CONNECTION, 1967
eQCWWAN_ERR_NO_DEVICE, 1967
eQCWWAN_ERR_NO_SIGNAL, 1968
eQCWWAN_ERR_NONE, 1967
eQCWWAN_ERR_NULL_TLV, 1971
eQCWWAN_ERR_OFFLINE, 1968
eQCWWAN_ERR_PDU_GENERATION, 1968
eQCWWAN_ERR_QMI_ABORTED, 1968
eQCWWAN_ERR_QMI_ACCESS_DENIED, 1970
eQCWWAN_ERR_QMI_ACK_NOT_SENT, 1970
eQCWWAN_ERR_QMI_ARG_TOO_LONG, 1968
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, 1969
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, 1969
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, 1970
eQCWWAN_ERR_QMI_CALL_FAILED, 1968
eQCWWAN_ERR_QMI_CARD_BUSY_RSP, 1971
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, 1970
eQCWWAN_ERR_QMI_CAT_END, 1971
eQCWWAN_ERR_QMI_CAT_START, 1971
eQCWWAN_ERR_QMI_CAUSE_CODE, 1969
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, 1968
eQCWWAN_ERR_QMI_CONNECT, 1967
eQCWWAN_ERR_QMI_DEVICE_IN_USE, 1968
eQCWWAN_ERR_QMI_DEVICE_NOT_READY, 1969
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, 1969
eQCWWAN_ERR_QMI_DISABLED, 1969
eQCWWAN_ERR_QMI_ENCODING, 1969
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, 1971
eQCWWAN_ERR_QMI_EVENT_REG_FAILED, 1971
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, 1970
eQCWWAN_ERR_QMI_FDN_RESTRICT, 1970
eQCWWAN_ERR_QMI_FLOW_SUSPENDED, 1969
eQCWWAN_ERR_QMI_GENERAL, 1969
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, 1970
eQCWWAN_ERR_QMI_IFACE, 1967
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, 1970
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, 1969
eQCWWAN_ERR_QMI_INCORRECT_PIN, 1968
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, 1970
eQCWWAN_ERR_QMI_INJECT_TIMEOUT, 1970
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, 1969
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, 1969
eQCWWAN_ERR_QMI_INTERNAL, 1968
eQCWWAN_ERR_QMI_INVALID_ARG, 1969
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, 1968
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, 1969
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, 1971
eQCWWAN_ERR_QMI_INVALID_HANDLE, 1968
eQCWWAN_ERR_QMI_INVALID_ID, 1969
eQCWWAN_ERR_QMI_INVALID_INDEX, 1969
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, 1969
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, 1969
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, 1969
eQCWWAN_ERR_QMI_INVALID_OPERATION, 1970
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, 1968
eQCWWAN_ERR_QMI_INVALID_PINID, 1968
eQCWWAN_ERR_QMI_INVALID_PROFILE, 1968
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, 1968
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, 1969
eQCWWAN_ERR_QMI_INVALID_QMI_CMD, 1970

eQCWWAN_ERR_QMI_INVALID_QOS_ID, 1969
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, 1969
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, 1968
eQCWWAN_ERR_QMI_INVALID_TECH_PREF, 1968
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, 1971
eQCWWAN_ERR_QMI_INVALID_TRANSITION, 1969
eQCWWAN_ERR_QMI_INVALID_TX_ID, 1968
eQCWWAN_ERR_QMI_MALFORMED_MSG, 1968
eQCWWAN_ERR_QMI_MAX, 1970
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, 1969
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, 1969
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, 1969
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, 1969
eQCWWAN_ERR_QMI_MISSING_ARG, 1968
eQCWWAN_ERR_QMI_MSG_BLOCKED, 1970
eQCWWAN_ERR_QMI_NETWORK_ABORTED, 1970
eQCWWAN_ERR_QMI_NETWORK_NOT_READY, 1969
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, 1969
eQCWWAN_ERR_QMI_NO_EFFECT, 1968
eQCWWAN_ERR_QMI_NO_ENTRY, 1969
eQCWWAN_ERR_QMI_NO_FREE_PROFILE, 1968
eQCWWAN_ERR_QMI_NO_MEMORY, 1968
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, 1968
eQCWWAN_ERR_QMI_NO_RADIO, 1970
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, 1970
eQCWWAN_ERR_QMI_NO_THRESHOLDS, 1968
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, 1969
eQCWWAN_ERR_QMI_NOT_PROVISIONED, 1968
eQCWWAN_ERR_QMI_NOT_SUPPORTED, 1970
eQCWWAN_ERR_QMI_OFFSET, 1968
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, 1968
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, 1968
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, 1970
eQCWWAN_ERR_QMI_OUT_OF_CALL, 1968
eQCWWAN_ERR_QMI_PIN_BLOCKED, 1969
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, 1969
eQCWWAN_ERR_QMI_POLICY_MISMATCH, 1970
eQCWWAN_ERR_QMI_REQ, 1967
eQCWWAN_ERR_QMI_REQ_SCH, 1967
eQCWWAN_ERR_QMI_REQ_TO, 1967
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, 1969
eQCWWAN_ERR_QMI_RSP, 1967
eQCWWAN_ERR_QMI_RSP_TO, 1967
eQCWWAN_ERR_QMI_SEGMENT_ORDER, 1970
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, 1970
eQCWWAN_ERR_QMI_SESSION_INACTIVE, 1969
eQCWWAN_ERR_QMI_SESSION_INVALID, 1969
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, 1969
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, 1970
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, 1969
eQCWWAN_ERR_QMI_SMSC_ADDR, 1970
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, 1970
eQCWWAN_ERR_QMI_TPDU_TYPE, 1970
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, 1968
eQCWWAN_ERR_QMI_UNKNOWN, 1969
eQCWWAN_ERR_QMI_WIDTH, 1971
eQCWWAN_ERR_RESET, 1968
eQCWWAN_ERR_SWICM_AM_VERS_ERROR, 1970
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, 1970
eQCWWAN_ERR_SWICM_END, 1971
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, 1970
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, 1970
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, 1970
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, 1970
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, 1970
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, 1970
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, 1970
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, 1970
eQCWWAN_ERR_SWICM_SOCKET_IN_USE, 1970
eQCWWAN_ERR_SWICM_START, 1970

eQCWWAN_ERR_SWICM_TIMEOUT, 1970
eQCWWAN_ERR_SWICM_V4DWN_V6DWN, 1970
eQCWWAN_ERR_SWICM_V4DWN_V6UP, 1970
eQCWWAN_ERR_SWICM_V4UP_V6DWN, 1970
eQCWWAN_ERR_SWICM_V4UP_V6UP, 1970
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, 1971
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, 1971
eQCWWAN_ERR_SWIDCS_END, 1971
eQCWWAN_ERR_SWIDCS_FILEIO_ERR, 1971
eQCWWAN_ERR_SWIDCS_IOCTL_ERR, 1971
eQCWWAN_ERR_SWIDCS_START, 1971
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, 1971
eQCWWAN_ERR_SWIIM_END, 1971
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, 1971
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, 1971
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, 1971
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, 1971
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX, 1971
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, 1971
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE, 1971
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, 1971
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, 1971
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, 1971
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE, 1971
eQCWWAN_ERR_SWIIM_INVALID_PATH, 1971
eQCWWAN_ERR_SWIIM_OPENING_DIR, 1971
eQCWWAN_ERR_SWIIM_OPENING_FILE, 1971
eQCWWAN_ERR_SWIIM_START, 1971
eQCWWAN_ERR_SWISM_END, 1971
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, 1971
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, 1971
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, 1971
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, 1971
eQCWWAN_ERR_SWISMS_START, 1971
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, 1972
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, 1972
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, 1972
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES, 1972
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY, 1972
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET, 1972
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET, 1972
eWDS_ERR_PROFILE_REG_END, 1972
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID, 1972
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED, 1972
eWDS_ERR_PROFILE_REG_RESULT_FAIL, 1972
eWDS_ERR_PROFILE_REG_RESULT_LIST_END, 1972
qmerrno.h, 1965
eQCWWANError, 1967
qm_wds_ds_profile_extended_err_codes, 1971
QmiCbkCatEventStatusReportInd, 769
CCETlv, 769
event_Index, 769
QmiCbkLocBestAvailPosInd, 769
pAltitudeWrtEllipsoid, 774
pAltitudeWrtMeanSeaLevel, 774
pGpsTime, 774
pHeading, 774
pHeadingUnc, 774
pHorCirConf, 774
pHorEllpConf, 774
pHorReliability, 774
pHorUncCircular, 774
pHorUncEllipseOrientAzimuth, 774
pHorUncEllipseSemiMajor, 774
pHorUncEllipseSemiMinor, 774
pLatitude, 775
pLongitude, 775
pMagneticDeviation, 775
pPrecisionDilution, 775

pSensorDataUsage, 775
pSpeedHorizontal, 775
pSpeedUnc, 775
pSpeedVertical, 775
pSpeedVerticalUnc, 775
pSvUsedforFix, 775
pTechnologyMask, 775
pTimeSrc, 775
pTimeUnc, 775
pTimestampUtc, 775
pVertConfidence, 775
pVertReliability, 775
pVertUnc, 775
pXid, 775
status, 775
QmiCbkLocCradleMountInd, 775
cradleMountConfigStatus, 776
QmiCbkLocEngineStateInd, 776
engineState, 776
QmiCbkLocEventTimeSyncInd, 776
timeSyncRefCounter, 777
QmiCbkLocGetServerInd, 777
pIPv4AddrInfo, 778
pIPv6AddrInfo, 778
pURL, 778
serverStatus, 778
serverType, 778
QmiCbkLocInjectPositionInd, 778
status, 779
QmiCbkLocInjectSensorDataInd, 779
injectSensorDataStatus, 780
pAccelSamplesAccepted, 780
pAccelTempSamplesAccepted, 780
pGyroSamplesAccepted, 780
pGyroTempSamplesAccepted, 780
pOpaqueIdentifier, 780
QmiCbkLocInjectTimeInd, 780
injectTimeSyncStatus, 781
QmiCbkLocInjectUTCTimeInd, 781
status, 782
QmiCbkLocPositionReportInd, 782
pAltitudeAssumed, 786
pAltitudeWrtEllipsoid, 786
pAltitudeWrtMeanSeaLevel, 786
pFixId, 786
pGpsTime, 786
pHeading, 786
pHeadingUnc, 787
pHorConfidence, 787
pHorReliability, 787
pHorUncCircular, 787
pHorUncEllipseOrientAzimuth, 787
pHorUncEllipseSemiMajor, 787
pHorUncEllipseSemiMinor, 787
pLatitude, 787
pLeapSeconds, 787
pLongitude, 787
pMagneticDeviation, 787
pPrecisionDilution, 787
pSensorDataUsage, 787
pSpeedHorizontal, 787
pSpeedUnc, 787
pSpeedVertical, 787
pSvUsedforFix, 787
pTechnologyMask, 787
pTimeSrc, 787
pTimeUnc, 787
pTimestampUtc, 787
pVertConfidence, 787
pVertReliability, 787
pVertUnc, 787
sessionId, 787
sessionStatus, 787
QmiCbkLocSensorStreamingInd, 787
pAccelAcceptReady, 788
pAccelTempAcceptReady, 788
pGyroAcceptReady, 788
pGyroTempAcceptReady, 788
QmiCbkLocSetExtPowerConfigInd, 788
status, 789
QmiCbkNasLTECphyCaInfo, 789
sPhyCaAggPcellInfo, 790
sPhyCaAggScellDlBw, 790
sPhyCaAggScellIndType, 790
sPhyCaAggScellIndex, 790
sPhyCaAggScellInfo, 790
QmiCbkSwiOmaDmEventStatusReportInd, 790
SITlv, 790
QmiCbkSwiOmaDmEventStatusReportIndExt, 790
SITlv, 790
QmiCbkTmdMitiLvlRptInd, 790
currentMitigationLvl, 791
MitigationDevInfo, 791
QmiCbkWdsStatisticsIndState, 791
RxDropConutTlv, 791
RxOkByteCountTlv, 791
RxOkConutTlv, 791
TxDropConutTlv, 791
TxOkByteCountTlv, 791
TxOkConutTlv, 791
qmiDataBearerMasks
qaGobiApiWds.h, 1927
QmiNas3GppNetworkInfo, 792
pDesription, 793
pForbidden, 794
pInUse, 794
pMCC, 794
pMNC, 794
pPreferred, 794
pRoaming, 794
QmiNasGetRFBandInfoResp, 794
pInstancesSize, 794
pRFBandInfoElements, 794
results, 794
QmiNasPerformNetworkScanResp, 794
pInstanceSize, 794

pInstances, 794
results, 794
QmiProfileInfo
qaGobiApiWds.h, 1924
qmiSmsMessageList, 794
messageIndex, 795
messageTag, 795
QmiSwiAvmsEventReportBinaryUpdateSessionInfo-Resp
qaGobiApiSwiAvms.h, 1824
QmiSwiAvmsEventReportConfigResp
qaGobiApiSwiAvms.h, 1825
QmiSwiAvmsEventReportConnectionRequestResp
qaGobiApiSwiAvms.h, 1825
QmiSwiAvmsEventReportDataSessionStatusResp
qaGobiApiSwiAvms.h, 1826
QmiSwiAvmsEventReportHTTPStatusResp
qaGobiApiSwiAvms.h, 1826
QmiSwiAvmsEventReportNotificationResp
qaGobiApiSwiAvms.h, 1827
QmiSwiAvmsEventReportPackageIDResp
qaGobiApiSwiAvms.h, 1828
QmiSwiAvmsEventReportRegStatusResp
qaGobiApiSwiAvms.h, 1828
QmiSwiAvmsEventReportSessionTypeResp
qaGobiApiSwiAvms.h, 1828
QmiSwiAvmsEventReportWAMSParamChangeResp
qaGobiApiSwiAvms.h, 1829
QmiSwiAvmsSetSettingsAPNInfo
qaGobiApiSwiAvms.h, 1829
QmiSwiAvmsSetSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 1829
QmiSwiAvmsSetSettingsPeriodInfo
qaGobiApiSwiAvms.h, 1829
QmiWDSDataBearerTechnology
qaGobiApiWds.h, 1925
qmiWDSDataBearerTechnology, 795
currentNetwork, 796
ratMask, 796
soMask, 796
QmiWDSDataBearers
qaGobiApiWds.h, 1924
QmiWdsIpAddressInfo, 796
pIPAddressV4, 797
pIPAddressV6, 797
pIPv6prefixlen, 797
qmiWdsRunTimeSettings, 797
pAPNName, 799
pAuthentication, 799
pDomainList, 800
pGPRSGrantedQoS, 800
pGWAddressV4, 800
pIMCNflag, 800
pIPAddressV4, 800
pIPFamilyPreference, 800
pIPV6AddrInfo, 800
pIPV6GWAddrInfo, 800
pMtu, 800
pPCSCFAddrPCO, 800
pPCSCFFQDNAddrList, 800
pPDPType, 800
pPrimaryDNSV4, 800
pPrimaryDNSV6, 800
pProfileID, 800
pProfileName, 800
pSecondaryDNSV4, 800
pSecondaryDNSV6, 800
pServerAddrList, 800
pSubnetMaskV4, 800
pTechnology, 800
pUMTSGrantedQoS, 800
pUsername, 800
qmifwinfo_s, 792
dev, 792
g, 792
s, 792
qos.h, 1972
pack_qos_SLQSQosGetNetworkStatus, 1974
pack_qos_SLQSQosSwiReadApnExtraParams, 1974
pack_qos_SLQSQosSwiReadDataStats, 1975
pack_qos_SLQSSetQosEventCallback, 1976
unpack_qos_SLQSQosGetNetworkStatus, 1976
unpack_qos_SLQSQosSwiReadApnExtraParams, 1977
unpack_qos_SLQSQosSwiReadDataStats, 1977
unpack_qos_SLQSSetQosEventCallback, 1978
unpack_qos_SLQSSetQosEventCallback_ind, 1978
unpack_qos_SLQSSetQosNWStatusCallback_ind, 1978
unpack_qos_SLQSSetQosPriEventCallback_ind, 1979
unpack_qos_SLQSSetQosStatusCallback_ind, 1980
qos_id
QosMap, 805
QosClassID, 800
gDlBitRate, 801
gUlBitRate, 801
maxDlBitRate, 801
maxUlBitRate, 801
QCI, 801
qosDeliveryOrder
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
QosEventInfo, 801
pDataBearer, 802
pPacketsCountRX, 802
pPacketsCountTX, 802
pTotalBytesRX, 802
pTotalBytesTX, 802
qosFlow
sQosStat, 917

unpack_qos_SLQSQosSwiReadDataStats_t, 1162
QosFlowInfo, 803
pBearerID, 803
pQFlowState, 803
pRxQFilter, 803
pRxQFlowGranted, 803
pTxQFilter, 803
pTxQFlowGranted, 803
unpack_qos_SLQSSetQosEventCallback_ind_t, 1163
QosFlowInfoState, 804
id, 804
isNewFlow, 804
state, 804
QosMap, 804
dscp, 805
qos_id, 805
state, 805
Quality of Service (QOS), 54

RAN
unpack_nas_GetServingNetwork_t, 1100
RAT
SlqsNas3GppNetworkRAT, 87
nas_QmiNas3GppNetworkRAT, 493
RATMask
CurrNetworkInfo, 202
currNetworkInfo, 203
wds_currNetworkInfo, 1366
REGISTER_EVENT
qaGobiApiCbk.h, 1559
REGISTER_SRV
qaGobiApiCbk.h, 1559
RFBandInfoElements, 812
activeBandClass, 813
activeChannel, 813
radioInterface, 813
unpack_nas_GetRFInfo_t, 1098
RFTlv
unpack_nas_SetEventReportInd_t, 1104
RMAutoConnect
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
RMSetTransferStatistics
qaGobiApiWds.h, 1939
ROAMING_INDEX
qaNasPerformNetworkScan.h, 1964
RPCause
SMSAsyncRawSend_s, 901
RPTlv
NASQmiCbkNasSystemSelPrefInd, 560
RRTlv
unpack_nas_SetEventReportInd_t, 1104
RSRPThresListLen
nas_RSRPThresh, 497
RSRPThresh, 816
RSRPThresh, 816
pRSRPThresList, 816
RSRPThresListLen, 816
RSRQThresListLen
nas_RSRQThresh, 498
RSRQThresh, 817
RSRQThresh, 817
pRSRQThresList, 817
RSRQThresListLen, 817
RSSIThresListLen
nas_RSSIThresh, 498
RSSIThresh, 818
RSSIThresh, 817
pRSSIThresList, 818
RSSIThresListLen, 818
RX_EC_IO
nas_NetworkStat1x, 476
NetworkStat1x, 575
RX_PWR
nas_NetworkStat1x, 476
nas_NetworkStatEVDO, 478
NetworkStat1x, 575
NetworkStatEVDO, 576
RXAGCList, 818
pRXAIG, 819
pRXComprSlope, 819
pRXComprThres, 819
pRXExpSlope, 819
pRXExpThres, 819
pRXStaticGain, 819
RXAVCList, 819
pAVRXAVCHeadroom, 819
pAVRXAVCSens, 820
RXChan
LTEInfo, 393
nas_LTEInfo, 457
rXDroppedCount
unpack_wds_GetPacketStatus_t, 1239
RXOKBytesCount
DUNCallInfoInd, 233
rXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1239
rXOkBytesCount
unpack_wds_GetPacketStatus_t, 1239
RXPCMIIRFltr, 821
pFlag, 822
pStage0Val, 822
pStage1Val, 822
pStage2Val, 822
pStage3Val, 822
pStage4Val, 822
pStageCnt, 822
rXPacketErrors
unpack_wds_GetPacketStatus_t, 1239
rXPacketOverflows
unpack_wds_GetPacketStatus_t, 1239
rXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1239
radio
unpack_nas_GetSignalStrengths_t, 1101
radio_if

nasGetTxRxInfoReq, 538
pack_nas_SLQSNasGetTxRxInfo_t, 633
radioIf
ecioListElement, 235
errorRateListElement, 239
nas_ecioListElement, 436
nas_errorRateListElement, 438
nas_rsrqInformation, 497
nas_rxSignalStrengthListElement, 501
nasNetworkReject, 548
rsrqInformation, 817
rxSignalStrengthListElement, 824
RadioIfaces
unpack_dms_GetDeviceCap_t, 1030
unpack_dms_GetDeviceCapabilities_t, 1031
unpack_nas_GetServingNetwork_t, 1100
RadioIfacesSize
unpack_dms_GetDeviceCap_t, 1030
unpack_nas_GetServingNetwork_t, 1100
radioIfacesSize
unpack_dms_GetDeviceCapabilities_t, 1031
radioInterface
nas_RFInfoTlv, 495
nas_roamIndList, 496
nas_servSystem, 504
nas_SignalStrengthTlv, 505
nas_timeInfo, 516
RFBandInfoElements, 813
roamIndList, 815
servSystem, 837
timeInfo, 955
radioInterfaceList
NASServingSystemInfo, 562
ServingSystemInfo, 835
radioInterfaceNo
NASServingSystemInfo, 562
ServingSystemInfo, 836
radioInterfaceSize
nas_RFInfoTlv, 495
range
Port, 747
unpack_qos_Port_t, 1157
RankIndicatorInd, 805
Count1, 805
Count2, 805
rat
CSGID, 194
MNRInfo, 412
nas_CSGID, 429
nas_MNRInfo, 473
ratMask
dataBearerTechnology, 213
qmiWDSDataBearerTechnology, 796
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
ratValue
DataBearerTech, 211
rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 616
rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 616
rawLen
fileAttributes, 243
uim_fileAttributes, 978
rawValue
fileAttributes, 243
uim_fileAttributes, 978
rcv4
ssdatasession_params, 921
rcv6
ssdatasession_params, 921
readResult, 805
content, 806
contentLen, 806
readTransparent
pack_uim_ReadTransparent_t, 676
UIMReadTransparentReq, 1000
readTransparentInfo, 806
length, 806
offset, 806
Reason
pack_voice_SLQSVoiceGetCallForwardingStatus-_t, 693
pack_voice_SLQSVoiceSetCallBarringPassword_t, 699
voiceGetCallFWReq, 1318
voiceSetCallBarringPwdInfo, 1341
reason
ccSUPSType, 161
pack_voice_SLQSVoiceGetCallBarring_t, 692
pack_voice_SLQSVoiceSetSUPSService_t, 703
redirNumInfo, 808
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1165
voice_ccSUPSType, 1284
voiceGetCallBarringReq, 1316
voiceSetSUPSServiceReq, 1348
receiptAction
sms_routeEntry, 895
smsRouteEntry, 913
receivedBytes
omaDmFotaTlvExt, 587
reconfigReqd
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1252
reconfiguration_required
slqsSessionStateInfo, 879
recordCount
fileAttributes, 243
uim_fileAttributes, 978
recordSize
fileAttributes, 243
uim_fileAttributes, 978
redirNumInfo, 806
numLen, 808

numPlan, 808
numType, 808
number, 808
PI, 808
reason, 808
SI, 808
RedirPartyNum
arrRedirPartyNum, 133
voice_arrRedirPartyNum, 1275
refData
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1060
refString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1060
ReferenceID
CatAlPhaIdentifierTlv, 158
CatEventIDDataTlv, 159
refpn
CDMAInfo, 164
nas_CDMAInfo, 421
refreshComplete
pack_uim_SLQSUIMRefreshComplete_t, 681
UIMRefreshCompleteReq, 1002
RefreshMode
CatRefreshTlv, 160
RefreshStage
CatRefreshTlv, 160
regAction
nasInitNetworkReg, 543
pack_nas_SLQSInitiateNetworkRegistration_t, 628
RegForeignNID
unpack_nas_GetCDMANetworkParameters_t, 1093
RegForeignSID
unpack_nas_GetCDMANetworkParameters_t, 1093
RegHomeSID
unpack_nas_GetCDMANetworkParameters_t, 1093
regInd
_transLayerInfoNotification, 112
regPrd
AddCDMASysInfo, 116
nas_AddCDMASysInfo, 416
regRefresh
pack_uim_SLQSUIMRefreshRegister_t, 683
UIMRefreshRegisterReq, 1005
regRejectInfoValid
GSMSysInfo, 296
LTESysInfo, 408
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
regState
nas_servSystem, 504
servSystem, 837
RegStatus
_SwiAvmsEventInd, 100
Region
fwinfo_s, 250
registerFlag
registerRefresh, 809
uim_registerRefresh, 984
registerRefresh, 808
arrfileInfo, 809
numFiles, 809
registerFlag, 809
voteForInit, 809
RegistrationState
unpack_nas_GetServingNetwork_t, 1100
registrationState
NASServingSystemInfo, 562
ServingSystemInfo, 836
rejCause
GSMSysInfo, 296
LTESysInfo, 408
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
rejectCause
nas_RejectReasonTlv, 494
nasNetworkReject, 548
rejectSrvDomain
GSMSysInfo, 296
LTESysInfo, 408
nas_GSMSysInfo, 446
nas_LTESysInfo, 470
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
reliabilityClass
GPRSQoS, 285
GPRSRequestedQoS, 286
LibPackGPRSRequestedQoS, 333
wds_GPRSQoS, 1373
remPartyNumber
remotePartyNum, 811
voice_remotePartyNum, 1300
remainingRetries, 809
unblockLeft, 809
verifyLeft, 809
Remote Management Service (RMS), 45
RemotePartyName
getAllCallRmtPtyName, 254
voice_getAllCallRmtPtyName, 1293
remotePartyName, 809
callerName, 810
codingScheme, 810
nameLen, 810
namePI, 810
RemotePartyNum
getAllCallRmtPtyNum, 254
voice_getAllCallRmtPtyNum, 1294
remotePartyNum, 810

numLen, 811
presentationInd, 811
remPartyNumber, 811
ReqFieldsList, 811
requestFields, 811
requestFieldsLen, 811
requestFields
ReqFieldsList, 811
requestFieldsLen
ReqFieldsList, 811
resBerRatio
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
ResCode
FirmwareUpdatStat, 246
GetAudioVolTLBConfigResp, 260
SetAudioVolTLBConfigResp, 843
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1061
reserved
omaDmFotaTlvExt, 587
resetInfoInd
pack_dms_SLQSDmsSwiIndicationRegister_t, 598
ResetInfoNotification
qaGobiApiCbk.h, 1564
ResetPDSData
qaGobiApiPds.h, 1774
ResetToFactoryDefaults
qaGobiApiDms.h, 1662
RespFieldsList, 812
responseFields, 812
responseFieldsLen, 812
responseFields
RespFieldsList, 812
responseFieldsLen
RespFieldsList, 812
results
QmiNasGetRFBandInfoResp, 794
QmiNasPerformNetworkScanResp, 794
RetryCount
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
revPolarity
lineCtrlInfo, 361
revTunneling
unpack_wds_GetMobileIPProfile_t, 1236
ReverseMac
nas_protocolSubtypeElement, 491
protocolSubtypeElement, 763
RfSarState
pack_sar_SLQSSetRfSarState_t, 660
RmTrasnferStaticsReq
pack_wds_RMSetTransferStatistics_t, 708
rmTrasnferStaticsReq, 813
bResetStatistics, 814
ulMask, 814
rms.h, 1980
pack_rms_GetSMSWake, 1981
pack_rms_SetSMSWake, 1981
unpack_rms_GetSMSWake, 1981
unpack_rms_SetSMSWake, 1982
RmtPtyNum
arrRemotePartyNum, 134
voice_arrRemotePartyNum, 1276
roamIndList, 814
numInstances, 815
radioInterface, 815
roamIndicator, 815
roamIndicator
nas_roamIndList, 496
roamIndList, 815
RoamIndicatorVal
unpack_nas_SLQSGetServingSystem_t, 1114
roamIndicatorVal
qaQmiServingSystemParam, 769
roamOrigVoiceSO
prefVoiceSO, 750
voice_prefVoiceSO, 1299
RoamPref
NASRoamPreferenceTlv, 560
roamStatus
nas_sysInfoCommon, 511
sysInfoCommon, 944
roamStatusValid
nas_sysInfoCommon, 511
sysInfoCommon, 944
roamTimer, 815
namID, 816
roamTimerValue, 816
roamTimerValue
roamTimer, 816
voice_roamTimer, 1301
Roaming
nas_QmiNas3GppNetworkInfo, 493
SlqsNas3GppNetworkInfo, 874
unpack_nas_GetCDMANetworkParameters_t, 1093
unpack_nas_GetServingNetwork_t, 1100
roaming
unpack_nas_SetRoamingIndicatorCallback_ind_t, 1106
roaming_ind
RoamingInfo, 815
RoamingIndicatorList
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1114
RoamingInfo, 815
roaming_ind, 815
TlvPresent, 815
routeList
sms_setRoutesReq, 899
smsSetRoutesReq, 913
routeStorage
sms_routeEntry, 895

smsRouteEntry, 913
rptPosData
pack_pds_SetEventReportCallback_t, 648
rptRate
LTESigRptCfg, 401
LTESigRptConfig, 402
nas_LTESigRptCfg, 465
nas_LTESigRptConfig, 466
rscp
nas_rxInfo, 499
nas_UMTSInfo, 519
rxInfo, 821
TDSCDMASigInfoExt, 948
tdscdmaSigInfoExt, 947
UMTSInfo, 1014
rsrp
cellParams, 176
LTESSInfo, 405
lteSSInfo, 405
nas_cellParams, 426
nas_rxInfo, 499
nas_RxSigInfo, 500
nas_umtsLTENbrCell, 520
rxInfo, 821
RxSigInfo, 823
umtsLTENbrCell, 1015
rsrplevel
lteRsrpinformation, 399
nas_lteRsrpinformation, 463
rsrq
cellParams, 176
LTESSInfo, 405
lteSSInfo, 405
nas_cellParams, 426
nas_rsrqInformation, 497
nas_SccRxInfo, 502
nas_umtsLTENbrCell, 521
rsrqInformation, 817
SccRxInfo, 828
umtsLTENbrCell, 1015
rsrqDelta
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
rsrqInfo
nas_SLQSSignalStrengthsInformation, 508
slqsSignalStrengthInfo, 881
SLQSSignalStrengthsInformation, 885
unpack_nas_SLQSGetSignalStrength_t, 1116
rsrqInformation, 816
radioIf, 817
rsrq, 817
rssi
CDMASSInfo, 170
cdmaSSInfo, 170
cellParams, 176
gsmCellInfo, 291
HDRSSInfo, 303
hdrSSInfo, 304
LTESSInfo, 405
lteSSInfo, 405
nas_cellParams, 426
nas_gsmCellInfo, 442
TDSCDMASigInfoExt, 948
tdscdmaSigInfoExt, 947
unpack_nas_GetSignalStrengths_t, 1101
rts
WdsRunTimeSettings, 1395
rx_bytes
NetStats, 571
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
rx_errors
NetStats, 571
rx_overflows
NetStats, 572
rx_packets
NetStats, 572
rx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
rxChainIndex
nas_RxSigInfo, 500
RxSigInfo, 823
RxDropConutTlv
QmiCbkWdsStatisticsIndState, 791
unpack_RMTransferStatistics_ind_t, 1176
rxInfo, 820
ecio, 820
isRadioTuned, 820
phase, 820
rscp, 821
rsrp, 821
rxPower, 821
rxLev
GERANInfo, 252
nas_GERANInfo, 440
rxOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1247
RxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 791
unpack_RMTransferStatistics_ind_t, 1176
RxOkConutTlv
QmiCbkWdsStatisticsIndState, 791
unpack_RMTransferStatistics_ind_t, 1176
rxPower
nas_rxInfo, 500
nas_RxSigInfo, 501
rxInfo, 821
RxSigInfo, 823
RxQFilter
unpack_qos_QosFlowInfo_t, 1159
RxQFlowGranted
unpack_qos_QosFlowInfo_t, 1159
RxSigInfo, 822
isRadioTuned, 823
rsrp, 823

rxChainIndex, 823
rxPower, 823
rxSignalStrength
nas_rxSignalStrengthListElement, 501
rxSignalStrengthListElement, 824
rxSignalStrengthDelta
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
rxSignalStrengthInfo
nas_SLQSSignalStrengthsInformation, 508
SLQSSignalStrengthsInformation, 885
rxSignalStrengthList
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
rxSignalStrengthListElement, 823
radioIf, 824
rxSignalStrength, 824
rxSignalStrengthListLen
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116

s
qmifwinfo_s, 792
SMS_EVENT_ETWS
qaGobiApiCbk.h, 1598
SMS_EVENT_ETWS_PLMN
qaGobiApiCbk.h, 1598
SMS_EVENT_MESSAGE_MODE
qaGobiApiCbk.h, 1598
SMS_EVENT_MT_MESSAGE
qaGobiApiCbk.h, 1598
SMS_EVENT_SMS_ON_IMS
qaGobiApiCbk.h, 1598
SMS_EVENT_SMSC_ADDRESS
qaGobiApiCbk.h, 1598
SMS_EVENT_TRANSFER_ROUTE_MT_MESSAGE
qaGobiApiCbk.h, 1598
SYS_SRV_DOMAIN_CAMPED
qaGobiApiCbk.h, 1598
SYS_SRV_DOMAIN_CS_ONLY
qaGobiApiCbk.h, 1598
SYS_SRV_DOMAIN_CS_PS
qaGobiApiCbk.h, 1598
SYS_SRV_DOMAIN_NO_SRV
qaGobiApiCbk.h, 1598
SYS_SRV_DOMAIN_PS_ONLY
qaGobiApiCbk.h, 1598
sApnExtraParams, 824
ambr_dl, 825
ambr_dl_ext, 825
ambr_dl_ext2, 825
ambr_ul, 825
ambr_ul_ext, 825
ambr_ul_ext2, 825
apnId, 825
SCI
unpack_nas_GetCDMANetworkParameters_t, 1093
SCM
unpack_nas_GetCDMANetworkParameters_t, 1093
SDPTlv
NASQmiCbkNasSystemSelPrefInd, 560
SDU_HDR_LEN
common.h, 1401
SECOND_INSTANCE
qaGobiApiCbk.h, 1559
sGetDeviceSeriesResult, 863
eDevice, 863
uResult, 863
SHORT
SwiDataTypes.h, 2007
SI
calledPartyInfo, 146
callFWExtInfo, 150
callingPartyInfo, 154
redirNumInfo, 808
voice_callFWExtInfo, 1281
SITlv
QmiCbkSwiOmaDmEventStatusReportInd, 790
QmiCbkSwiOmaDmEventStatusReportIndExt, 790
sIntraSearch
LTEInfoIntrafreq, 395
nas_LTEInfoIntrafreq, 460
SLQSAVMSGetSessionInfo
qaGobiApiSwiAvms.h, 1831
SLQSAVMSGetSettings
qaGobiApiSwiAvms.h, 1831
SLQSAVMSSelection
qaGobiApiSwiAvms.h, 1831
SLQSAVMSSetEventReport
qaGobiApiSwiAvms.h, 1832
SLQSAVMSSetSettings
qaGobiApiSwiAvms.h, 1832
SLQSAVMSSetSettingsIgnoreAutoReboot
qaGobiApiSwiAvms.h, 1833
SLQSAVMSSetSettingsReq
qaGobiApiSwiAvms.h, 1829
SLQSAVMSSettings
qaGobiApiSwiAvms.h, 1830
SLQSAVMSSettingsAPNInfo
qaGobiApiSwiAvms.h, 1830
SLQSAVMSSettingsConnectionRetryTimers
qaGobiApiSwiAvms.h, 1830
SLQSAVMSSettingsPeriodsInfo
qaGobiApiSwiAvms.h, 1830
SLQSAVMSStartSession
qaGobiApiSwiAvms.h, 1833
SLQSAVMSStopSession
qaGobiApiSwiAvms.h, 1834
SLQSAutoConnect
qaGobiApiWds.h, 1948
SLQSCDMADecodeMTTextMsg
qaGobiApiSms.h, 1801
SLQSCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1802
SLQSConfigSigInfo

qaGobiApiNas.h, 1749
SLQSCreateProfile
qaGobiApiWds.h, 1948
SLQSDeleteProfile
qaGobiApiWds.h, 1949
SLQSDeleteProfileParams, 871
profileIndex, 872
profileType, 872
SLQSDeleteSMS
qaGobiApiSms.h, 1802
SLQSDmsSwiGetResetInfo
qaGobiApiDms.h, 1663
SLQSDmsSwiIndicationRegister
qaGobiApiDms.h, 1663
SLQSDownloadFirmwareToSlot
qaGobiApiFms.h, 1692
SLQSFWINFO_SKU_SZ
qaGobiApiFms.h, 1686
SLQSGet3GPPConfigItem
qaGobiApiWds.h, 1950
SLQSGetAGPSConfig
qaGobiApiPds.h, 1778
SLQSGetAudioPathConfig
qaGobiApiAudio.h, 1544
SLQSGetAudioProfile
qaGobiApiAudio.h, 1545
SLQSGetAudioVolTLBConfig
qaGobiApiAudio.h, 1545
SLQSGetBandCapabilities
qaGobiApiDms.h, 1663
SLQSGetBandCapability
qaGobiApiDms.h, 1663
SLQSGetBootVersionNumber
qaGobiApiFms.h, 1693
SLQSGetByteTotals
qaGobiApiWds.h, 1950
SLQSGetConnectionRate
qaGobiApiWds.h, 1950
SLQSGetCurrDataSystemStat
qaGobiApiWds.h, 1951
SLQSGetCurrentChannelRate
qaGobiApiWds.h, 1951
SLQSGetCurrentPRLInfo
qaGobiApiDms.h, 1665
SLQSGetCustFeatures
qaGobiApiDms.h, 1665
SLQSGetCustFeaturesV2
qaGobiApiDms.h, 1666
SLQSGetDUNCallInfo
qaGobiApiWds.h, 1953
SLQSGetDataBearerTechnology
qaGobiApiWds.h, 1952
SLQSGetDataBearerTechnologyExt
qaGobiApiWds.h, 1952
SLQSGetDeviceMode
qaGobiApiDcs.h, 1640
SLQSGetERIFile
qaGobiApiDms.h, 1666
SLQSGetErrorRate
qaGobiApiNas.h, 1750
SLQSGetFirmwareInfo
qaGobiApiFms.h, 1694
SLQSGetGPSStateInfo
qaGobiApiPds.h, 1779
SLQSGetIMSARegStatus
qaGobiApiImsa.h, 1708
SLQSGetIMSAServiceStatus
qaGobiApiImsa.h, 1708
SLQSGetIMSASupportedFields
qaGobiApiImsa.h, 1709
SLQSGetIMSASupportedMsg
qaGobiApiImsa.h, 1709
SLQSGetIMSSMSConfig
qaGobiApiIms.h, 1702
SLQSGetIMSUserConfig
qaGobiApiIms.h, 1702
SLQSGetIMSVoIPConfig
qaGobiApiIms.h, 1703
SLQSGetImageInfo
qaGobiApiFms.h, 1694
SLQSGetImageInfo_9x15
qaGobiApiFms.h, 1695
SLQSGetImageInfoMC77xx
qaGobiApiFms.h, 1695
SLQSGetImageInfoMC83xx
qaGobiApiFms.h, 1696
SLQSGetIndicationRegister
qaGobiApiSms.h, 1803
SLQSGetM2MAVMute
qaGobiApiSwiAudio.h, 1818
SLQSGetM2MAudioProfile
qaGobiApiSwiAudio.h, 1817
SLQSGetM2MAudioVolume
qaGobiApiSwiAudio.h, 1818
SLQSGetM2MSpkrGain
qaGobiApiSwiAudio.h, 1818
SLQSGetMessageWaiting
qaGobiApiSms.h, 1804
SLQSGetNetStatistic
qaGobiApiDcs.h, 1640
SLQSGetNetworkTime
qaGobiApiNas.h, 1750
SLQSGetOperatorNameData
qaGobiApiNas.h, 1750
SLQSGetPLMNName
qaGobiApiNas.h, 1751
SLQSGetPacketStatistics
qaGobiApiWds.h, 1953
SLQSGetPidof
qaGobiApiSwi.h, 1815
SLQSGetProfile
qaGobiApiWds.h, 1954
SLQSGetProfileSettings
qaGobiApiWds.h, 1955
SLQSGetRegMgrConfig
qaGobiApiIms.h, 1703

SLQSGetRfSarState
qaGobiApiSar.h, 1792
SLQSGetRuntimeSettings
qaGobiApiWds.h, 1956
SLQSGetSIPConfig
qaGobiApiIms.h, 1704
SLQSGetSMS
qaGobiApiSms.h, 1804
SLQSGetSMSList
qaGobiApiSms.h, 1806
SLQSGetSdkVersion
qaGobiApiSwi.h, 1816
SLQSGetSerialNumbers
qaGobiApiDms.h, 1666
SLQSGetServingSystem
qaGobiApiNas.h, 1751
SLQSGetSessionState
qaGobiApiWds.h, 1956
SLQSGetSignalStrength
qaGobiApiNas.h, 1752
SLQSGetSmsBroadcastConfig
qaGobiApiSms.h, 1805
SLQSGetSysSelectionPref
qaGobiApiNas.h, 1752
SLQSGetTransLayerInfo
qaGobiApiSms.h, 1807
SLQSGetTransNWRegInfo
qaGobiApiSms.h, 1807
SLQSGetUsbPortNames
qaGobiApiDcs.h, 1641
SLQSGetValidFwPriCombinations
qaGobiApiFms.h, 1696
SLQSImsConfigIndicationRegister
qaGobiApiIms.h, 1704
SLQSInitiateNetworkRegistration
qaGobiApiNas.h, 1753
SLQSIsSpkgFormatRequired
qaGobiApiFms.h, 1697
SLQSKillSDKProcess
qaGobiApiDcs.h, 1641
SLQSLOCDelAssData
qaGobiApiLoc.h, 1711
SLQSLOCEventRegister
qaGobiApiLoc.h, 1712
SLQSLOCGetBestAvailPos
qaGobiApiLoc.h, 1712
SLQSLOCGetServer
qaGobiApiLoc.h, 1713
SLQSLOCInjectPosition
qaGobiApiLoc.h, 1713
SLQSLOCInjectSensorData
qaGobiApiLoc.h, 1714
SLQSLOCInjectUTCTime
qaGobiApiLoc.h, 1714
SLQSLOCSetCradleMountConfig
qaGobiApiLoc.h, 1714
SLQSLOCSetExtPowerState
qaGobiApiLoc.h, 1715
SLQSLOCSetOpMode
qaGobiApiLoc.h, 1715
SLQSLOCStart
qaGobiApiLoc.h, 1716
SLQSLOCStop
qaGobiApiLoc.h, 1716
SLQSModifyProfile
qaGobiApiWds.h, 1957
SLQSModifySMSStatus
qaGobiApiSms.h, 1807
SLQSNASGetLTECPHYCaInfo
qaGobiApiNas.h, 1755
SLQSNASSwiGetChannelLock
qaGobiApiNas.h, 1758
SLQSNASSwiSetChannelLock
qaGobiApiNas.h, 1760
SLQSNasConfigSigInfo2
qaGobiApiNas.h, 1753
SLQSNasGet3GPP2Subscription
qaGobiApiNas.h, 1754
SLQSNasGetCellLocationInfo
qaGobiApiNas.h, 1754
SLQSNasGetHDRColorCode
qaGobiApiNas.h, 1755
SLQSNasGetSigInfo
qaGobiApiNas.h, 1755
SLQSNasGetSysInfo
qaGobiApiNas.h, 1756
SLQSNasGetTxRxInfo
qaGobiApiNas.h, 1756
SLQSNasIndicationRegister
qaGobiApiNas.h, 1757
SLQSNasIndicationRegisterExt
qaGobiApiNas.h, 1758
SLQSNasIndicationRegisterLTECphyCa
qaGobiApiNas.h, 1758
SLQSNasNetworkRejectCallback
qaGobiApiCbk.h, 1614
SLQSNasNetworkTimeCallBack
qaGobiApiCbk.h, 1614
SLQSNasSigInfo2CallBack
qaGobiApiCbk.h, 1614
SLQSNasSigInfoCallBack
qaGobiApiCbk.h, 1615
SLQSNasSwiIndicationRegister
qaGobiApiNas.h, 1759
SLQSNasSwiModemStatus
qaGobiApiNas.h, 1759
SLQSNasSwiOTAMessageCallback
qaGobiApiCbk.h, 1615
SLQSNasSysInfoCallBack
qaGobiApiCbk.h, 1616
SLQSNasTimerCallback
qaGobiApiCbk.h, 1616
SLQSOMADMCancelSession
qaGobiApiSwiOmadms.h, 1840
SLQSOMADMGetSessionInfo
qaGobiApiSwiOmadms.h, 1840

SLQSOMADMGetSettings
qaGobiApiSwiOmadms.h, 1841
SLQSOMADMGetSettings2
qaGobiApiSwiOmadms.h, 1841
SLQSOMADMSendSelection
qaGobiApiSwiOmadms.h, 1842
SLQSOMADMSendSelection2
qaGobiApiSwiOmadms.h, 1842
SLQSOMADMSessionInfo
qaGobiApiSwiOmadms.h, 1835
SLQSOMADMSetSettings
qaGobiApiSwiOmadms.h, 1843
SLQSOMADMSetSettings2
qaGobiApiSwiOmadms.h, 1844
SLQSOMADMSetSettings3
qaGobiApiSwiOmadms.h, 1844
SLQSOMADMSettings
qaGobiApiSwiOmadms.h, 1837
SLQSOMADMSettingsReqParams
qaGobiApiSwiOmadms.h, 1838
SLQSOMADMSettingsReqParams3
qaGobiApiSwiOmadms.h, 1839
SLQSOMADMStartSession
qaGobiApiSwiOmadms.h, 1844
SLQSOMADMStartSession2
qaGobiApiSwiOmadms.h, 1845
SLQSOriginateUSSD
qaGobiApiVoice.h, 1904
SLQSPDSDeterminePosition
qaGobiApiPds.h, 1779
SLQSPDSInjectAbsoluteTimeReference
qaGobiApiPds.h, 1780
SLQSPDSInjectPositionData
qaGobiApiPds.h, 1780
SLQSPerformNetworkScan
qaGobiApiNas.h, 1760
SLQSQosGetFlowStatus
qaGobiApiQos.h, 1784
SLQSQosGetGranted
qaGobiApiQos.h, 1784
SLQSQosGetNWProf
qaGobiApiQos.h, 1785
SLQSQosGetNetworkStatus
qaGobiApiQos.h, 1785
SLQSQosModify
qaGobiApiQos.h, 1786
SLQSQosRel
qaGobiApiQos.h, 1786
SLQSQosReq
qaGobiApiQos.h, 1787
SLQSQosReset
qaGobiApiQos.h, 1787
SLQSQosResume
qaGobiApiQos.h, 1788
SLQSQosSuspend
qaGobiApiQos.h, 1788
SLQSQosSwiReadApnExtraParams
qaGobiApiQos.h, 1788
SLQSQosSwiReadDataStats
qaGobiApiQos.h, 1789
SLQSRegisterIMSAIndication
qaGobiApiImsa.h, 1710
SLQSResetPacketStatics
qaGobiApiWds.h, 1957
SLQSSGetDHCPv4ClientConfig
qaGobiApiWds.h, 1959
SLQSSGetLoopback
qaGobiApiWds.h, 1960
SLQSSTlv
unpack_nas_SetEventReportInd_t, 1104
SLQSSSetDHCPv4ClientConfig
qaGobiApiWds.h, 1960
SLQSSSetLoopback
qaGobiApiWds.h, 1960
SLQSSendAsyncSMS
qaGobiApiSms.h, 1808
SLQSSendLongSMS
qaGobiApiSms.h, 1808
SLQSSendRawQMI
qaGobiApiSwi.h, 1816
SLQSSendSMS
qaGobiApiSms.h, 1809
SLQSSet3GPPConfigItem
qaGobiApiWds.h, 1957
SLQSSetAGPSConfig
qaGobiApiPds.h, 1781
SLQSSetAudioPathConfig
qaGobiApiAudio.h, 1546
SLQSSetAudioProfile
qaGobiApiAudio.h, 1546
SLQSSetAudioVolTLBConfig
qaGobiApiAudio.h, 1547
SLQSSetBandPreference
qaGobiApiNas.h, 1760
SLQSSetBandPreferenceCbk
qaGobiApiCbk.h, 1616
SLQSSetCrashStateCheckIgnore
qaGobiApiFms.h, 1697
SLQSSetCustFeatures
qaGobiApiDms.h, 1667
SLQSSetCustFeaturesV2
qaGobiApiDms.h, 1667
SLQSSetDHCPv4ClientLeaseStatusCallback
qaGobiApiCbk.h, 1617
SLQSSetDUNCallInfoCallback
qaGobiApiCbk.h, 1618
SLQSSetDataSystemStatusCallback
qaGobiApiCbk.h, 1617
SLQSSetIMSAPdpStatusCallback
qaGobiApiCbk.h, 1618
SLQSSetIMSARatStatusCallback
qaGobiApiCbk.h, 1618
SLQSSetIMSARegStatusCallback
qaGobiApiCbk.h, 1619
SLQSSetIMSASvcStatusCallback
qaGobiApiCbk.h, 1619

SLQSSetIMSSMSConfig
qaGobiApiIms.h, 1705
SLQSSetIMSSMSConfigCallback
qaGobiApiCbk.h, 1620
SLQSSetIMSUserConfig
qaGobiApiIms.h, 1705
SLQSSetIMSUserConfigCallback
qaGobiApiCbk.h, 1620
SLQSSetIMSVoIPConfig
qaGobiApiIms.h, 1706
SLQSSetIMSVoIPConfigCallback
qaGobiApiCbk.h, 1621
SLQSSetIndicationRegister
qaGobiApiSms.h, 1810
SLQSSetLocInjectPositionCallback
qaGobiApiCbk.h, 1621
SLQSSetLocInjectUTCTimeCallback
qaGobiApiCbk.h, 1621
SLQSSetLoggingMask
qaGobiApiDcs.h, 1642
SLQSSetM2MAVMute
qaGobiApiSwiAudio.h, 1821
SLQSSetM2MAudioAVCFG
qaGobiApiSwiAudio.h, 1819
SLQSSetM2MAudioLPBK
qaGobiApiSwiAudio.h, 1819
SLQSSetM2MAudioNVDef
qaGobiApiSwiAudio.h, 1820
SLQSSetM2MAudioProfile
qaGobiApiSwiAudio.h, 1820
SLQSSetM2MAudioVolume
qaGobiApiSwiAudio.h, 1820
SLQSSetM2MSpkrGain
qaGobiApiSwiAudio.h, 1821
SLQSSetModemTempCallback
qaGobiApiCbk.h, 1621
SLQSSetPacketSrvStatusCallback
qaGobiApiCbk.h, 1622
SLQSSetPositionMethodState
qaGobiApiPds.h, 1781
SLQSSetProfile
qaGobiApiWds.h, 1958
SLQSSetQosEventCallback
qaGobiApiCbk.h, 1622
SLQSSetQosNWStatusCallback
qaGobiApiCbk.h, 1623
SLQSSetQosPriEventCallback
qaGobiApiCbk.h, 1623
SLQSSetQosStatusCallback
qaGobiApiCbk.h, 1623
SLQSSetRegMgrConfig
qaGobiApiIms.h, 1706
SLQSSetRegMgrConfigCallback
qaGobiApiCbk.h, 1624
SLQSSetRfSarState
qaGobiApiSar.h, 1793
SLQSSetSDKTerminatedCallback
qaGobiApiCbk.h, 1624
SLQSSetSIMBasedImageSwitching
qaGobiApiFms.h, 1698
SLQSSetSIPConfig
qaGobiApiIms.h, 1707
SLQSSetSIPConfigCallback
qaGobiApiCbk.h, 1626
SLQSSetSMSEventCallback
qaGobiApiCbk.h, 1627
SLQSSetServingSystemCallback
qaGobiApiCbk.h, 1625
SLQSSetSessionStateCallback
qaGobiApiCbk.h, 1625
SLQSSetSignalStrengthsCallback
qaGobiApiCbk.h, 1626
SLQSSetSmsBroadcastActivation
qaGobiApiSms.h, 1810
SLQSSetSmsBroadcastConfig
qaGobiApiSms.h, 1811
SLQSSetSmsStorage
qaGobiApiSms.h, 1811
SLQSSetSpkgFormatRequired
qaGobiApiFms.h, 1698
SLQSSetSwiGetResetInfoCallback
qaGobiApiCbk.h, 1627
SLQSSetSwiHDRPersCallback
qaGobiApiCbk.h, 1627
SLQSSetSysSelectionPref
qaGobiApiNas.h, 1762
SLQSSetSysSelectionPrefCallBack
qaGobiApiCbk.h, 1628
SLQSSetTransLayerInfoCallback
qaGobiApiCbk.h, 1628
SLQSSetTransNWRegInfoCallback
qaGobiApiCbk.h, 1628
SLQSSetWdsEventCallback
qaGobiApiCbk.h, 1629
SLQSSetWdsTransferStatisticCallback
qaGobiApiCbk.h, 1630
SLQSSignalStrengthsIndReq, 882
ecioDelta, 883
ecioThresholdList, 883
ecioThresholdListLen, 883
ioDelta, 883
lteRsrpDelta, 883
lteSnrDelta, 883
rsrqDelta, 883
rxSignalStrengthDelta, 883
sinrDelta, 883
sinrThresholdList, 883
sinrThresholdListLen, 883
SLQSSignalStrengthsInformation, 884
ecioInfo, 884
errorRateInfo, 885
io, 885
lteRsrpinfo, 885
lteSnrinfo, 885
rsrqInfo, 885
rxSignalStrengthInfo, 885

sinr, 885
SLQSSmsGetMaxStorageSize
qaGobiApiSms.h, 1812
SLQSSmsGetMessageProtocol
qaGobiApiSms.h, 1812
SLQSSmsSetRoutes
qaGobiApiSms.h, 1813
SLQSStart
qaGobiApiDcs.h, 1642
SLQSStart_AVAgent
qaGobiApiDcs.h, 1643
SLQSStartSrv
qaGobiApiDcs.h, 1643
SLQSStartStopDataSession
qaGobiApiWds.h, 1961
SLQSSwiAVMSEventReportCallback
qaGobiApiCbk.h, 1630
SLQSSwiClearDyingGaspStatistics
qaGobiApiDms.h, 1668
SLQSSwiGetAllCarrierImages
qaGobiApiFms.h, 1698
SLQSSwiGetCrashAction
qaGobiApiDms.h, 1668
SLQSSwiGetCrashInfo
qaGobiApiDms.h, 1668
SLQSSwiGetDyingGaspCfg
qaGobiApiDms.h, 1669
SLQSSwiGetDyingGaspStatistics
qaGobiApiDms.h, 1669
SLQSSwiGetFSN
qaGobiApiDms.h, 1669
SLQSSwiGetFirmwareCurr
qaGobiApiFms.h, 1699
SLQSSwiGetFwUpdateStatus
qaGobiApiDms.h, 1670
SLQSSwiGetHDRPersonality
qaGobiApiNas.h, 1762
SLQSSwiGetHDRProtSubtype
qaGobiApiNas.h, 1763
SLQSSwiGetHRPDStats
qaGobiApiNas.h, 1763
SLQSSwiGetHostDevInfo
qaGobiApiDms.h, 1670
SLQSSwiGetHostDevInfoParams
qaGobiApiDms.h, 1651
SLQSSwiGetLteCQI
qaGobiApiNas.h, 1763
SLQSSwiGetLteSccRxInfo
qaGobiApiNas.h, 1764
SLQSSwiGetOSInfo
qaGobiApiDms.h, 1671
SLQSSwiGetOSInfoParams
qaGobiApiDms.h, 1651
SLQSSwiGetSMSStorage
qaGobiApiSms.h, 1813
SLQSSwiGetSerialNoExt
qaGobiApiDms.h, 1671
SLQSSwiGetSerialNoExtParams
qaGobiApiDms.h, 1652
SLQSSwiGetUSBComp
qaGobiApiDms.h, 1671
SLQSSwiNetworkDebug
qaGobiApiNas.h, 1764
SLQSSwiPSDetach
qaGobiApiNas.h, 1765
SLQSSwiSetCrashAction
qaGobiApiDms.h, 1672
SLQSSwiSetDyingGaspCfg
qaGobiApiDms.h, 1672
SLQSSwiSetHostDevInfo
qaGobiApiDms.h, 1673
SLQSSwiSetHostDevInfoParams
qaGobiApiDms.h, 1652
SLQSSwiSetOSInfo
qaGobiApiDms.h, 1673
SLQSSwiSetOSInfoParams
qaGobiApiDms.h, 1653
SLQSSwiSetUSBComp
qaGobiApiDms.h, 1673
SLQSTmdDeRegNotMitigationLvl
qaGobiApiTmd.h, 1885
SLQSTmdGetMitigationDevList
qaGobiApiTmd.h, 1886
SLQSTmdGetMitigationLvl
qaGobiApiTmd.h, 1886
SLQSTmdMitigationLvlRptCallback
qaGobiApiCbk.h, 1630
SLQSTmdRegNotMitigationLvl
qaGobiApiTmd.h, 1886
SLQSUIMAuthenticate
qaGobiApiUim.h, 1889
SLQSUIMChangePin
qaGobiApiUim.h, 1890
SLQSUIMDepersonalization
qaGobiApiUim.h, 1890
SLQSUIMEventRegister
qaGobiApiUim.h, 1891
SLQSUIMGetCardStatus
qaGobiApiUim.h, 1891
SLQSUIMGetConfiguration
qaGobiApiUim.h, 1892
SLQSUIMGetFileAttributes
qaGobiApiUim.h, 1892
SLQSUIMGetSlotsStatus
qaGobiApiUim.h, 1893
SLQSUIMGetState
qaGobiApiDms.h, 1674
SLQSUIMPowerDown
qaGobiApiUim.h, 1893
SLQSUIMPowerUp
qaGobiApiUim.h, 1894
SLQSUIMReadTransparent
qaGobiApiUim.h, 1894
SLQSUIMRefreshComplete
qaGobiApiUim.h, 1895
SLQSUIMRefreshGetLastEvent

qaGobiApiUim.h, 1895
SLQSUIMRefreshOK
qaGobiApiUim.h, 1896
SLQSUIMRefreshRegister
qaGobiApiUim.h, 1896
SLQSUIMReset
qaGobiApiUim.h, 1897
SLQSUIMSetPinProtection
qaGobiApiUim.h, 1897
SLQSUIMSetRefreshCallBack
qaGobiApiCbk.h, 1631
SLQSUIMSetStatusChangeCallBack
qaGobiApiCbk.h, 1631
SLQSUIMSwitchSlot
qaGobiApiUim.h, 1898
SLQSUIMUnblockPin
qaGobiApiUim.h, 1898
SLQSUIMVerifyPin
qaGobiApiUim.h, 1899
SLQSUpgradeFirmware9x15
qaGobiApiFms.h, 1699
SLQSVoiceALSSelectLine
qaGobiApiVoice.h, 1905
SLQSVoiceALSSetLineSwitching
qaGobiApiVoice.h, 1905
SLQSVoiceAnswerCall
qaGobiApiVoice.h, 1906
SLQSVoiceBindSubscription
qaGobiApiVoice.h, 1906
SLQSVoiceBurstDTMF
qaGobiApiVoice.h, 1907
SLQSVoiceDialCall
qaGobiApiVoice.h, 1907
SLQSVoiceEndCall
qaGobiApiVoice.h, 1908
SLQSVoiceGetAllCallInfo
qaGobiApiVoice.h, 1908
SLQSVoiceGetCLIP
qaGobiApiVoice.h, 1911
SLQSVoiceGetCLIR
qaGobiApiVoice.h, 1912
SLQSVoiceGetCNAP
qaGobiApiVoice.h, 1912
SLQSVoiceGetCOLP
qaGobiApiVoice.h, 1913
SLQSVoiceGetCOLR
qaGobiApiVoice.h, 1913
SLQSVoiceGetCallBarring
qaGobiApiVoice.h, 1909
SLQSVoiceGetCallForwardingStatus
qaGobiApiVoice.h, 1909
SLQSVoiceGetCallInfo
qaGobiApiVoice.h, 1910
SLQSVoiceGetCallWaiting
qaGobiApiVoice.h, 1910
SLQSVoiceGetConfig
qaGobiApiVoice.h, 1914
SLQSVoiceIndicationRegister
qaGobiApiVoice.h, 1914
SLQSVoiceInfoRecCallback
qaGobiApiCbk.h, 1631
SLQSVoiceManageCalls
qaGobiApiVoice.h, 1915
SLQSVoiceOrigUSSDNoWait
qaGobiApiVoice.h, 1915
SLQSVoiceSendFlash
qaGobiApiVoice.h, 1916
SLQSVoiceSetAllCallStatusCallBack
qaGobiApiCbk.h, 1632
SLQSVoiceSetCallBarringPassword
qaGobiApiVoice.h, 1916
SLQSVoiceSetConfig
qaGobiApiVoice.h, 1917
SLQSVoiceSetDTMFEventCallBack
qaGobiApiCbk.h, 1632
SLQSVoiceSetOTASPStatusCallBack
qaGobiApiCbk.h, 1633
SLQSVoiceSetPreferredPrivacy
qaGobiApiVoice.h, 1917
SLQSVoiceSetPrivacyChangeCallBack
qaGobiApiCbk.h, 1633
SLQSVoiceSetSUPSCallBack
qaGobiApiCbk.h, 1634
SLQSVoiceSetSUPSNotificationCallback
qaGobiApiCbk.h, 1634
SLQSVoiceSetSUPSService
qaGobiApiVoice.h, 1918
SLQSVoiceStartContDTMF
qaGobiApiVoice.h, 1919
SLQSVoiceStopContDTMF
qaGobiApiVoice.h, 1919
SLQSWCDMADecodeLongTextMsg
qaGobiApiSms.h, 1814
SLQSWCDMADecodeMTTextMsg
qaGobiApiSms.h, 1814
SLQSWCDMAEncodeMOTextMsg
qaGobiApiSms.h, 1814
SLQSWdsGoActive
qaGobiApiWds.h, 1961
SLQSWdsGoDormant
qaGobiApiWds.h, 1962
SLQSWdsSetEventReport
qaGobiApiWds.h, 1962
SLQSWdsSwiPDPRuntimeSettings
qaGobiApiWds.h, 1962
SLQSWmsAsyncRawSendCallBack
qaGobiApiCbk.h, 1634
SLQSWmsMemoryFullCallBack
qaGobiApiCbk.h, 1635
SLQSWmsMessageWaitingCallBack
qaGobiApiCbk.h, 1635
SMS_CONFIG_LEN
sms.h, 1987
SMS_MAX_SMS_ROUTES
sms.h, 1987
SMS_NUM_OF_SET

sms.h, 1987
SMSAsyncRawSend
qaGobiApiCbk.h, 1565
SMSAsyncRawSend_s, 900
alphaIDLen, 901
causeCode, 901
errorClass, 901
messageID, 901
msgDelFailureCause, 901
msgDelFailureType, 901
pAlphaID, 901
RPCause, 901
sendStatus, 901
TPCause, 901
userData, 901
SMSC_TYPE_LEN
sms.h, 1987
SMSCAddress, 901
data, 902
length, 902
sMSCAddress, 902
data, 902
length, 902
SMSCAddressInfo
qaGobiApiCbk.h, 1566
sMSCAddressInfo
sms.h, 1988
sMSCAddressTlv, 902
SMSCInfo, 903
TlvPresent, 903
SMSCInfo
sMSCAddressTlv, 903
SMSCTlv
unpack_sms_SetNewSMSCallback_ind_t, 1179
SMSEtwsMessage, 903
data, 904
length, 904
notificationType, 904
sMSEtwsMessage, 903
data, 903
length, 903
notificationType, 903
SMSEtwsMessageInfo
qaGobiApiCbk.h, 1566
sMSEtwsMessageInfo
sms.h, 1988
sMSEtwsMessageTlv, 904
EtwsMessageInfo, 904
TlvPresent, 904
SMSEtwsPlmn, 904
mobileCountryCode, 905
mobileNetworkCode, 905
sMSEtwsPlmn, 905
mobileCountryCode, 905
mobileNetworkCode, 905
SMSEtwsPlmnInfo
qaGobiApiCbk.h, 1567
sMSEtwsPlmnInfo
sms.h, 1988
SMSEventInfo
qaGobiApiCbk.h, 1567
SMSEventInfo_s, 905
pEtwsMessageInfo, 906
pEtwsPlmnInfo, 906
pMTMessageInfo, 907
pMessageModeInfo, 906
pSMSCAddressInfo, 907
pSMSOnIMSInfo, 907
pTransferRouteMTMessageInfo, 907
smsEventType, 907
SMSEventType
qaGobiApiCbk.h, 1598
SMSMTMessage, 910
messageIndex, 910
storageType, 910
sMSMTMessage, 910
messageIndex, 910
storageType, 910
SMSMTMessageInfo
qaGobiApiCbk.h, 1568
sMSMTMessageInfo
sms.h, 1988
SMSMemoryInfo, 908
messageMode, 908
storageType, 908
SMSMessageMode, 908
messageMode, 909
sMSMessageMode, 909
messageMode, 909
SMSMessageModeInfo
qaGobiApiCbk.h, 1568
sMSMessageModeInfo
sms.h, 1988
SMSOnIMS, 911
smsOnIMS, 911
sMSOnIMS, 911
smsOnIMS, 911
SMSOnIMSInfo
qaGobiApiCbk.h, 1568
sMSOnIMSInfo
sms.h, 1988
sMSOnIMSTlv, 911
IMSInfo, 912
TlvPresent, 912
SMSSupport
pack_dms_SetCustFeature_t, 594
unpack_dms_GetCustFeature_t, 1028
SMSTransferRouteMTMessage, 914
ackIndicator, 914
data, 914
format, 914
length, 914
transactionID, 914
sMSTransferRouteMTMessage, 914
ackIndicator, 915
data, 915

format, 915
length, 915
transactionID, 915
SMSTransferRouteMTMessageInfo
qaGobiApiCbk.h, 1568
sMSTransferRouteMTMessageInfo
sms.h, 1989
SNR
nas_NetworkStatEVDO, 478
NetworkStatEVDO, 576
sNonIntraSearch
LTEInfoIntrafreq, 395
nas_LTEInfoIntrafreq, 460
SO
nas_NetworkStat1x, 476
NetworkStat1x, 575
SOMask
CurrNetworkInfo, 202
currNetworkInfo, 203
wds_currNetworkInfo, 1366
SPC_SIZE
dms.h, 1411
sPhyCaAggPcellInfo
nasGetLTECphyCa, 533
QmiCbkNasLTECphyCaInfo, 790
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1104
sPhyCaAggScellDlBw
nasGetLTECphyCa, 533
QmiCbkNasLTECphyCaInfo, 790
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1104
sPhyCaAggScellIndType
nasGetLTECphyCa, 533
QmiCbkNasLTECphyCaInfo, 790
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1104
sPhyCaAggScellIndex
nasGetLTECphyCa, 533
QmiCbkNasLTECphyCaInfo, 790
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1104
sPhyCaAggScellInfo
nasGetLTECphyCa, 533
QmiCbkNasLTECphyCaInfo, 790
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1104
sQosFlowStat, 915
bearerId, 916
tx_bytes, 916
tx_bytes_drp, 916
tx_pkt, 916
tx_pkt_drp, 916
sQosStat, 916
apnId, 917
numQosFlow, 917
qosFlow, 917
total_rx_bytes, 917
total_rx_pkt, 917
total_tx_bytes, 917
total_tx_bytes_drp, 917
total_tx_pkt, 917
total_tx_pkt_drp, 917
SSInfo
unpack_nas_SetServingSystemCallback_ind_t, 1106
sSLQSSignalStrengthsInfo
nas_SLQSSignalStrengthsTlv, 508
SSTlv
unpack_nas_SetEventReportInd_t, 1104
SUPSInfo, 921
isModByCC, 922
svcType, 922
SUPSInformation
voiceSUPSInfo, 1352
SUPSType
pack_voice_SLQSVoiceManageCalls_t, 697
voiceManageCallsReq, 1335
SV, 922
id, 923
mask, 923
system, 923
SVInfo, 923
len, 923
pSV, 924
SWI Audio Service(SWIAUDIO), 56
SWI Open Mobile Alliance Service (SWIOMA), 49
SWI_API
SwiDataTypes.h, 2007
SWI_STRUCT_CarrierImage, 924
m_FwBuildId, 925
m_FwImageId, 925
m_PriBuildId, 925
m_PriImageId, 925
m_nCarrierId, 925
m_nFolderId, 925
m_nStorage, 925
SWIWWANCMAPI.h, 2018
samplesPerBatch
accelAcceptReady_s, 113
accelTempAcceptReady_s, 114
gyroAcceptReady_s, 297
gyroTempAcceptReady_s, 297
sar.h, 1982
pack_sar_SLQSGetRfSarState, 1982
pack_sar_SLQSSetRfSarState, 1983
unpack_sar_SLQSGetRfSarState, 1983
unpack_sar_SLQSSetRfSarState, 1983
satelliteInfo, 825
azimuth, 827
elevation, 827
gnssSvId, 827
healthStatus, 827
snr, 827
svInfoMask, 827
svListLen, 827

svStatus, 827
system, 827
validMask, 827
SaveSMS
qaGobiApiSms.h, 1799
sbas_almanac_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
sbas_ephemeris_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
sbas_health_sv_msk
GPSStateInfo, 289
unpack_pds_SLQSGetGPSStateInfo_t, 1154
sbas_visible_sv_msk
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
SccRxInfo, 828
numInstances, 828
rsrq, 828
sigInfo, 828
snr, 828
TlvPresent, 828
scell_idx
nas_PhyCaAggScellIndex, 483
NASPhyCaAggScellIndex, 552
PhyCaAggScellIndex, 739
scell_state
nas_PhyCaAggScellIndType, 483
nas_PhyCaAggScellInfo, 486
NASPhyCaAggScellIndType, 553
NASPhyCaAggScellInfo, 554
PhyCaAggScellIndType, 740
PhyCaAggScellInfo, 742
screeningInd
connectNumInfo, 188
voice_connectNumInfo, 1290
sduErrorRatio
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
seDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
seDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
sePCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
sePCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 1258
secActivate
fileAttributes, 243
uim_fileAttributes, 978
secActivateMask
fileAttributes, 243
uim_fileAttributes, 978
secChA
CDMAChannel, 162
nas_CDMAChannel, 419
secChB
CDMAChannel, 163
nas_CDMAChannel, 419
secDeactivate
fileAttributes, 243
uim_fileAttributes, 978
secDeactivateMask
fileAttributes, 243
uim_fileAttributes, 978
secIncrease
fileAttributes, 244
uim_fileAttributes, 978
secIncreaseMask
fileAttributes, 244
uim_fileAttributes, 978
SecProt
nas_protocolSubtypeElement, 491
protocolSubtypeElement, 763
secRead
fileAttributes, 244
uim_fileAttributes, 978
secReadMask
fileAttributes, 244
uim_fileAttributes, 978
secWrite
fileAttributes, 244
uim_fileAttributes, 978
secWriteMask
fileAttributes, 244
uim_fileAttributes, 978
secdns
unpack_wds_GetDefaultProfile_t, 1232
secdnsv6
unpack_wds_GetDefaultProfile_t, 1232
second
nas_timeInfo, 516
nas_UniversalTime, 522
timeInfo, 955
UniversalTime, 1023
secondaryDNS
pack_wds_SetDefaultProfile_t, 710
SecondaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 1250
SecondaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 1250
secondaryHA
unpack_wds_GetMobileIPProfile_t, 1236
SectorIDLen
nas_NetworkStatEVDO, 478
NetworkStatEVDO, 576
selNetwork
nas_servSystem, 504
servSystem, 837

selected
BroadcastConfig, 142
CDMABroadcastConfig, 162
sms_BroadcastConfig, 887
sms_CDMABroadcastConfig, 888
selectedNetwork
NASServingSystemInfo, 562
ServingSystemInfo, 836
selection
pack_swioma_SLQSOMADMSendSelection_t, 672
SendSMS
qaGobiApiSms.h, 1800
sendStatus
SMSAsyncRawSend_s, 901
sensorData, 828
flags, 830
sensorDataLen, 830
timeOfFirstSample, 830
timeOffset, 830
xAxis, 830
yAxis, 830
zAxis, 830
sensorData_t, 830
flags, 831
sensorDataLen, 831
timeOfFirstSample, 831
timeOffset, 831
xAxis, 831
yAxis, 831
zAxis, 831
sensorDataLen
sensorData, 830
sensorData_t, 831
sensorDataUsage
qaGobiApiCbk.h, 1564
sensorDataUsage_s, 832
aidingIndicatorMask, 832
usageMask, 832
serialNumbersInfo, 832
esnSize, 833
imeiSize, 833
imeiSvnSize, 833
meidSize, 833
pESNString, 833
pIMEIString, 833
pImeiSvnString, 833
pMEIDString, 833
qaGobiApiDms.h, 1650
servSystem, 836
csAttachState, 837
numRadioInterfaces, 837
psAttachState, 837
radioInterface, 837
regState, 837
selNetwork, 837
ServerAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 1250
serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 611
serverStatus
QmiCbkLocGetServerInd, 778
unpack_loc_GetServer_Ind_t, 1080
serverType
LocGetServerReq, 380
pack_loc_SLQSLOCGetServer_t, 611
QmiCbkLocGetServerInd, 778
unpack_loc_GetServer_Ind_t, 1080
serviceCategory
CDMABroadcastConfig, 162
sms_CDMABroadcastConfig, 888
serviceClassInformation
qaGobiApiVoice.h, 1903
serviceDomain
nas_RejectReasonTlv, 494
nasNetworkReject, 548
serviceProviderName, 834
displayCondition, 834
spn, 834
spnLength, 834
servingCellId
LTEInfoIntrafreq, 395
nas_LTEInfoIntrafreq, 460
ServingSystem
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1114
ServingSystemInfo, 834
csAttachState, 835
hdrPersonality, 835
psAttachState, 835
radioInterfaceList, 835
radioInterfaceNo, 836
registrationState, 836
selectedNetwork, 836
sessionControl
pack_pds_StartPDSTrackingSessionExt_t, 657
sessionEndReason
_packetSrvStatus, 70
slqsSessionStateInfo, 879
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1252
SessionId
LOCStartReq, 388
pack_loc_Start_t, 621
pack_loc_Stop_t, 621
sessionId
LOCStopReq, 388
QmiCbkLocPositionReportInd, 787
ssdatasession_params, 921
unpack_loc_PositionRpt_Ind_t, 1087
sessionInfo, 837
omaDmConfig, 838
omaDmFota, 838
omaDmNotifications, 838
pack_uim_ChangePin_t, 675
pack_uim_ReadTransparent_t, 676

pack_uim_SetPinProtection_t, 677
pack_uim_SLQSUIMAuthenticate_t, 678
pack_uim_SLQSUIMGetFileAttributes_t, 680
pack_uim_SLQSUIMRefreshComplete_t, 681
pack_uim_SLQSUIMRefreshGetLastEvent_t, 682
pack_uim_SLQSUIMRefreshOK_t, 682
pack_uim_SLQSUIMRefreshRegister_t, 683
pack_uim_UnblockPin_t, 684
pack_uim_VerifyPin_t, 686
sessionInfoTlv, 838
sessionInfoTlvExt, 839
UIMAuthenticateReq, 991
UIMChangePinReq, 992
UIMGetFileAttributesReq, 996
UIMReadTransparentReq, 1000
UIMRefreshCompleteReq, 1002
UIMRefreshGetLastEventReq, 1004
UIMRefreshOKReq, 1005
UIMRefreshRegisterReq, 1005
UIMSetPinProtectionReq, 1007
UIMUnblockPinReq, 1011
UIMVerifyPinReq, 1012
SessionInfoConfig
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1188
sessionInfoExt, 838
omaDmConfig, 838
omaDmFota, 838
SessionInfoFota
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1188
SessionInfoNotification
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1188
sessionInfoTlv, 838
sessionInfo, 838
sessionType, 838
TlvPresent, 838
sessionInfoTlvExt, 839
sessionInfo, 839
sessionType, 839
TlvPresent, 839
sessionInformation
qaGobiApiCbk.h, 1565
sessionInformationExt
qaGobiApiCbk.h, 1565
sessionOperation
pack_pds_StartPDSTrackingSessionExt_t, 657
sessionServerOption
pack_pds_StartPDSTrackingSessionExt_t, 657
SessionState
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
SessionStatus
unpack_pds_SetEventReport_Ind_t, 1149
sessionStatus
omaDmNotificationsTlv, 588
QmiCbkLocPositionReportInd, 787
unpack_loc_PositionRpt_Ind_t, 1087
unpack_omaDmNotificationsTlv_t, 1143
SessionType
_SwiAvmsEventInd, 100
pack_swioma_SLQSOMADMGetSessionInfo_t, 672
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
sessionType
omaDmFotaTlv, 585
pack_pds_StartPDSTrackingSessionExt_t, 657
pack_swioma_SLQSOMADMCancelSession_t, 671
pack_swioma_SLQSOMADMStartSession_t, 674
sessionInfoTlv, 838
sessionInfoTlvExt, 839
uim_refreshevent, 983
uim_sessionInformation, 985
uim_UIMSessionInformation, 989
UIMRefreshEvent, 1003
UIMSessionInformation, 1006
unpack_omaDmFotaTlv_t, 1142
set_fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 671
SwiLocSetAutoStartReq, 928
set_fix_type
pack_swiloc_SwiLocSetAutoStart_t, 671
SwiLocSetAutoStartReq, 929
set_function
pack_swiloc_SwiLocSetAutoStart_t, 671
SwiLocSetAutoStartReq, 929
set_max_dist
pack_swiloc_SwiLocSetAutoStart_t, 671
SwiLocSetAutoStartReq, 929
set_max_time
pack_swiloc_SwiLocSetAutoStart_t, 671
SwiLocSetAutoStartReq, 929
SetACCOLC
qaGobiApiNas.h, 1746
SetActivationStatusCallback
qaGobiApiCbk.h, 1599
SetActiveMobileIPProfile
qaGobiApiWds.h, 1940
SetAudioPathConfigReq, 839
pCodecSTGain, 840
pDTMFTXGain, 840
pECMode, 841
pNSEnable, 841
pRXAGCList, 841
pRXAVCAGCSwitch, 841
pRXAVCList, 841
pRXPCMIIRFltr, 841
pTXAGCList, 841
pTXAVCSwitch, 841
pTXGain, 841
pTXPCMIIRFltr, 841
Profile, 841
SetAudioProfileReq, 841

EarMute, 842
Generator, 842
MicMute, 842
Profile, 842
Volume, 842
SetAudioVolTLBConfigReq, 842
Generator, 843
Item, 843
Profile, 843
VolValue, 843
Volume, 843
SetAudioVolTLBConfigResp, 843
ResCode, 843
SetAutoconnect
qaGobiApiWds.h, 1940
SetCATEventCallback
qaGobiApiCbk.h, 1599
SetCDMANetworkParameters
qaGobiApiNas.h, 1747
setCustomSettingV2, 844
cust_id, 844
cust_value, 844
value_length, 844
SetDataCapabilitiesCallback
qaGobiApiCbk.h, 1600
SetDefaultProfile
qaGobiApiWds.h, 1940
SetDefaultProfileLTE
qaGobiApiWds.h, 1942
SetDefaultProfileLTEV2
qaGobiApiWds.h, 1943
SetDefaultProfileNum
qaGobiApiWds.h, 1945
SetDeviceStateChangeCbk
qaGobiApiCbk.h, 1601
setDyingGaspCfg, 844
pDestSMSContent, 845
pDestSMSNum, 845
SetFwDldCompletionCbk
qaGobiApiCbk.h, 1601
SetGPSCallback
qaGobiApiCbk.h, 1602
SetIMSSMSConfigReq, 845
pPhoneCtxtURI, 845
pPhoneCtxtURILen, 845
pSMSFormat, 845
pSMSOverIPNwInd, 845
SetIMSSMSConfigResp, 845
pSettingResp, 846
SetIMSUserConfigReq, 846
pIMSDomain, 846
pIMSDomainLen, 846
SetIMSUserConfigResp, 846
pSettingResp, 847
SetIMSVoIPConfigReq, 847
pAmrMode, 849
pAmrOctetAligned, 849
pAmrWBMode, 849
pAmrWBOctetAligned, 849
pAmrWbEnable, 849
pMinSessionExpiryTimer, 849
pRTPRTCPInactTimer, 849
pRingBackTimer, 849
pRingingTimer, 849
pScrAmrEnable, 849
pScrAmrWbEnable, 849
pSessionExpiryTimer, 849
SetIMSVoIPConfigResp, 849
pSettingResp, 850
SetImagesPreference
qaGobiApiFms.h, 1691
setIndicationRegReq
qaGobiApiSms.h, 1798
SetLURejectCallback
qaGobiApiCbk.h, 1606
SetLocBestAvailPosCallback
qaGobiApiCbk.h, 1602
SetLocCradleMountCallback
qaGobiApiCbk.h, 1602
SetLocDeleteAssistDataCallback
qaGobiApiCbk.h, 1603
SetLocEngineStateCallback
qaGobiApiCbk.h, 1603
SetLocEventMaskNMEACallback
qaGobiApiCbk.h, 1603
SetLocEventPositionCallback
qaGobiApiCbk.h, 1603
SetLocEventTimeSyncCallback
qaGobiApiCbk.h, 1604
SetLocGetServerCallback
qaGobiApiCbk.h, 1604
SetLocGnssSvInfoCallback
qaGobiApiCbk.h, 1604
SetLocInjectSensorDataCallback
qaGobiApiCbk.h, 1605
SetLocInjectTimeCallback
qaGobiApiCbk.h, 1605
SetLocOpModeCallback
qaGobiApiCbk.h, 1605
SetLocSensorStreamingCallback
qaGobiApiCbk.h, 1605
SetLocSetExtPowerConfigCallback
qaGobiApiCbk.h, 1606
SetM2MAVMuteReq, 853
EarMute, 853
MicMute, 853
pCwtMute, 853
Profile, 854
SetM2MAudioAVCFGReq, 850
Device, 850
PIFACEId, 850
pPCMParams, 850
Profile, 850
SetM2MAudioLPBKReq, 851
Enable, 851
SetM2MAudioProfileReq, 851

pCwtMute, 852
pEarMute, 852
pGenerator, 852
pMicMute, 852
pVolume, 852
Profile, 852
SetM2MAudioVolumeReq, 852
Generator, 853
Level, 853
Profile, 853
SetM2MSpkrGainReq, 854
Profile, 854
Value, 854
SetMobileIP
qaGobiApiWds.h, 1945
SetMobileIPParameters
qaGobiApiWds.h, 1946
SetMobileIPProfile
qaGobiApiWds.h, 1947
SetMobileIPStatusCallback
qaGobiApiCbk.h, 1606
SetNMEACallback
qaGobiApiCbk.h, 1608
SetNasLTECphyCaIndCallback
qaGobiApiCbk.h, 1607
SetNetChangeCbk
qaGobiApiCbk.h, 1607
SetNetworkPreference
qaGobiApiNas.h, 1748
SetNewSMSCallback
qaGobiApiCbk.h, 1608
SetOMADMStateCallback
qaGobiApiCbk.h, 1609
SetPDSDefaults
qaGobiApiPds.h, 1775
SetPDSState
qaGobiApiPds.h, 1776
SetPDSStateCallback
qaGobiApiCbk.h, 1609
setPINProtection, 854
pinID, 855
pinLength, 855
pinOperation, 855
pinValue, 855
SetPortAutomaticTracking
qaGobiApiPds.h, 1776
SetPower
qaGobiApiDms.h, 1662
SetPowerCallback
qaGobiApiCbk.h, 1609
SetRFInfoCallback
qaGobiApiCbk.h, 1610
SetRMTransferStatisticsCallback
qaGobiApiCbk.h, 1610
SetRankIndicatorCallback
qaGobiApiCbk.h, 1610
SetRegMgrConfigReq, 855
pCSCFPortName, 856
pCSCFPortNameLen, 856
pIMSTestMode, 856
pPriCSCFPort, 856
SetRegMgrConfigResp, 856
pSettingResp, 856
SetRoamingIndicatorCallback
qaGobiApiCbk.h, 1610
SetSDKImagePath
qaGobiApiDcs.h, 1640
SetSIPConfigReq, 861
pSIPLocalPort, 862
pSigCompEnabled, 862
pSubscribeTimer, 862
pTimerSIPReg, 862
pTimerT1, 862
pTimerT2, 862
pTimerTf, 862
SetSIPConfigResp, 862
pSettingResp, 863
SetSLQSOMADMAlertCallback
qaGobiApiCbk.h, 1612
SetSLQSOMADMAlertCallbackExt
qaGobiApiCbk.h, 1612
SetSMSCAddress
qaGobiApiSms.h, 1801
SetSMSWake
qaGobiApiRms.h, 1790
SetServiceAutomaticTracking
qaGobiApiPds.h, 1777
SetSignalStrengthCallback
qaGobiApiCbk.h, 1611
setSignalStrengthInfo, 856
pCDMAECIODelta, 860
pCDMAECIOThresh, 860
pCDMARSSIDelta, 860
pCDMARSSIThresh, 860
pGSMRSSIDelta, 860
pGSMRSSIThresh, 860
pHDRECIODelta, 860
pHDRECIOThresh, 860
pHDRIODelta, 860
pHDRIOThresh, 860
pHDRRSSIDelta, 860
pHDRRSSIThresh, 860
pHDRSINRDelta, 860
pHDRSINRThresh, 860
pLTERSRPDelta, 860
pLTERSRPThresh, 860
pLTERSRQDelta, 860
pLTERSRQThresh, 860
pLTERSSIDelta, 861
pLTERSSIThresh, 861
pLTESNRDelta, 861
pLTESNRThresh, 861
pLTESigRptConfig, 861
pTDSCDMAECIODelta, 861
pTDSCDMAECIOThresh, 861
pTDSCDMARSCPDelta, 861

pTDSCDMARSCPThresh, 861
pTDSCDMARSSIDelta, 861
pTDSCDMARSSIThresh, 861
pTDSCDMASINRDelta, 861
pTDSCDMASINRThresh, 861
pWCDMAECIODelta, 861
pWCDMAECIOThresh, 861
pWCDMARSSIDelta, 861
pWCDMARSSIThresh, 861
SetUSSDNoWaitIndicationCallback
qaGobiApiCbk.h, 1613
SetUSSDNotificationCallback
qaGobiApiCbk.h, 1613
SetUSSDReleaseCallback
qaGobiApiCbk.h, 1613
SetUimSlotStatusChangeCallback
qaGobiApiCbk.h, 1612
SetXTRAAutomaticDownload
qaGobiApiPds.h, 1777
SetXTRANetwork
qaGobiApiPds.h, 1778
SetupEventList
CatEventListTlv, 160
Severity
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
severity
omaDmFotaTlv, 585
unpack_omaDmFotaTlv_t, 1142
Short Message Service (SMS), 42
shortName
nas_PLMNNetworkNameData, 489
nasPLMNNameResp, 557
PLMNNetworkNameData, 746
unpack_nas_SLQSGetPLMNName_t, 1110
shortNameCI
nasPLMNNameResp, 557
unpack_nas_SLQSGetPLMNName_t, 1110
shortNameEn
nasPLMNNameResp, 557
unpack_nas_SLQSGetPLMNName_t, 1110
shortNameLen
nas_PLMNNetworkNameData, 489
nasPLMNNameResp, 557
PLMNNetworkNameData, 746
unpack_nas_SLQSGetPLMNName_t, 1110
shortNameSB
nasPLMNNameResp, 557
unpack_nas_SLQSGetPLMNName_t, 1110
shortNameSpareBits
nas_PLMNNetworkNameData, 489
PLMNNetworkNameData, 747
sid
CDMAInfo, 164
nas_CDMAInfo, 421
nas_sidNid, 505
sidNid, 864
unpack_nas_GetHomeNetwork_t, 1096
SidNid
homeSIDNID, 307
nas_homeSIDNID, 451
sidNid, 863
nid, 864
sid, 864
SigInd
LibPackUMTSReqQoSSigInd, 361
UMTSReqQoSSigInd, 1021
sigInfo, 864
nas_SccRxInfo, 502
pECIOThresh, 865
pHDRSINRThresh, 865
pIOThresh, 865
pLTESNRThresh, 865
pLTESigRptCfg, 865
pRSRPThresh, 865
pRSRQThresh, 865
pRSSIThresh, 865
pTDSCDMASINRCONFThresh, 865
SccRxInfo, 828
signal
signalInfo, 866
signalInfo, 865
alertPitch, 866
signal, 866
signalType, 866
signalStrength
nas_SignalStrengthTlv, 505
SignalStrengthDataType, 866
thresholds, 866
thresholdsSize, 866
signalStrengthReqMask
slqsSignalStrengthInfo, 881
unpack_nas_SLQSGetSignalStrength_t, 1116
signalType
signalInfo, 866
SimCapability
unpack_dms_GetDeviceCap_t, 1030
simCapability
unpack_dms_GetDeviceCapabilities_t, 1031
sinr
HDRSSInfo, 303
hdrSSInfo, 304
nas_SLQSSignalStrengthsInformation, 508
slqsSignalStrengthInfo, 882
SLQSSignalStrengthsInformation, 885
TDSCDMASigInfoExt, 948
tdscdmaSigInfoExt, 947
unpack_nas_SLQSGetSignalStrength_t, 1116
sinrDelta
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
sinrThresholdList
nas_SLQSSignalStrengthsIndReq, 507
SLQSSignalStrengthsIndReq, 883
sinrThresholdListLen
nas_SLQSSignalStrengthsIndReq, 507

SLQSSignalStrengthsIndReq, 883
sku_str
slqsfwinfo_s, 873
unpack_dms_GetFirmwareInfo_t, 1035
slot
pack_uim_SLQSUIMPowerDown_t, 680
pack_uim_SLQSUIMPowerUp_t, 681
UIMPowerDownReq, 999
UIMPowerUpReq, 999
slot_t, 866
bICCID, 867
bICCIDLength, 867
bLogicalSlot, 867
uPhyCardStatus, 867
uPhySlotStatus, 867
slotInf, 867
AppStatus, 868
cardState, 868
errorState, 868
numApp, 868
upinRetries, 868
upinState, 869
upukRetries, 869
SlotInfo
cardStatus, 156
uim_cardStatus, 972
slotInfo, 869
AppStatus, 870
cardState, 870
errorState, 870
numApp, 870
upinRetries, 870
upinState, 870
upukRetries, 870
slots_t, 870
uimSlotStatus, 870
slotsstatusChange
UIMSlotStatusChangeInfo, 1009
unpack_uim_SetUimSlotStatusChangeCallback_-ind_t, 1196
slqs3GPPConfigItem
qaGobiApiWds.h, 1925
SlqsNas3GppNetworkInfo, 873
Description, 874
Forbidden, 874
InUse, 874
MCC, 874
MNC, 874
Preferred, 874
Roaming, 874
SlqsNas3GppNetworkRAT
qaGobiApiNas.h, 1723
SlqsNasPcsDigit, 874
includes_pcs_digit, 875
MCC, 875
MNC, 875
slqsNetworkScanInfo
qaGobiApiNas.h, 1724
SlqsProfile3GPP
unpackWdsProfileParam, 1259
wds_profileInfo, 1377
WdsProfileParam, 1394
SlqsProfile3GPP2
unpackWdsProfileParam, 1259
wds_profileInfo, 1377
WdsProfileParam, 1394
slqsSessionStateInfo, 878
pQmiInterfaceInfo, 879
reconfiguration_required, 879
sessionEndReason, 879
state, 879
slqsSignalStrengthInfo, 879
ecioList, 881
ecioListLen, 881
errorRateList, 881
errorRateListLen, 881
Io, 881
ltersrp, 881
ltesnr, 881
rsrqInfo, 881
rxSignalStrengthList, 881
rxSignalStrengthListLen, 881
signalStrengthReqMask, 881
sinr, 882
slqsWdsEventInfo, 885
pDataBearer, 886
pDormancyStatus, 886
pPacketsCountRX, 886
pPacketsCountTX, 886
pQmiInterfaceInfo, 886
pTotalBytesRX, 887
pTotalBytesTX, 887
slqsautoconnect, 870
acroamsetting, 871
acsetting, 871
action, 871
slqsfwinfo_s, 872
appversion_str, 873
bootversion_str, 873
carrier_str, 873
cur_carr_name, 873
cur_carr_rev, 873
modelid_str, 873
packageid_str, 873
priversion_str, 873
sku_str, 873
slqssendasyncsmsparams_s, 875
messageFormat, 877
messageSize, 877
pFollowOnDC, 877
pForceOnDC, 877
pLinktimer, 877
pMessage, 877
pRetryMessage, 877
pRetryMessageId, 877
pServiceOption, 877

pSmsOnIms, 877
pUserData, 877
slqssendsmsparams_s, 877
messageFailureCode, 878
messageFormat, 878
messageID, 878
messageSize, 878
pLinktimer, 878
pMessage, 878
pSmsOnIms, 878
sms.h
LITEQMI_QMI_CBK_PARAM_NOCHANGE, 1989
LITEQMI_QMI_CBK_PARAM_RESET, 1989
LITEQMI_QMI_CBK_PARAM_SET, 1989
sms.h, 1984
eqmiCbkSetStatus, 1989
MAX_MSE_TWS_MSG, 1987
MAX_SMS_LIST_SIZE, 1987
pack_sms_GetSMSCAddress, 1989
pack_sms_SLQSDeleteSMS, 1991
pack_sms_SLQSGetIndicationRegister, 1991
pack_sms_SLQSGetMessageWaiting, 1992
pack_sms_SLQSGetSMS, 1992
pack_sms_SLQSGetSMSList, 1993
pack_sms_SLQSGetSmsBroadcastConfig, 1992
pack_sms_SLQSGetTransLayerInfo, 1993
pack_sms_SLQSGetTransNWRegInfo, 1994
pack_sms_SLQSModifySMSStatus, 1994
pack_sms_SLQSSendAsyncSMS, 1994
pack_sms_SLQSSetIndicationRegister, 1995
pack_sms_SLQSSetSmsBroadcastActivation, 1995
pack_sms_SLQSSetSmsBroadcastConfig, 1995
pack_sms_SLQSSetSmsStorage, 1996
pack_sms_SLQSSmsGetMaxStorageSize, 1996
pack_sms_SLQSSmsGetMessageProtocol, 1996
pack_sms_SLQSSmsSetRoutes, 1997
pack_sms_SLQSSwiGetSMSStorage, 1997
pack_sms_SaveSMS, 1989
pack_sms_SendSMS, 1990
pack_sms_SetNewSMSCallback, 1990
pack_sms_SetSMSCAddress, 1991
SMS_CONFIG_LEN, 1987
SMS_MAX_SMS_ROUTES, 1987
SMS_NUM_OF_SET, 1987
SMSC_TYPE_LEN, 1987
sMSCAddressInfo, 1988
sMSEtwsMessageInfo, 1988
sMSEtwsPlmnInfo, 1988
sMSMTMessageInfo, 1988
sMSMessageModeInfo, 1988
sMSOnIMSInfo, 1988
sMSTransferRouteMTMessageInfo, 1989
unpack_sms_GetSMSCAddress, 1998
unpack_sms_SLQSDeleteSMS, 2000
unpack_sms_SLQSGetIndicationRegister, 2000
unpack_sms_SLQSGetMessageWaiting, 2000
unpack_sms_SLQSGetSMS, 2001
unpack_sms_SLQSGetSMSList, 2001
unpack_sms_SLQSGetSmsBroadcastConfig, 2001
unpack_sms_SLQSGetTransLayerInfo, 2002
unpack_sms_SLQSGetTransNWRegInfo, 2002
unpack_sms_SLQSModifySMSStatus, 2002
unpack_sms_SLQSSendAsyncSMS, 2003
unpack_sms_SLQSSetIndicationRegister, 2003
unpack_sms_SLQSSetSmsBroadcastActivation, 2003
unpack_sms_SLQSSetSmsBroadcastConfig, 2004
unpack_sms_SLQSSetSmsStorage, 2004
unpack_sms_SLQSSmsGetMaxStorageSize, 2004
unpack_sms_SLQSSmsGetMessageProtocol, 2005
unpack_sms_SLQSSmsSetRoutes, 2005
unpack_sms_SLQSSwiGetSMSStorage, 2005
unpack_sms_SLQSWmsMemoryFullCallBack_ind, 2006
unpack_sms_SaveSMS, 1998
unpack_sms_SendSMS, 1998
unpack_sms_SetNewSMSCallback, 1999
unpack_sms_SetNewSMSCallback_ind, 1999
unpack_sms_SetSMSCAddress, 1999
sms_BroadcastConfig, 887
fromServiceId, 887
selected, 887
toServiceId, 887
sms_CDMABroadcastConfig, 887
language, 888
selected, 888
serviceCategory, 888
sms_getIndicationReg, 888
pRegCallStatInfoEvt, 889
pRegTransLayerInfoEvt, 889
pRegTransNWRegInfoEvt, 889
sms_getMsgWaitingInfo, 889
msgWaitInfo, 889
numInstances, 889
sms_getTransLayerInfo, 889
pRegInd, 890
pTransLayerInfo, 890
sms_getTransNWRegInfo, 890
pRegStatus, 890
sms_maxStorageSizeReq, 891
pMessageMode, 891
storageType, 891
sms_maxStorageSizeResp, 891
freeSlots, 892
maxStorageSize, 892
sms_messageWaitingInfoContent, 892
activeInd, 892
msgCount, 892
msgType, 892
sms_msgProtocolResp, 893
msgProtocol, 893
sms_qaQmi3GPP2BroadcastCfgInfo, 893
activated_ind, 894

CDMABroadcastConfig, 894
num_instances, 894
sms_qaQmi3GPPBroadcastCfgInfo, 894
activated_ind, 894
broadcastConfig, 894
num_instances, 894
sms_routeEntry, 894
messageClass, 895
messageType, 895
receiptAction, 895
routeStorage, 895
sms_sendAsyncsmsParams, 895
messageFormat, 897
messageSize, 897
pFollowOnDC, 897
pForceOnDC, 897
pLinktimer, 897
pMessage, 897
pRetryMessage, 897
pRetryMessageId, 897
pServiceOption, 897
pSmsOnIms, 897
pUserData, 897
sms_setIndicationReg, 898
pRegCallStatInfoEvt, 898
pRegTransLayerInfoEvt, 898
pRegTransNWRegInfoEvt, 898
sms_setRoutesReq, 898
numOfRoutes, 899
pTransferStatusReport, 899
routeList, 899
sms_transLayerInfo, 899
TransCap, 899
TransType, 900
smsEventType
SMSEventInfo_s, 907
smsMaxStorageSizeReq, 907
pMessageMode, 907
storageType, 907
smsMaxStorageSizeResp, 907
freeSlots, 908
maxStorageSize, 908
smsMsgprotocolResp, 909
msgProtocol, 910
smsOnIMS
SMSOnIMS, 911
sMSOnIMS, 911
smsRouteEntry, 912
messageClass, 913
messageType, 913
receiptAction, 913
routeStorage, 913
smsSetRoutesReq, 913
numOfRoutes, 913
pTransferStatusReport, 913
routeList, 913
smsStorage
pack_sms_SLQSSetSmsStorage_t, 668
snr
loc_satelliteInfo, 371
LTESSInfo, 405
lteSSInfo, 405
nas_SccRxInfo, 502
satelliteInfo, 827
SccRxInfo, 828
snrlevel
lteSnrinformation, 403
nas_lteSnrinformation, 466
soMask
DataBearerTech, 211
dataBearerTechnology, 213
qmiWDSDataBearerTechnology, 796
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
Source
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
source
_getResetInfoNotification, 66
altitudeSrcInfo, 120
altSrcInfo_t, 121
dmsSwiGetResetInfo, 230
unpack_dms_GetNetworkTime_t, 1040
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1051
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1052
sourceIPMask
LibPackTFTIDParams, 358
TFTIDParams, 953
SourceLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
spc
pack_dms_ResetToFactoryDefaults_t, 591
pack_nas_SetACCOLC_t, 622
pack_wds_SetMobileIPProfile_t, 714
Specific Absorption Rate (SAR), 48
spn
nas_serviceProviderName, 503
nasPLMNNameResp, 557
serviceProviderName, 834
unpack_nas_SLQSGetPLMNName_t, 1110
spnEncoding
nasPLMNNameResp, 557
unpack_nas_SLQSGetPLMNName_t, 1110
spnLength
nas_serviceProviderName, 503
nasPLMNNameResp, 558
serviceProviderName, 834
unpack_nas_SLQSGetPLMNName_t, 1110
srcPortRangeEnd
LibPackTFTIDParams, 358
TFTIDParams, 953
srcPortRangeStart
LibPackTFTIDParams, 358
TFTIDParams, 953

srvCapability
detailSvcInfo, 220
nas_detailSvcInfo, 434
nas_sysInfoCommon, 511
sysInfoCommon, 944
srvCapabilityValid
nas_sysInfoCommon, 511
sysInfoCommon, 944
srvDomain
nas_sysInfoCommon, 511
sysInfoCommon, 944
SrvDomainPref
NASServDomainPrefTlv, 561
srvDomainValid
nas_sysInfoCommon, 511
sysInfoCommon, 944
srvOption
arrSvcOption, 134
voice_arrSvcOption, 1276
srvStatus
detailSvcInfo, 221
GSMSrvStatusInfo, 293
nas_detailSvcInfo, 434
nas_GSMSrvStatusInfo, 443
nas_SrvStatusInfo, 509
SrvStatusInfo, 918
SrvStatusInfo, 917
isPrefDataPath, 918
srvStatus, 918
srxlev
cellParams, 176
gsmCellInfo, 291
nas_cellParams, 426
nas_gsmCellInfo, 442
nas_umtsLTENbrCell, 521
nas_wcdmaCellInfo, 523
umtsLTENbrCell, 1015
wcdmaCellInfo, 1355
ssdatasession_params, 918
action, 920
failureReason, 920
failureReasonv4, 920
failureReasonv6, 920
instanceId, 920
ipfamily, 920
pAuthentication, 920
pPassword, 920
pProfileId3GPP, 920
pProfileId3GPP2, 920
pTechnology, 920
pUsername, 921
rcv4, 921
rcv6, 921
sessionId, 921
v4sessionId, 921
v6sessionId, 921
verbFailReason, 921
verbFailReasonType, 921
stage
uim_refreshevent, 983
UIMRefreshEvent, 1003
StartPDSTrackingSessionExt
qaGobiApiPds.h, 1782
State
nas_NetworkStat1x, 476
nas_NetworkStatEVDO, 478
NetworkStat1x, 575
NetworkStatEVDO, 576
state
LocSetCradleMountReq, 386
omaDmConfigTlv, 580
omaDmConfigTlvExt, 583
omaDmFotaTlv, 585
omaDmFotaTlvExt, 587
pack_loc_SLQSLOCSetCradleMountConfig_t, 619
QosFlowInfoState, 804
QosMap, 805
slqsSessionStateInfo, 879
unpack_dms_GetActivationState_t, 1024
unpack_dms_SLQSUIMGetState_t, 1064
unpack_omaDmConfigTlv_t, 1140
unpack_omaDmFotaTlv_t, 1142
unpack_qos_QosFlowInfoState_t, 1160
statmask
pack_wds_GetPacketStatus_t, 707
StatsMask
TransferStatInd, 960
transferStatInd, 960
statsMask
TrStatInd, 961
wds_TrStatInd, 1378
StatsPeriod
TransferStatInd, 960
transferStatInd, 960
statsPeriod
TrStatInd, 961
wds_TrStatInd, 1378
Status
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
status
delAssistDataStatus, 218
lteEARFCN, 390
ltePCI, 398
nas_lteEARFCN, 454
nas_ltePCI, 462
nas_wcdmaUARFCN, 528
pack_sms_SetNewSMSCallback_t, 662
QmiCbkLocBestAvailPosInd, 775
QmiCbkLocInjectPositionInd, 779
QmiCbkLocInjectUTCTimeInd, 782
QmiCbkLocSetExtPowerConfigInd, 789
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_DeleteAssistData_Ind_t, 1077
unpack_loc_SetExtPowerConfig_Ind_t, 1088
unpack_loc_SetOperationMode_Ind_t, 1089

unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1163
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1165
wcdmaUARFCN, 1365
statusChange
UIMStatusChangeInfo, 1009
StopPDSTrackingSession
qaGobiApiPds.h, 1783
storageIndex
FMSImageIdElement, 247
ImageIdElement, 310
storageType
pack_sms_SaveSMS_t, 661
pack_sms_SLQSDeleteSMS_t, 663
pack_sms_SLQSGetSMS_t, 664
pack_sms_SLQSGetSMSList_t, 665
pack_sms_SLQSModifySMSStatus_t, 666
sms_maxStorageSizeReq, 891
smsMaxStorageSizeReq, 907
SMSMemoryInfo, 908
SMSMTMessage, 910
sMSMTMessage, 910
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1185
String
unpack_dms_GetDeviceHardwareRev_t, 1032
unpack_dms_GetDeviceMfr_t, 1032
unpack_dms_GetFSN_t, 1038
unpack_dms_UIMGetICCID_t, 1066
stringSize
unpack_dms_GetDeviceHardwareRev_t, 1032
unpack_dms_GetDeviceMfr_t, 1033
unpack_dms_UIMGetICCID_t, 1066
subAddr
calledPartySubAdd, 146
voice_calledPartySubAdd, 1278
subAddrLen
calledPartySubAdd, 146
voice_calledPartySubAdd, 1278
subAddrType
calledPartySubAdd, 146
voice_calledPartySubAdd, 1278
subnetMask
IPv4Addr, 328
unpack_qos_IPv4Addr_t, 1156
SubnetMaskV4
unpack_wds_SLQSGetRuntimeSettings_t, 1250
subsType
pack_voice_SLQSVoiceBindSubscription_t, 689
voiceBindSubscriptionInfo, 1304
SupUSBComps
unpack_dms_GetUSBComp_t, 1045
SuppOA
CUGInfo, 195
voice_CUGInfo, 1290
SuppPrefCUG
CUGInfo, 195
voice_CUGInfo, 1290
supportedMsgLen
SupportedMsgList, 921
SupportedMsgList, 921
supportedMsgLen, 921
supportedMsgs, 921
supportedMsgs
SupportedMsgList, 921
svInfoMask
loc_satelliteInfo, 371
satelliteInfo, 827
svListLen
loc_satelliteInfo, 371
satelliteInfo, 827
svStatus
loc_satelliteInfo, 371
satelliteInfo, 827
svUsedforFix
qaGobiApiCbk.h, 1569
svUsedforFix_s, 924
gnssSvUsedList, 924
gnssSvUsedList_len, 924
svc
pack_qmi_t, 658
SvcClass
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1281
voice_callFWInfo, 1282
SvcStatus
callFWExtInfo, 150
callFWInfo, 151
voice_callFWExtInfo, 1281
voice_callFWInfo, 1282
svcType
ccSUPSType, 161
SUPSInfo, 922
voice_ccSUPSType, 1284
sw1
cardResult, 155
uim_cardResult, 971
sw2
cardResult, 155
uim_cardResult, 971
swVerString
pack_dms_SLQSSwiSetHostDevInfo_t, 599
unpack_dms_SLQSSwiGetHostDevInfo_t, 1061
SwiAvmsEventInd
qaGobiApiCbk.h, 1569
SwiDataTypes.h, 2006
BOOL, 2007
BYTE, 2007
CHAR, 2007
FLOAT, 2007
INT32, 2007
INT8, 2007
LPCSTR, 2007
SHORT, 2007

SWI_API, 2007
ULONG, 2007
ULONGLONG, 2007
UNUSEDPARAM, 2007
USHORT, 2007
WORD, 2007
SwiLocGetAutoStart
qaGobiApiLoc.h, 1717
SwiLocGetAutoStartResp, 925
fix_rate, 927
fix_rate_reported, 927
fix_type, 927
fix_type_reported, 927
function, 927
function_reported, 927
max_dist, 927
max_dist_reported, 927
max_time, 927
max_time_reported, 927
SwiLocSetAutoStart
qaGobiApiLoc.h, 1717
SwiLocSetAutoStartReq, 927
fix_rate, 928
fix_type, 928
function, 928
max_dist, 928
max_time, 928
set_fix_rate, 928
set_fix_type, 929
set_function, 929
set_max_dist, 929
set_max_time, 929
swiModemStatusResp, 929
commonInfo, 929
pLTEInfo, 929
SwiOTAMsg
qaGobiApiCbk.h, 1569
SwiOTAMsg_s, 929
data, 930
data_len, 930
pLteNasRelInfo, 930
pTime, 930
type, 930
swiPDPRuntimeSettingsReq, 930
contextId, 931
contextType, 931
swiPDPRuntimeSettingsResp, 931
pAPNName, 932
pBearerId, 933
pContextId, 933
pIPv4Address, 933
pIPv4GWAddress, 933
pIPv6Address, 933
pIPv6GWAddress, 933
pPrDNSIPv4Address, 933
pPrDNSIPv6Address, 933
pPrPCSCFIPv4Address, 933
pPrPCSCFIPv6Address, 933
pSeDNSIPv4Address, 933
pSeDNSIPv6Address, 933
pSePCSCFIPv4Address, 933
pSePCSCFIPv6Address, 933
swiQosFilter, 933
index, 935
pEspSpi, 935
pIPv4DstAddr, 935
pIPv4SrcAddr, 935
pIPv6DstAddr, 935
pIPv6Label, 935
pIPv6SrcAddr, 935
pIPv6TrafCls, 935
pId, 935
pNxtHdrProto, 935
pPrecedence, 935
pTCPDstPort, 935
pTCPSrcPort, 935
pTos, 935
pTranDstPort, 935
pTranSrcPort, 935
pUDPDstPort, 935
pUDPSrcPort, 935
version, 935
swiQosFlow, 936
index, 938
p3GPP2Pri, 938
p3GPPImCn, 938
p3GPPResResidualBER, 938
p3GPPSigInd, 938
p3GPPTraHdlPri, 938
pDataRate, 938
pJitter, 938
pLatency, 938
pLteQci, 938
pMaxAllowedPktSz, 938
pMinPolicedPktSz, 939
pPktErrRate, 939
pProfileId3GPP2, 939
pTokenBucket, 939
pTrafficClass, 939
swiQosGranted, 939
pRxFlow, 939
pTxFlow, 939
swiQosIds, 939
pIds, 939
sz, 940
swiQosModifyReq, 940
id, 940
pRxFilter, 940
pRxFlow, 940
pTxFilter, 940
pTxFlow, 940
swiQosReq, 940
index, 941
pRxFilter, 941
pRxFlow, 941
pTxFilter, 941

pTxFlow, 941
swiRMTrasnferStaticsReq, 941
bResetStatistics, 942
ulMask, 942
swiloc.h, 2007
pack_swiloc_SwiLocGetAutoStart, 2008
pack_swiloc_SwiLocSetAutoStart, 2008
unpack_swiloc_SwiLocGetAutoStart, 2008
unpack_swiloc_SwiLocSetAutoStart, 2009
swioma.h, 2009
pack_swioma_SLQSOMADMAlertCallback, 2010
pack_swioma_SLQSOMADMCancelSession, 2011
pack_swioma_SLQSOMADMGetSessionInfo, 2011
pack_swioma_SLQSOMADMGetSettings, 2012
pack_swioma_SLQSOMADMSendSelection, 2012
pack_swioma_SLQSOMADMSetSettings, 2013
pack_swioma_SLQSOMADMStartSession, 2014
unpack_swioma_SLQSOMADMAlertCallback, 2014
unpack_swioma_SLQSOMADMAlertCallback_ind, 2015
unpack_swioma_SLQSOMADMCancelSession, 2015
unpack_swioma_SLQSOMADMGetSessionInfo, 2016
unpack_swioma_SLQSOMADMGetSettings, 2016
unpack_swioma_SLQSOMADMSendSelection, 2017
unpack_swioma_SLQSOMADMSetSettings, 2017
unpack_swioma_SLQSOMADMStartSession, 2017
switchOption
pack_voice_SLQSVoiceALSSetLineSwitching_t, 688
voiceALSSetLineSwitchInfo, 1303
sysInfoCDMA
CDMASysInfo, 174
nas_CDMASysInfo, 425
sysInfoCommon, 942
isSysForbidden, 944
isSysForbiddenValid, 944
roamStatus, 944
roamStatusValid, 944
srvCapability, 944
srvCapabilityValid, 944
srvDomain, 944
srvDomainValid, 944
sysInfoGSM
GSMSysInfo, 296
nas_GSMSysInfo, 446
sysInfoHDR
HDRSysInfo, 306
nas_HDRSysInfo, 451
sysInfoLTE
LTESysInfo, 408
nas_LTESysInfo, 470
sysInfoWCDMA
nas_WCDMASysInfo, 528
WCDMASysInfo, 1364
sysSelectPrefInfo
qaGobiApiNas.h, 1724
sysSelectPrefParams
qaGobiApiNas.h, 1729
system
loc_satelliteInfo, 371
loc_SV, 373
satelliteInfo, 827
SV, 923
systemDiscontinuities
pack_pds_PDSInjectTimeReference_t, 646
SystemID
qaQmiServingSystemParam, 769
unpack_nas_SLQSGetServingSystem_t, 1114
systemID
CDMASysInfo, 174
nas_CDMASysInfo, 425
systemMode
CommInfo, 185
nas_CommInfo, 428
systemTime
pack_pds_PDSInjectTimeReference_t, 646
sz
swiQosIds, 940
szAPN
_QmiSwiAvmsSetSettingsAPNInfo, 80
szAlertMsg
_QmiSwiAvmsEventReportConfigResp, 74
szDescription
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
szName
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
szPWD
_QmiSwiAvmsSetSettingsAPNInfo, 80
szUname
_QmiSwiAvmsSetSettingsAPNInfo, 80
szVersion
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73

t3396_apn
nasTimers, 570
unpack_nas_SLQSNasTimerCallback_ind_t, 1132
t3396_plmn_id
nasTimers, 570
unpack_nas_SLQSNasTimerCallback_ind_t, 1132
t3396_val
nasTimers, 570
unpack_nas_SLQSNasTimerCallback_ind_t, 1132
t_Sv, 945
entries, 945
len, 945
t_gpsTime, 944
gpsTimeOfWeekMs, 944
gpsWeek, 944

t_sensor, 944
aidingIndicatorMask, 944
usageMask, 944
TCPDstPort
unpack_qos_swiQosFilter_t, 1168
TCPSrcPort
unpack_qos_swiQosFilter_t, 1169
TDSCDMAECIOThresh, 945
TDSCDMAECIOThreshListLen
nas_TDSCDMAECIOThresh, 512
TDSCDMAECIOThresh, 945
TDSCDMARSCPThresh, 945
TDSCDMARSCPThreshListLen
nas_TDSCDMARSCPThresh, 512
TDSCDMARSCPThresh, 946
TDSCDMARSSIThresh, 946
TDSCDMARSSIThreshListLen
nas_TDSCDMARSSIThresh, 513
TDSCDMARSSIThresh, 946
TDSCDMASINRCONFThresh, 948
TDSCDMASINRThresh, 948
TDSCDMASINRThreshListLen
nas_TDSCDMASINRThresh, 514
TDSCDMASINRThresh, 949
TDSCDMASigInfoExt, 947
ecio, 948
rscp, 948
rssi, 948
sinr, 948
tFNASwiLTECphyCaInfo
qaGobiApiCbk.h, 1570
tFNASwiOTAMsg
qaGobiApiCbk.h, 1571
tFNActivationStatus
qaGobiApiCbk.h, 1570
tFNAllCallStatus
qaGobiApiCbk.h, 1570
tFNAsyncRawSend
qaGobiApiCbk.h, 1571
tFNBandPreference
qaGobiApiCbk.h, 1571
tFNBestAvailPos
qaGobiApiCbk.h, 1572
tFNCATEvent
qaGobiApiCbk.h, 1573
tFNCbkUimSlotStatusChangeInd
qaGobiApiCbk.h, 1573
tFNDHCPv4ClientLeaseStatus
qaGobiApiCbk.h, 1575
tFNDTMFEvent
qaGobiApiCbk.h, 1575
tFNDUNCallInfo
qaGobiApiCbk.h, 1575
tFNDataCapabilities
qaGobiApiCbk.h, 1573
tFNDataSysStatus
qaGobiApiCbk.h, 1574
tFNDelAssistData
qaGobiApiCbk.h, 1574
tFNDeviceStateChange
qaGobiApiCbk.h, 1574
tFNEventNmea
qaGobiApiCbk.h, 1575
tFNEventPosition
qaGobiApiCbk.h, 1575
tFNFwDldCompletion
qaGobiApiCbk.h, 1576
tFNGetServer
qaGobiApiCbk.h, 1576
tFNGnssSvInfo
qaGobiApiCbk.h, 1576
tFNHDRPersonaity
qaGobiApiCbk.h, 1576
tFNImsRegMgrConfig
qaGobiApiCbk.h, 1577
tFNImsSIPConfig
qaGobiApiCbk.h, 1578
tFNImsSMSConfig
qaGobiApiCbk.h, 1578
tFNImsUserConfig
qaGobiApiCbk.h, 1578
tFNImsVoIPConfig
qaGobiApiCbk.h, 1578
tFNImsaPdpStatus
qaGobiApiCbk.h, 1577
tFNImsaRatStatus
qaGobiApiCbk.h, 1577
tFNImsaRegStatus
qaGobiApiCbk.h, 1577
tFNImsaSvcStatus
qaGobiApiCbk.h, 1577
tFNInfoRec
qaGobiApiCbk.h, 1578
tFNInjectPosition
qaGobiApiCbk.h, 1579
tFNInjectSensorData
qaGobiApiCbk.h, 1579
tFNInjectTimeStatus
qaGobiApiCbk.h, 1579
tFNInjectUTCTime
qaGobiApiCbk.h, 1579
tFNLUReject
qaGobiApiCbk.h, 1580
tFNMemoryFull
qaGobiApiCbk.h, 1582
tFNMessageWaiting
qaGobiApiCbk.h, 1582
tFNMitiLvlRpt
qaGobiApiCbk.h, 1582
tFNMobileIPStatus
qaGobiApiCbk.h, 1582
tFNModemTempInfo
qaGobiApiCbk.h, 1582
tFNNasTimer
qaGobiApiCbk.h, 1583
tFNNet

qaGobiApiCbk.h, 1583
tFNNetworkReject
qaGobiApiCbk.h, 1583
tFNNetworkTime
qaGobiApiCbk.h, 1583
tFNNewGPS
qaGobiApiCbk.h, 1584
tFNNewNMEA
qaGobiApiCbk.h, 1584
tFNNewRMTransferStatistics
qaGobiApiCbk.h, 1584
tFNNewSMS
qaGobiApiCbk.h, 1585
tFNOMADMState
qaGobiApiCbk.h, 1585
tFNOTASPStatus
qaGobiApiCbk.h, 1587
tFNOpMode
qaGobiApiCbk.h, 1586
tFNPDSState
qaGobiApiCbk.h, 1587
tFNPacketSrvState
qaGobiApiCbk.h, 1587
tFNPower
qaGobiApiCbk.h, 1588
tFNPrivacyChange
qaGobiApiCbk.h, 1588
tFNQosNWStatus
qaGobiApiCbk.h, 1588
tFNQosPriEvent
qaGobiApiCbk.h, 1589
tFNQosStatus
qaGobiApiCbk.h, 1589
tFNRFInfo
qaGobiApiCbk.h, 1590
tFNRankIndicator
qaGobiApiCbk.h, 1590
tFNResetInfo
qaGobiApiCbk.h, 1590
tFNRoamingIndicator
qaGobiApiCbk.h, 1591
tFNSDKTerminated
qaGobiApiCbk.h, 1591
tFNSLQSOMADMAlert
qaGobiApiCbk.h, 1593
tFNSLQSQOSEvent
qaGobiApiCbk.h, 1593
tFNSLQSSessionState
qaGobiApiCbk.h, 1593
tFNSLQSSignalStrengths
qaGobiApiCbk.h, 1594
tFNSLQSWDSEvent
qaGobiApiCbk.h, 1594
tFNSMSEvents
qaGobiApiCbk.h, 1594
tFNSUPSInfo
qaGobiApiCbk.h, 1594
tFNSUPSNotification
qaGobiApiCbk.h, 1594
tFNSensorStreaming
qaGobiApiCbk.h, 1591
tFNServingSystem
qaGobiApiCbk.h, 1591
tFNSetCradleMount
qaGobiApiCbk.h, 1592
tFNSetEngineState
qaGobiApiCbk.h, 1592
tFNSetEventTimeSync
qaGobiApiCbk.h, 1592
tFNSetExtPowerConfig
qaGobiApiCbk.h, 1592
tFNSigInfo
qaGobiApiCbk.h, 1592
tFNSignalStrength
qaGobiApiCbk.h, 1593
tFNSwiAvmEventReport
qaGobiApiCbk.h, 1595
tFNSysInfo
qaGobiApiCbk.h, 1595
tFNSysSelectionPref
qaGobiApiCbk.h, 1595
tFNUIMRefresh
qaGobiApiCbk.h, 1596
tFNUIMStatusChangeInfo
qaGobiApiCbk.h, 1596
tFNUSSDNoWaitIndication
qaGobiApiCbk.h, 1596
tFNUSSDNotification
qaGobiApiCbk.h, 1596
tFNUSSDRelease
qaGobiApiCbk.h, 1597
tFNtransLayerInfo
qaGobiApiCbk.h, 1595
tFNtransNWRegInfo
qaGobiApiCbk.h, 1595
TFTIDParams, 951
destPortRangeEnd, 952
destPortRangeStart, 952
eValid, 952
filterId, 952
flowLabel, 953
IPSECSPI, 953
ipVersion, 953
nextHeader, 953
pSourceIP, 953
sourceIPMask, 953
srcPortRangeEnd, 953
srcPortRangeStart, 953
tosMask, 953
THIRD_INSTANCE
qaGobiApiCbk.h, 1559
TIME_DATE_BUF
qaGobiApiSms.h, 1795
TIME_STAMP_BUF
qaGobiApiSms.h, 1795
TPCause

SMSAsyncRawSend_s, 901
TRMessageTlv
unpack_sms_SetNewSMSCallback_ind_t, 1180
TX_PWR
nas_NetworkStat1x, 476
NetworkStat1x, 575
TXAGCList, 962
pTXAIG, 963
pTXComprSlope, 963
pTXComprThres, 963
pTXExpSlope, 963
pTXExpThres, 963
pTXStaticGain, 963
TXChan
LTEInfo, 393
nas_LTEInfo, 457
tXDroppedCount
unpack_wds_GetPacketStatus_t, 1239
TXOKBytesCount
DUNCallInfoInd, 233
tXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 1239
tXOkBytesCount
unpack_wds_GetPacketStatus_t, 1239
TXPCMIIRFltr, 964
pFlag, 965
pStage0Val, 965
pStage1Val, 965
pStage2Val, 965
pStage3Val, 965
pStage4Val, 965
pStageCnt, 965
tXPacketErrors
unpack_wds_GetPacketStatus_t, 1239
tXPacketOverflows
unpack_wds_GetPacketStatus_t, 1239
tXPacketSuccesses
unpack_wds_GetPacketStatus_t, 1239
Tables, 61
tac
LTEInfoIntrafreq, 395
LTESysInfo, 408
nas_LTEInfoIntrafreq, 460
nas_LTESysInfo, 470
tacValid
LTESysInfo, 408
nas_LTESysInfo, 470
TdsBandCapability
unpack_dms_SLQSGetBandCapability_t, 1056
tdscdmaSigInfoExt, 946
ecio, 947
rscp, 947
rssi, 947
sinr, 947
tech
NWProfile, 580
techName
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 1252
techType
DataBearerTech, 211
Technology
DeviceConfigDetail, 222
fwinfo_s, 250
nas_DeviceConfigDetail, 435
unpack_wds_SLQSGetRuntimeSettings_t, 1250
TechnologyPref
pack_nas_SetNetworkPreference_t, 624
tempData_t, 949
temperature, 950
temperatureDataLen, 950
timeOfFirstSample, 950
timeOffset, 950
timeSource, 950
temperature
CommInfo, 185
nas_CommInfo, 428
tempData_t, 950
tempratureData, 951
temperatureDataLen
tempData_t, 950
tempratureData, 951
tempratureData, 950
temperature, 951
temperatureDataLen, 951
timeOfFirstSample, 951
timeOffset, 951
timeSource, 951
textMsgLength
cdmaMsgEncodingParams, 169
Thermal Mitigation Device(TMD), 58
threshGsmHigh
lteGsmCellInfo, 391
nas_lteGsmCellInfo, 455
threshGsmLow
lteGsmCellInfo, 391
nas_lteGsmCellInfo, 455
threshServingLow
LTEInfoIntrafreq, 395
nas_LTEInfoIntrafreq, 460
threshXHigh
infoInterFreq, 327
nas_infoInterFreq, 453
threshXLow
infoInterFreq, 327
nas_infoInterFreq, 453
threshXhigh
lteWcdmaCellInfo, 409
nas_lteWcdmaCellInfo, 471
threshXlow
lteWcdmaCellInfo, 409
nas_lteWcdmaCellInfo, 471
thresholds
SignalStrengthDataType, 866
thresholdsSize

SignalStrengthDataType, 866
Time
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
wcdmaLongMsgDecodingParams, 1358
wcdmaMsgDecodingParams, 1359
time
NASTimeInfoTlv, 569
Time_uncert_ms
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
timeBase
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 652
timeInfo, 953
day, 954
dayLtSavingAdj, 954
dayOfWeek, 954
hour, 954
minute, 954
month, 954
radioInterface, 955
second, 955
timeZone, 955
TlvPresent, 955
year, 955
TimeLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
timeMsec
pack_loc_SLQSLOCInjectUTCTime_t, 618
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 652
timeOfFirstSample
sensorData, 830
sensorData_t, 831
tempData_t, 950
tempratureData, 951
timeOffset
sensorData, 830
sensorData_t, 831
tempData_t, 950
tempratureData, 951
timeSource
tempData_t, 950
tempratureData, 951
TimeStmp_gps_week
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
TimeStmp_tow_ms
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
timeSyncRefCounter
QmiCbkLocEventTimeSyncInd, 777
timeTlv
NASQmiCbkNasSwiOTAMessageInd, 559
timeUncMsec
pack_loc_SLQSLOCInjectUTCTime_t, 618
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 652
timeZone
nas_timeInfo, 516
timeInfo, 955
timeout
pack_pds_SetPDSDefaults_t, 649
pack_qmi_t, 658
Timers
_QmiSwiAvmsSetSettingsConnectionRetryTimers, 80
_SLQSAVMSSettingsConnectionRetryTimers, 86
timestamp
unpack_dms_GetNetworkTime_t, 1040
timestampAge
pack_loc_SLQSLOCInjectPosition_t, 616
timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 616
timingAdvance
GERANInfo, 252
nas_GERANInfo, 440
TlvPresent
_QmiSwiAvmsEventReportBinaryUpdateSessionInfoResp, 73
_QmiSwiAvmsEventReportConfigResp, 74
_QmiSwiAvmsEventReportConnectionRequestResp, 75
_QmiSwiAvmsEventReportDataSessionStatusResp, 75
_QmiSwiAvmsEventReportHTTPStatusResp, 76
_QmiSwiAvmsEventReportNotificationResp, 77
_QmiSwiAvmsEventReportPackageIDResp, 78
_QmiSwiAvmsEventReportRegStatusResp, 78
_QmiSwiAvmsEventReportSessionTypeResp, 79
_QmiSwiAvmsEventReportWAMSParamChangeResp, 79
CatCommonEventTlv, 158
CsgId, 193
DataULongLongTlv, 217
DataULongTlv, 217
dms_ActivationStatusTlv, 225
dms_OperatingModeTlv, 225
eTWSPLMNInfoTlv, 239
IPv4Info, 329
IPv6Info, 332
loc_IPv4Info, 367
loc_IPv6Info, 367
loc_urlAddr, 374
messageModeTlv, 409
nas_PhyCaAggPcellInfo, 482
nas_PhyCaAggScellDlBw, 482
nas_PhyCaAggScellIndex, 483
nas_PhyCaAggScellIndType, 483
nas_PhyCaAggScellInfo, 486
nas_RejectReasonTlv, 494
nas_RFInfoTlv, 495
nas_SccRxInfo, 502
nas_SignalStrengthTlv, 505

nas_SLQSSignalStrengthsTlv, 508
nas_timeInfo, 516
NASBandPreferenceTlv, 529
NASEmergencyModeTlv, 531
NASGWAcqOrderPrefTlv, 539
NASLTEBandPreferenceTlv, 544
NASLteNasReleaseInfoTlv, 544
NASModePreferenceTlv, 545
NASNetSelPreferenceTlv, 545
NASOTAMessageTlv, 551
NASPhyCaAggPcellInfo, 552
NASPhyCaAggScellDlBw, 552
NASPhyCaAggScellIndex, 553
NASPhyCaAggScellIndType, 553
NASPhyCaAggScellInfo, 554
NASPRLPreferenceTlv, 558
NASRoamPreferenceTlv, 560
NASServDomainPrefTlv, 561
NASTimeInfoTlv, 569
newMTMessageTlv, 577
PhyCaAggPcellInfo, 738
PhyCaAggScellDlBw, 738
PhyCaAggScellIndex, 739
PhyCaAggScellIndType, 740
PhyCaAggScellInfo, 742
PlmnID, 744
RoamingInfo, 815
SccRxInfo, 828
sessionInfoTlv, 838
sessionInfoTlvExt, 839
sMSCAddressTlv, 903
sMSEtwsMessageTlv, 904
sMSOnIMSTlv, 912
timeInfo, 955
transferRouteMessageTlv, 959
unpack_pds_SetPdsState_Ind_t, 1149
urlAddr, 1259
wds_DataULongLongTlv, 1366
wds_DataULongTlv, 1367
wds_DHCPLeaseOptTlv, 1367
wds_DHCPLeaseStateTlv, 1368
wds_DHCPProfileIdTlv, 1369
wds_IPv4AdTlv, 1374

Tlvresult
pack_dms_GetCustFeaturesV2_t, 591
pack_dms_SetCustFeaturesV2_t, 595
pack_dms_SetPower_t, 596
pack_dms_SetUSBComp_t, 597
pack_dms_UIMGetICCID_t, 601
pack_fms_GetImagesPreference_t, 605
pack_fms_GetStoredImages_t, 605
pack_fms_SetImagesPreference_t, 606
pack_loc_Delete_Assist_Data_t, 607
pack_loc_EventRegister_t, 609
pack_loc_SetExtPowerState_t, 609
pack_loc_SetOperationMode_t, 610
pack_loc_SLQSLOCGetBestAvailPos_t, 610
pack_loc_Start_t, 621
pack_loc_Stop_t, 621
pack_nas_SetNetworkPreference_t, 624
pack_uim_ChangePin_t, 675
pack_uim_ReadTransparent_t, 676
pack_uim_SetPinProtection_t, 677
pack_uim_UnblockPin_t, 684
pack_uim_VerifyPin_t, 686
unpack_dms_GetBandCapability_t, 1026
unpack_dms_GetCrashAction_t, 1026
unpack_dms_GetCustFeature_t, 1028
unpack_dms_GetCustFeaturesV2_t, 1029
unpack_dms_GetDeviceCap_t, 1030
unpack_dms_GetDeviceHardwareRev_t, 1032
unpack_dms_GetDeviceMfr_t, 1033
unpack_dms_GetDeviceSerialNumbers_t, 1034
unpack_dms_GetFirmwareInfo_t, 1035
unpack_dms_GetFirmwareRevision_t, 1036
unpack_dms_GetFirmwareRevisions_t, 1037
unpack_dms_GetFSN_t, 1038
unpack_dms_GetIMSI_t, 1039
unpack_dms_GetManufacturer_t, 1039
unpack_dms_GetModelID_t, 1040
unpack_dms_GetNetworkTime_t, 1040
unpack_dms_GetOfflineReason_t, 1041
unpack_dms_GetPower_t, 1042
unpack_dms_GetPRLVersion_t, 1043
unpack_dms_GetUSBComp_t, 1045
unpack_dms_GetVoiceNumber_t, 1046
unpack_dms_ResetToFactoryDefaults_t, 1047
unpack_dms_SetActivationStatusCallback_t, 1047
unpack_dms_SetCustFeature_t, 1048
unpack_dms_SetCustFeaturesV2_t, 1048
unpack_dms_SetEventReport_ind_t, 1049
unpack_dms_SetEventReport_t, 1049
unpack_dms_SetFirmwarePreference_t, 1049
unpack_dms_SetPower_t, 1050
unpack_dms_SetUSBComp_t, 1050
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1051
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1052
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1053
unpack_dms_SLQSGetERIFile_t, 1056
unpack_dms_SLQSSwiClearDyingGaspStatistics-_t, 1057
unpack_dms_SLQSSwiGetCrashInfo_t, 1057
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1058
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1058
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1061
unpack_dms_SLQSSwiGetHostDevInfo_t, 1061
unpack_dms_SLQSSwiGetOSInfo_t, 1062
unpack_dms_SLQSSwiGetSerialNoExt_t, 1063
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1063
unpack_dms_SLQSSwiSetHostDevInfo_t, 1063
unpack_dms_SLQSSwiSetOSInfo_t, 1064
unpack_dms_SLQSUIMGetState_t, 1064

unpack_dms_UIMGetControlKeyStatus_t, 1065
unpack_dms_UIMGetICCID_t, 1066
unpack_dms_UIMGetPINStatus_t, 1067
unpack_dms_UIMSetControlKeyProtection_t, 1068
unpack_dms_UIMSetPINProtection_t, 1068
unpack_dms_UIMUnblockControlKey_t, 1069
unpack_fms_GetImagesPreference_t, 1070
unpack_fms_GetStoredImages_t, 1070
unpack_fms_SetImagesPreference_t, 1071
unpack_loc_BestAvailPos_Ind_t, 1076
unpack_loc_Delete_Assist_Data_t, 1077
unpack_loc_DeleteAssistData_Ind_t, 1078
unpack_loc_EngineState_Ind_t, 1078
unpack_loc_EventNMEA_Ind_t, 1079
unpack_loc_EventRegister_t, 1079
unpack_loc_GetServer_Ind_t, 1080
unpack_loc_GnssSvInfo_Ind_t, 1081
unpack_loc_PositionRpt_Ind_t, 1087
unpack_loc_SetExtPowerConfig_Ind_t, 1088
unpack_loc_SetExtPowerState_t, 1089
unpack_loc_SetOperationMode_Ind_t, 1089
unpack_loc_SetOperationMode_t, 1090
unpack_loc_SLQSLOCGetBestAvailPos_t, 1090
unpack_loc_Start_t, 1090
unpack_loc_Stop_t, 1091
unpack_nas_GetNetworkPreference_t, 1097
unpack_nas_SetNetworkPreference_t, 1105
unpack_nas_SetServingSystemCallback_ind_t, 1106
unpack_nas_SlqsGetLTECphyCAInfo_t, 1108
unpack_nas_SLQSNasSwiOTAMessageCallback-_ind_t, 1131
unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind_t, 1132
unpack_uim_ChangePin_t, 1193
unpack_uim_GetCardStatus_t, 1194
unpack_uim_ReadTransparent_t, 1195
unpack_uim_SetPinProtection_t, 1195
unpack_uim_UnblockPin_t, 1201
unpack_uim_VerifyPin_t, 1201
unpack_wds_SLQSCreateProfile_t, 1241
unpack_wds_SLQSGetProfileSettings_t, 1248
unpack_wds_SLQSSetIPFamilyPreference_t, 1251
TmdDeRegNotMitigationLvlReq, 955
mitigationDevID, 955
mitigationDevIDLen, 955
TmdGetMitigationDevListResp, 955
pMitigationDevList, 956
pMitigationDevListLen, 956
TmdGetMitigationLvlReq, 956
mitigationDevID, 956
mitigationDevIDLen, 956
TmdGetMitigationLvlResp, 956
pCurrentmitigationLvl, 957
pReqMitigationLvl, 957
TmdMitigationLvlIndReq, 957
mitigationDevID, 957
mitigationDevIDLen, 957
TmdRegNotMitigationLvlReq, 957
mitigationDevID, 958
mitigationDevIDLen, 958
toServiceId
BroadcastConfig, 142
sms_BroadcastConfig, 887
toggleMode
lineCtrlInfo, 361
TokenBucket
unpack_qos_swiQosFlow_t, 1173
tokenBucket, 958
bucketSz, 958
peakRate, 958
tokenRate, 958
tokenRate
tokenBucket, 958
unpack_qos_tokenBucket_t, 1173
Tos, 958
mask, 959
val, 959
tosMask
LibPackTFTIDParams, 358
TFTIDParams, 953
total_rx_bytes
sQosStat, 917
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
total_rx_pkt
sQosStat, 917
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
total_tx_bytes
sQosStat, 917
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
total_tx_bytes_drp
sQosStat, 917
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
total_tx_pkt
sQosStat, 917
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
total_tx_pkt_drp
sQosStat, 917
unpack_qos_SLQSQosSwiReadDataStats_t, 1162
TrStatInd, 961
statsMask, 961
statsPeriod, 961
TrackAreaCode
unpack_nas_SLQSGetServingSystem_t, 1114
trackAreaCode
qaQmiServingSystemParam, 769
TrackingStatus
unpack_pds_SetPdsState_Ind_t, 1149
TrafficClass
unpack_qos_swiQosFlow_t, 1173
trafficClass
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380

trafficPriority
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
TranDstPort
unpack_qos_swiQosFilter_t, 1169
TranSrcPort
unpack_qos_swiQosFilter_t, 1169
TransCap
_transLayerinfo, 111
sms_transLayerInfo, 899
transLayerInfo
qaGobiApiSms.h, 1798
transLayerNotification
qaGobiApiCbk.h, 1597
transNWRegInfoNotification
qaGobiApiCbk.h, 1597
TransType
_transLayerinfo, 111
sms_transLayerInfo, 900
transactionID
SMSTransferRouteMTMessage, 914
sMSTransferRouteMTMessage, 915
transferDelay
LibPackUMTSQoS, 360
UMTSMinQoS, 1018
UMTSQoS, 1020
wds_UMTSMinQoS, 1380
TransferRouteMTMessageInfo
transferRouteMessageTlv, 959
transferRouteMessageTlv, 959
TlvPresent, 959
TransferRouteMTMessageInfo, 959
TransferStatInd, 960
StatsMask, 960
StatsPeriod, 960
transferStatInd, 960
StatsMask, 960
StatsPeriod, 960
transferStats
pack_wds_SLQSSetWdsEventCallback_t, 723
TransferStatsDataType, 961
interval, 961
trueIMSI, 961
imsiT1112, 962
imsiTS1, 962
imsiTS2, 962
imsiTaddrNum, 962
mccT, 962
trueSrvStatus
GSMSrvStatusInfo, 293
nas_GSMSrvStatusInfo, 443
tx_bytes
NetStats, 572
sQosFlowStat, 916
unpack_QosFlowStat_t, 1175
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
tx_bytes_drp
sQosFlowStat, 916
unpack_QosFlowStat_t, 1175
tx_errors
NetStats, 572
tx_overflows
NetStats, 572
tx_packets
NetStats, 572
tx_pkt
sQosFlowStat, 916
unpack_QosFlowStat_t, 1175
tx_pkt_drp
sQosFlowStat, 916
unpack_QosFlowStat_t, 1175
tx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
TxDropConutTlv
QmiCbkWdsStatisticsIndState, 791
unpack_RMTransferStatistics_ind_t, 1176
txInfo, 963
isInTraffic, 964
txPower, 964
txOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 1248
TxOkByteCountTlv
QmiCbkWdsStatisticsIndState, 791
unpack_RMTransferStatistics_ind_t, 1176
TxOkConutTlv
QmiCbkWdsStatisticsIndState, 791
unpack_RMTransferStatistics_ind_t, 1176
txPower
nas_txInfo, 517
txInfo, 964
TxQFilter
unpack_qos_QosFlowInfo_t, 1159
TxQFlowGranted
unpack_qos_QosFlowInfo_t, 1159
type
_getResetInfoNotification, 66
dmsSwiGetResetInfo, 230
pack_wds_GetDefaultProfileNum_t, 705
pack_wds_SetDefaultProfileNum_t, 711
SwiOTAMsg_s, 930
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1051
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1052
unpack_qmi_t, 1155
typeSize
unpack_sms_GetSMSCAddress_t, 1178

u16PRLVersion
unpack_dms_GetPRLVersion_t, 1043
u8PRLPreference
unpack_dms_GetPRLVersion_t, 1043
UATISIZE

qaGobiApiNas.h, 1723
UDPDstPort
unpack_qos_swiQosFilter_t, 1169
UDPSrcPort
unpack_qos_swiQosFilter_t, 1169
UIMAuthenticateReq, 990
authData, 991
pIndicationToken, 991
sessionInfo, 991
UIMAuthenticateResp, 991
pAuthenticateResult, 992
pCardResult, 992
pIndicationToken, 992
UIMChangePIN
qaGobiApiDms.h, 1674
UIMChangePinReq, 992
changePIN, 992
pIndicationToken, 992
pKeyReferenceID, 992
sessionInfo, 992
UIMDepersonalizationReq, 993
depersonilisationInfo, 993
UIMDepersonalizationResp, 993
pRemainingRetries, 993
UIMEventRegisterReqResp, 993
eventMask, 994
UIMGetCardStatusResp, 994
pCardStatus, 994
pHotSwapStatus, 994
UIMGetConfigurationReq, 994
pConfigurationMask, 995
UIMGetConfigurationResp, 995
pAutoSelection, 996
pHaltSubscription, 996
pPersonalizationStatus, 996
UIMGetControlKeyStatus
qaGobiApiDms.h, 1675
UIMGetFileAttributesReq, 996
fileIndex, 996
pIndicationToken, 996
sessionInfo, 996
UIMGetFileAttributesResp, 996
pCardResult, 997
pFileAttributes, 997
pIndicationToken, 997
UIMGetICCID
qaGobiApiDms.h, 1676
UIMGetPINStatus
qaGobiApiDms.h, 1677
UIMGetSlotsStatusResp, 997
pNumberOfPhySlot, 997
pUimSlotsStatus, 997
UIMPinResp, 998
pEncryptedPIN1, 998
pIndicationToken, 998
pRemainingRetries, 998
UIMPowerDownReq, 998
slot, 999
UIMPowerUpReq, 999
pIgnoreHotSwapSwitch, 999
slot, 999
UIMReadTransparentReq, 999
fileIndex, 1000
pEncryptData, 1000
pIndicationToken, 1000
readTransparent, 1000
sessionInfo, 1000
UIMReadTransparentResp, 1000
pCardResult, 1001
pEncryptedData, 1001
pIndicationToken, 1001
pReadResult, 1001
UIMRefreshCompleteReq, 1001
refreshComplete, 1002
sessionInfo, 1002
UIMRefreshEvent, 1002
aid, 1003
aidLength, 1003
arrfileInfo, 1003
mode, 1003
numOfFiles, 1003
sessionType, 1003
stage, 1003
UIMRefreshGetLastEventReq, 1003
sessionInfo, 1004
UIMRefreshGetLastEventResp, 1004
pRefreshEvent, 1004
UIMRefreshOKReq, 1004
OKtoRefresh, 1005
sessionInfo, 1005
UIMRefreshRegisterReq, 1005
regRefresh, 1005
sessionInfo, 1005
UIMSessionInformation, 1005
aid, 1006
aidLength, 1006
sessionType, 1006
UIMSetControlKeyProtection
qaGobiApiDms.h, 1678
UIMSetPINProtection
qaGobiApiDms.h, 1678
UIMSetPinProtectionReq, 1006
pIndicationToken, 1007
pKeyReferenceID, 1007
pinProtection, 1007
sessionInfo, 1007
UIMSlotStatus, 1007
bICCID, 1008
bICCIDLength, 1008
bLogicalSlot, 1008
uPhyCardStatus, 1008
uPhySlotStatus, 1008
UIMSlotStatusChangeInfo, 1009
bNumberOfPhySlots, 1009
slotsstatusChange, 1009
UIMSlotsStatus, 1007

uimSlotStatus, 1007
UIMStatusChangeInfo, 1009
statusChange, 1009
UIMSwitchSlotReq, 1009
bLogicalSlot, 1010
ulPhysicalSlot, 1010
UIMUnblockControlKey
qaGobiApiDms.h, 1679
UIMUnblockPIN
qaGobiApiDms.h, 1680
UIMUnblockPinReq, 1010
pIndicationToken, 1011
pKeyReferenceID, 1011
sessionInfo, 1011
unblockPIN, 1011
UIMVerifyPIN
qaGobiApiDms.h, 1681
UIMVerifyPinReq, 1011
pEncryptedPIN1, 1012
pIndicationToken, 1012
pKeyReferenceID, 1012
sessionInfo, 1012
verifyPIN, 1012
ULONG
SwiDataTypes.h, 2007
ULONGLONG
SwiDataTypes.h, 2007
UMTSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 1250
UMTSInfo, 1012
cellID, 1013
ecio, 1013
geranInst, 1013
GeranInstInfo, 1013
lac, 1013
plmn, 1013
psc, 1014
rscp, 1014
UMTSInstInfo, 1014
uarfcn, 1014
umtsInst, 1014
UMTSInstInfo
nas_UMTSInfo, 519
UMTSInfo, 1014
UMTSLTENbrCell
nas_WCDMAInfoLTENeighborCell, 524
WCDMAInfoLTENeighborCell, 1356
UMTSMinQoS, 1015
deliveryErrSDU, 1017
grntDownlinkBitrate, 1017
grntUplinkBitrate, 1018
maxDownlinkBitrate, 1018
maxSDUSize, 1018
maxUplinkBitrate, 1018
qosDeliveryOrder, 1018
resBerRatio, 1018
sduErrorRatio, 1018
trafficClass, 1018
trafficPriority, 1018
transferDelay, 1018
UMTSQoS, 1018
deliveryErrSDU, 1020
grntDownlinkBitrate, 1020
grntUplinkBitrate, 1020
maxDownlinkBitrate, 1020
maxSDUSize, 1020
maxUplinkBitrate, 1020
qosDeliveryOrder, 1020
resBerRatio, 1020
sduErrorRatio, 1020
trafficClass, 1020
trafficPriority, 1020
transferDelay, 1020
UMTSReqQoS
LibPackUMTSReqQoSSigInd, 361
UMTSReqQoSSigInd, 1021
UMTSReqQoSSigInd, 1020
SigInd, 1021
UMTSReqQoS, 1021
UMTSinstInfo, 1014
umtsEcio, 1014
umtsPsc, 1014
umtsRscp, 1014
umtsUarfcn, 1014
UNIQUE_ID_LEN
dms.h, 1411
qaGobiApiFms.h, 1686
UNUSEDPARAM
common.h, 1401
SwiDataTypes.h, 2007
uPhyCardStatus
slot_t, 867
UIMSlotStatus, 1008
uPhySlotStatus
slot_t, 867
UIMSlotStatus, 1008
uResult
sGetDeviceSeriesResult, 863
USBComp
pack_dms_SetUSBComp_t, 597
unpack_dms_GetUSBComp_t, 1045
USBCompConfig, 1259
pUSBComp, 1260
USBCompParams, 1260
pNumSupUSBComps, 1262
pSupUSBComps, 1262
pUSBComp, 1262
USHORT
SwiDataTypes.h, 2007
USSD_DCS_8BIT
qaGobiApiCbk.h, 1559
USSD_DCS_ASCII
qaGobiApiCbk.h, 1559
USSD_DCS_UCS2
qaGobiApiCbk.h, 1559
USSDNoWaitIndicationInfo, 1262

pAlphaIdentifier, 1263
pError, 1263
pFailureCause, 1263
pUSSDData, 1263
USSDRespFNetwork, 1263
pRespData, 1263
pTypeCode, 1263
USSInfo, 1263
ussDCS, 1264
ussData, 1264
ussLen, 1264
USSInformation
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 698
voiceOrigUSSDNoWaitInfo, 1336
USSResp, 1264
pAlphaIDInfo, 1264
pCCSuppsType, 1264
pCallId, 1264
pCcResultType, 1264
pUSSDInfo, 1265
pfailureCause, 1264
UUSData
UUSInfo, 1266
voice_UUSInfo, 1303
UUSDatalen
UUSInfo, 1266
voice_UUSInfo, 1303
UUSDcs
UUSInfo, 1266
voice_UUSInfo, 1303
UUSInfo, 1265
UUSData, 1266
UUSDatalen, 1266
UUSDcs, 1266
UUSType, 1266
UUSType
UUSInfo, 1266
voice_UUSInfo, 1303
uarfcn
lteWcdmaCellInfo, 409
nas_lteWcdmaCellInfo, 471
nas_UMTSInfo, 519
nas_wcdmaUARFCN, 528
UMTSInfo, 1014
wcdmaUARFCN, 1365
ueInIdle
LTEInfoInterfreq, 393
LTEInfoIntrafreq, 395
LTEInfoNeighboringGSM, 396
LTEInfoNeighboringWCDMA, 397
nas_LTEInfoInterfreq, 458
nas_LTEInfoIntrafreq, 460
nas_LTEInfoNeighboringGSM, 461
nas_LTEInfoNeighboringWCDMA, 462
uim.h, 2018
MAX_ICCID_LENGTH, 2021
MAX_NO_OF_SLOTS, 2021
MAX_SLOTS_STATUS, 2021
pack_uim_ChangePin, 2021
pack_uim_GetCardStatus, 2022
pack_uim_ReadTransparent, 2022
pack_uim_SLQSUIMAuthenticate, 2023
pack_uim_SLQSUIMDepersonalization, 2023
pack_uim_SLQSUIMEventRegister, 2023
pack_uim_SLQSUIMGetConfiguration, 2024
pack_uim_SLQSUIMGetFileAttributes, 2024
pack_uim_SLQSUIMGetSlotsStatus, 2025
pack_uim_SLQSUIMPowerDown, 2025
pack_uim_SLQSUIMPowerUp, 2025
pack_uim_SLQSUIMRefreshComplete, 2026
pack_uim_SLQSUIMRefreshGetLastEvent, 2026
pack_uim_SLQSUIMRefreshOK, 2026
pack_uim_SLQSUIMRefreshRegister, 2027
pack_uim_SLQSUIMReset, 2027
pack_uim_SLQSUIMSwitchSlot, 2027
pack_uim_SetPinProtection, 2022
pack_uim_UnblockPin, 2028
pack_uim_VerifyPin, 2028
unpack_uim_ChangePin, 2029
unpack_uim_GetCardStatus, 2029
unpack_uim_ReadTransparent, 2029
unpack_uim_SLQSUIMAuthenticate, 2030
unpack_uim_SLQSUIMDepersonalization, 2031
unpack_uim_SLQSUIMEventRegister, 2031
unpack_uim_SLQSUIMGetConfiguration, 2031
unpack_uim_SLQSUIMGetFileAttributes, 2032
unpack_uim_SLQSUIMGetSlotsStatus, 2032
unpack_uim_SLQSUIMPowerDown, 2033
unpack_uim_SLQSUIMPowerUp, 2033
unpack_uim_SLQSUIMRefreshComplete, 2033
unpack_uim_SLQSUIMRefreshGetLastEvent, 2033
unpack_uim_SLQSUIMRefreshOK, 2034
unpack_uim_SLQSUIMRefreshRegister, 2034
unpack_uim_SLQSUIMReset, 2034
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind, 2035
unpack_uim_SLQSUIMSwitchSlot, 2035
unpack_uim_SetPinProtection, 2030
unpack_uim_SetUimSlotStatusChangeCallback_ind, 2030
unpack_uim_UnblockPin, 2036
unpack_uim_VerifyPin, 2036
uim_UIMSessionInformation, 988
aid, 989
aidLength, 989
sessionType, 989
uim_appStatus, 965
aidLength, 968
aidVal, 968
appState, 968
appType, 968
persoFeature, 968
persoRetries, 968
persoState, 968
persoUnblockRetries, 968

pin1Retries, 968
pin1State, 968
pin2Retries, 968
pin2State, 968
puk1Retries, 968
puk2Retries, 968
univPin, 968
uim_authenticateResult, 968
content, 969
contentLen, 969
uim_authenticationData, 969
context, 970
data, 970
dataLen, 970
uim_cardResult, 970
sw1, 971
sw2, 971
uim_cardStatus, 971
index1xPri, 972
index1xSec, 972
indexGwPri, 972
indexGwSec, 972
numSlot, 972
SlotInfo, 972
uim_changeUIMPIN, 972
oldPINLen, 973
oldPINVal, 973
pinID, 973
pinLen, 973
pinVal, 973
uim_depersonalizationInformation, 973
ckLen, 974
ckVal, 974
feature, 974
operation, 974
uim_encryptedPIN1, 974
pin1Len, 975
pin1Val, 975
uim_fileAttributes, 975
fileID, 978
fileSize, 978
fileType, 978
rawLen, 978
rawValue, 978
recordCount, 978
recordSize, 978
secActivate, 978
secActivateMask, 978
secDeactivate, 978
secDeactivateMask, 978
secIncrease, 978
secIncreaseMask, 978
secRead, 978
secReadMask, 978
secWrite, 978
secWriteMask, 978
uim_fileInfo, 978
fileID, 979
path, 979
pathLen, 979
uim_hotSwapStatus, 979
hotSwap, 979
hotSwapLength, 979
uim_personalizationStatus, 979
feature, 980
numFeatures, 980
unblockLeft, 980
verifyLeft, 980
uim_readResult, 980
content, 981
contentLen, 981
uim_readTransparentInfo, 981
length, 981
offset, 981
uim_refreshevent, 981
aid, 983
aidLength, 983
arrfileInfo, 983
mode, 983
numOfFiles, 983
sessionType, 983
stage, 983
uim_registerRefresh, 983
arrfileInfo, 984
numFiles, 984
registerFlag, 984
voteForInit, 984
uim_remainingRetries, 984
unblockLeft, 984
verifyLeft, 984
uim_sessionInformation, 985
aid, 985
aidLength, 985
sessionType, 985
uim_setPINProtection, 985
pinID, 986
pinLength, 986
pinOperation, 986
pinValue, 986
uim_slotInfo, 986
AppStatus, 988
cardState, 988
errorState, 988
numApp, 988
upinRetries, 988
upinState, 988
upukRetries, 988
uim_unblockUIMPIN, 989
newPINLen, 989
newPINVal, 989
pinID, 989
pukLen, 989
pukVal, 990
uim_verifyUIMPIN, 990
pinID, 990
pinLen, 990

pinVal, 990
uimSlotStatus
slots_t, 870
UIMSlotsStatus, 1007
ulData
DataULongTlv, 217
wds_DataULongTlv, 1367
ulMask
rmTrasnferStaticsReq, 814
swiRMTrasnferStaticsReq, 942
ulMax
_QmiSwiAvmsSetSettingsPeriodInfo, 80
ulMin
_QmiSwiAvmsSetSettingsPeriodInfo, 80
ulPhysicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 683
UIMSwitchSlotReq, 1010
ulPkgDownloadComplete
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
ulPkgDownloadSize
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
ullData
DataULongLongTlv, 217
wds_DataULongLongTlv, 1366
umtsEcio
nas_UMTSinstInfo, 519
UMTSinstInfo, 1014
umtsInst
nas_UMTSInfo, 519
UMTSInfo, 1014
umtsLTENbrCell, 1014
cellIsTDD, 1015
earfcn, 1015
pci, 1015
rsrp, 1015
rsrq, 1015
srxlev, 1015
umtsLTENbrCellLen
nas_WCDMAInfoLTENeighborCell, 524
WCDMAInfoLTENeighborCell, 1356
umtsPsc
nas_UMTSinstInfo, 519
UMTSinstInfo, 1014
umtsRscp
nas_UMTSinstInfo, 520
UMTSinstInfo, 1014
umtsUarfcn
nas_UMTSinstInfo, 520
UMTSinstInfo, 1014
UnPackGetProfileSettingOut, 1258
curProfile, 1258
pExtErrCode, 1258
unblockLeft
personalizationStatus, 737
remainingRetries, 809
uim_personalizationStatus, 980
uim_remainingRetries, 984
unblockPIN
UIMUnblockPinReq, 1011
unblockRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1065
unpack_dms_UIMSetPINProtection_t, 1068
unpack_dms_UIMUnblockControlKey_t, 1069
unblockUIMPIN, 1021
newPINLen, 1022
newPINVal, 1022
pinID, 1022
pukLen, 1022
pukVal, 1022
uniqueID
CurrImageInfo, 200
image_info_t, 308
univPin
appStats, 124
appStatus, 127
uim_appStatus, 968
UniversalTime, 1022
day, 1023
dayOfWeek, 1023
hour, 1023
minute, 1023
month, 1023
second, 1023
year, 1023
universalTime
nasNetworkTime, 549
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 1128
unpack_QosFlowStat_t, 1174
bearerId, 1175
tx_bytes, 1175
tx_bytes_drp, 1175
tx_pkt, 1175
tx_pkt_drp, 1175
unpack_RMTransferStatistics_ind_t, 1175
RxDropConutTlv, 1176
RxOkByteCountTlv, 1176
RxOkConutTlv, 1176
TxDropConutTlv, 1176
TxOkByteCountTlv, 1176
TxOkConutTlv, 1176
unpack_dms_ActivateAutomatic
dms.h, 1434
unpack_dms_GetActivationState
dms.h, 1434
unpack_dms_GetActivationState_t, 1023
state, 1024
unpack_dms_GetBandCapability
dms.h, 1435
unpack_dms_GetBandCapability_t, 1024
BandCapability, 1025
Tlvresult, 1026
unpack_dms_GetCrashAction
dms.h, 1435

unpack_dms_GetCrashAction_t, 1026
DevCrashState, 1026
Tlvresult, 1026
unpack_dms_GetCustFeature
dms.h, 1435
unpack_dms_GetCustFeature_t, 1026
DHCPRelayEnabled, 1028
DisableIMSI, 1028
GPSLPM, 1028
GPSSel, 1028
GpsEnable, 1028
IPFamSupport, 1028
IsVoiceEnabled, 1028
RMAutoConnect, 1028
SMSSupport, 1028
Tlvresult, 1028
unpack_dms_GetCustFeaturesV2
dms.h, 1436
unpack_dms_GetCustFeaturesV2_t, 1028
GetCustomFeatureV2, 1029
Tlvresult, 1029
unpack_dms_GetDeviceCap
dms.h, 1436
unpack_dms_GetDeviceCap_t, 1029
DataServiceCapability, 1030
MaxRXChannelRate, 1030
MaxTXChannelRate, 1030
RadioIfaces, 1030
RadioIfacesSize, 1030
SimCapability, 1030
Tlvresult, 1030
unpack_dms_GetDeviceCapabilities
dms.h, 1436
unpack_dms_GetDeviceCapabilities_t, 1030
dataServiceCaCapability, 1031
maxRxChannelRate, 1031
maxTxChannelRate, 1031
RadioIfaces, 1031
radioIfacesSize, 1031
simCapability, 1031
unpack_dms_GetDeviceHardwareRev
dms.h, 1437
unpack_dms_GetDeviceHardwareRev_t, 1031
String, 1032
stringSize, 1032
Tlvresult, 1032
unpack_dms_GetDeviceMfr
dms.h, 1437
unpack_dms_GetDeviceMfr_t, 1032
String, 1032
stringSize, 1033
Tlvresult, 1033
unpack_dms_GetDeviceSerialNumbers
dms.h, 1437
unpack_dms_GetDeviceSerialNumbers_t, 1033
ESNString, 1034
esnSize, 1034
IMEIString, 1034
imeiSize, 1034
imeiSvnSize, 1034
ImeiSvnString, 1034
MEIDString, 1034
meidSize, 1034
Tlvresult, 1034
unpack_dms_GetFSN
dms.h, 1439
unpack_dms_GetFSN_t, 1037
String, 1038
Tlvresult, 1038
unpack_dms_GetFirmwareInfo
dms.h, 1438
unpack_dms_GetFirmwareInfo_t, 1034
appversion_str, 1035
bootversion_str, 1035
carrier_str, 1035
cur_carr_name, 1035
cur_carr_rev, 1035
modelid_str, 1035
packageid_str, 1035
priversion_str, 1035
sku_str, 1035
Tlvresult, 1035
unpack_dms_GetFirmwareRevision
dms.h, 1438
unpack_dms_GetFirmwareRevision_t, 1036
AMSSString, 1036
amssSize, 1036
PRIString, 1036
Tlvresult, 1036
unpack_dms_GetFirmwareRevisions
dms.h, 1438
unpack_dms_GetFirmwareRevisions_t, 1036
AMSSString, 1037
amssSize, 1037
bootSize, 1037
BootString, 1037
PRIString, 1037
priSize, 1037
Tlvresult, 1037
unpack_dms_GetHardwareRevision
dms.h, 1439
unpack_dms_GetHardwareRevision_t, 1038
hwVer, 1038
unpack_dms_GetIMSI
dms.h, 1439
unpack_dms_GetIMSI_t, 1038
imsi, 1039
Tlvresult, 1039
unpack_dms_GetManufacturer
dms.h, 1440
unpack_dms_GetManufacturer_t, 1039
manufacturer, 1039
Tlvresult, 1039
unpack_dms_GetModelID
dms.h, 1440
unpack_dms_GetModelID_t, 1039

modelid, 1040
Tlvresult, 1040
unpack_dms_GetNetworkTime
dms.h, 1440
unpack_dms_GetNetworkTime_t, 1040
source, 1040
timestamp, 1040
Tlvresult, 1040
unpack_dms_GetOfflineReason
dms.h, 1441
unpack_dms_GetOfflineReason_t, 1041
pReasonMask, 1041
pbPlatform, 1041
Tlvresult, 1041
unpack_dms_GetPRLVersion
dms.h, 1441
unpack_dms_GetPRLVersion_t, 1042
Tlvresult, 1043
u16PRLVersion, 1043
u8PRLPreference, 1043
unpack_dms_GetPower
dms.h, 1441
unpack_dms_GetPower_t, 1041
HardwareControlledMode, 1042
OfflineReason, 1042
OperationMode, 1042
Tlvresult, 1042
unpack_dms_GetSerialNumbers
dms.h, 1442
unpack_dms_GetSerialNumbers_t, 1043
esn, 1044
imei_no, 1044
imeisv_svn, 1044
meid, 1044
unpack_dms_GetUSBComp
dms.h, 1442
unpack_dms_GetUSBComp_t, 1044
NumSupUSBComps, 1045
SupUSBComps, 1045
Tlvresult, 1045
USBComp, 1045
unpack_dms_GetVoiceNumber
dms.h, 1442
unpack_dms_GetVoiceNumber_t, 1045
MIN, 1046
minSize, 1046
Tlvresult, 1046
VoiceNumber, 1046
voiceNumberSize, 1046
unpack_dms_ResetToFactoryDefaults
dms.h, 1443
unpack_dms_ResetToFactoryDefaults_t, 1046
Tlvresult, 1047
unpack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1446
unpack_dms_SLQSDmsSwiGetResetInfo_Ind
dms.h, 1446
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 1050
source, 1051
Tlvresult, 1051
type, 1051
unpack_dms_SLQSDmsSwiGetResetInfo_t, 1051
source, 1052
Tlvresult, 1052
type, 1052
unpack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1447
unpack_dms_SLQSDmsSwiIndicationRegister_t, 1052
Tlvresult, 1053
unpack_dms_SLQSGetBandCapability
dms.h, 1447
unpack_dms_SLQSGetBandCapability_t, 1053
bandCapability, 1056
LteBandCapability, 1056
TdsBandCapability, 1056
unpack_dms_SLQSGetERIFile
dms.h, 1448
unpack_dms_SLQSGetERIFile_t, 1056
eriFile, 1056
Tlvresult, 1056
unpack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1448
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 1056
Tlvresult, 1057
unpack_dms_SLQSSwiGetCrashInfo
dms.h, 1448
unpack_dms_SLQSSwiGetCrashInfo_t, 1057
crashInfoParam, 1057
Tlvresult, 1057
unpack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1449
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 1057
pGetDyingGaspCfg, 1058
Tlvresult, 1058
unpack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1449
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 1058
pGetDyingGaspStatistics, 1058
Tlvresult, 1058
unpack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1449
unpack_dms_SLQSSwiGetFirmwareCurr_t, 1058
carrier, 1059
fwvers, 1059
numEntries, 1059
pCurrImgInfo, 1059
pkgver, 1059
priver, 1059
unpack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1450
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 1059
imgType, 1060
logString, 1060
refData, 1060
refString, 1060

ResCode, 1061
TlvResult, 1061
unpack_dms_SLQSSwiGetHostDevInfo
dms.h, 1450
unpack_dms_SLQSSwiGetHostDevInfo_t, 1061
manString, 1061
modelString, 1061
plasmaIDString, 1061
swVerString, 1061
TlvResult, 1061
unpack_dms_SLQSSwiGetOSInfo
dms.h, 1450
unpack_dms_SLQSSwiGetOSInfo_t, 1062
nameString, 1062
TlvResult, 1062
versionString, 1062
unpack_dms_SLQSSwiGetSerialNoExt
dms.h, 1451
unpack_dms_SLQSSwiGetSerialNoExt_t, 1062
meidString, 1063
TlvResult, 1063
unpack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1451
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 1063
TlvResult, 1063
unpack_dms_SLQSSwiSetHostDevInfo
dms.h, 1451
unpack_dms_SLQSSwiSetHostDevInfo_t, 1063
TlvResult, 1063
unpack_dms_SLQSSwiSetOSInfo
dms.h, 1452
unpack_dms_SLQSSwiSetOSInfo_t, 1063
TlvResult, 1064
unpack_dms_SLQSUIMGetState
dms.h, 1452
unpack_dms_SLQSUIMGetState_t, 1064
state, 1064
TlvResult, 1064
unpack_dms_SetActivationStatusCallback
dms.h, 1443
unpack_dms_SetActivationStatusCallback_t, 1047
TlvResult, 1047
unpack_dms_SetCrashAction
dms.h, 1443
unpack_dms_SetCrashAction_t, 1047
notused, 1047
unpack_dms_SetCustFeature
dms.h, 1444
unpack_dms_SetCustFeature_t, 1047
TlvResult, 1048
unpack_dms_SetCustFeaturesV2
dms.h, 1444
unpack_dms_SetCustFeaturesV2_t, 1048
TlvResult, 1048
unpack_dms_SetEventReport
dms.h, 1444
unpack_dms_SetEventReport_ind
dms.h, 1445
unpack_dms_SetEventReport_ind_t, 1048
ActivationStatusTlv, 1049
OperatingModeTlv, 1049
TlvResult, 1049
unpack_dms_SetEventReport_t, 1049
TlvResult, 1049
unpack_dms_SetFirmwarePreference
dms.h, 1445
unpack_dms_SetFirmwarePreference_t, 1049
TlvResult, 1049
unpack_dms_SetPower
dms.h, 1445
unpack_dms_SetPower_t, 1050
TlvResult, 1050
unpack_dms_SetUSBComp
dms.h, 1446
unpack_dms_SetUSBComp_t, 1050
TlvResult, 1050
unpack_dms_UIMChangePIN
dms.h, 1452
unpack_dms_UIMGetControlKeyStatus
dms.h, 1453
unpack_dms_UIMGetControlKeyStatus_t, 1064
facilityState, 1065
TlvResult, 1065
unblockRetriesLeft, 1065
verifyRetriesLeft, 1065
unpack_dms_UIMGetICCID
dms.h, 1453
unpack_dms_UIMGetICCID_t, 1065
String, 1066
stringSize, 1066
TlvResult, 1066
unpack_dms_UIMGetPINStatus
dms.h, 1453
unpack_dms_UIMGetPINStatus_t, 1066
p1Status, 1067
p1UnblockRetriesLeft, 1067
p1VerifyRetriesLeft, 1067
p2Status, 1067
p2UnblockRetriesLeft, 1067
p2VerifyRetriesLeft, 1067
TlvResult, 1067
unpack_dms_UIMSetControlKeyProtection
dms.h, 1454
unpack_dms_UIMSetControlKeyProtection_t, 1067
TlvResult, 1068
verifyRetriesLeft, 1068
unpack_dms_UIMSetPINProtection
dms.h, 1454
unpack_dms_UIMSetPINProtection_t, 1068
TlvResult, 1068
unblockRetriesLeft, 1068
verifyRetriesLeft, 1068
unpack_dms_UIMUnblockControlKey
dms.h, 1454
unpack_dms_UIMUnblockControlKey_t, 1069
TlvResult, 1069

unblockRetriesLeft, 1069
unpack_dms_UIMUnblockPIN
dms.h, 1455
unpack_dms_UIMVerifyPIN
dms.h, 1455
unpack_dms_ValidateSPC
dms.h, 1456
unpack_fms_GetImagesPreference
fms.h, 1458
unpack_fms_GetImagesPreference_t, 1069
ImageListSize, 1070
pImageList, 1070
Tlvresult, 1070
unpack_fms_GetStoredImages
fms.h, 1459
unpack_fms_GetStoredImages_t, 1070
imageList, 1070
imagelistSize, 1070
Tlvresult, 1070
unpack_fms_SetImagesPreference
fms.h, 1459
unpack_fms_SetImagesPreference_t, 1070
ImageTypes, 1071
ImageTypesSize, 1071
Tlvresult, 1071
unpack_loc_BestAvailPos_Ind
loc.h, 1469
unpack_loc_BestAvailPos_Ind_t, 1071
pAltitudeWrtEllipsoid, 1075
pAltitudeWrtMeanSeaLevel, 1075
pGpsTime, 1075
pHeading, 1075
pHeadingUnc, 1075
pHorCirConf, 1075
pHorEllpConf, 1075
pHorReliability, 1075
pHorUncCircular, 1076
pHorUncEllipseOrientAzimuth, 1076
pHorUncEllipseSemiMajor, 1076
pHorUncEllipseSemiMinor, 1076
pLatitude, 1076
pLongitude, 1076
pMagneticDeviation, 1076
pPrecisionDilution, 1076
pSensorDataUsage, 1076
pSpeedHorizontal, 1076
pSpeedUnc, 1076
pSpeedVertical, 1076
pSpeedVerticalUnc, 1076
pSvUsedforFix, 1076
pTechnologyMask, 1076
pTimeSrc, 1076
pTimeUnc, 1076
pTimestampUtc, 1076
pVertConfidence, 1076
pVertReliability, 1076
pVertUnc, 1076
pXid, 1076
status, 1076
Tlvresult, 1076
unpack_loc_Delete_Assist_Data_t, 1076
Tlvresult, 1077
unpack_loc_DeleteAssistData
loc.h, 1469
unpack_loc_DeleteAssistData_Ind
loc.h, 1470
unpack_loc_DeleteAssistData_Ind_t, 1077
status, 1077
Tlvresult, 1078
unpack_loc_EngineState_Ind
loc.h, 1470
unpack_loc_EngineState_Ind_t, 1078
engineState, 1078
Tlvresult, 1078
unpack_loc_EventNMEA_Ind
loc.h, 1470
unpack_loc_EventNMEA_Ind_t, 1078
NMEAData, 1079
Tlvresult, 1079
unpack_loc_EventRegister
loc.h, 1471
unpack_loc_EventRegister_t, 1079
Tlvresult, 1079
unpack_loc_GetServer_Ind
loc.h, 1471
unpack_loc_GetServer_Ind_t, 1079
pIPv4AddrInfo, 1080
pIPv6AddrInfo, 1080
pURL, 1080
serverStatus, 1080
serverType, 1080
Tlvresult, 1080
unpack_loc_GnssSvInfo_Ind
loc.h, 1471
unpack_loc_GnssSvInfo_Ind_t, 1081
altitudeAssumed, 1081
pSatelliteInfo, 1081
Tlvresult, 1081
unpack_loc_PositionRpt_Ind
loc.h, 1472
unpack_loc_PositionRpt_Ind_t, 1081
pAltitudeAssumed, 1086
pAltitudeWrtEllipsoid, 1086
pAltitudeWrtMeanSeaLevel, 1086
pFixId, 1086
pGpsTime, 1086
pHeading, 1086
pHeadingUnc, 1087
pHorConfidence, 1087
pHorReliability, 1087
pHorUncCircular, 1087
pHorUncEllipseOrientAzimuth, 1087
pHorUncEllipseSemiMajor, 1087
pHorUncEllipseSemiMinor, 1087
pLatitude, 1087
pLeapSeconds, 1087

pLongitude, 1087
pMagneticDeviation, 1087
pPrecisionDilution, 1087
pSensorDataUsage, 1087
pSpeedHorizontal, 1087
pSpeedUnc, 1087
pSpeedVertical, 1087
pSvUsedforFix, 1087
pTechnologyMask, 1087
pTimeSrc, 1087
pTimeUnc, 1087
pTimestampUtc, 1087
pVertConfidence, 1087
pVertReliability, 1087
pVertUnc, 1087
sessionId, 1087
sessionStatus, 1087
Tlvresult, 1087
unpack_loc_SLQSLOCGetBestAvailPos
loc.h, 1473
unpack_loc_SLQSLOCGetBestAvailPos_t, 1090
Tlvresult, 1090
unpack_loc_SLQSLOCGetServer
loc.h, 1474
unpack_loc_SLQSLOCInjectPosition
loc.h, 1474
unpack_loc_SLQSLOCInjectSensorData
loc.h, 1474
unpack_loc_SLQSLOCInjectUTCTime
loc.h, 1474
unpack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1475
unpack_loc_SetExtPowerConfig_Ind
loc.h, 1472
unpack_loc_SetExtPowerConfig_Ind_t, 1088
status, 1088
Tlvresult, 1088
unpack_loc_SetExtPowerState
loc.h, 1472
unpack_loc_SetExtPowerState_t, 1088
Tlvresult, 1089
unpack_loc_SetOperationMode
loc.h, 1473
unpack_loc_SetOperationMode_Ind
loc.h, 1473
unpack_loc_SetOperationMode_Ind_t, 1089
status, 1089
Tlvresult, 1089
unpack_loc_SetOperationMode_t, 1089
Tlvresult, 1090
unpack_loc_Start
loc.h, 1475
unpack_loc_Start_t, 1090
Tlvresult, 1090
unpack_loc_Stop
loc.h, 1475
unpack_loc_Stop_t, 1090
Tlvresult, 1091
unpack_nas_GetACCOLC
nas.h, 1501
unpack_nas_GetANAAAAuthenticationStatus
nas.h, 1501
unpack_nas_GetCDMANetworkParameters
nas.h, 1501
unpack_nas_GetCDMANetworkParameters_t, 1091
Application, 1093
Broadcast, 1093
CustomSCP, 1093
ForceRev0, 1093
Protocol, 1093
RegForeignNID, 1093
RegForeignSID, 1093
RegHomeSID, 1093
Roaming, 1093
SCI, 1093
SCM, 1093
unpack_nas_GetHomeNetwork
nas.h, 1502
unpack_nas_GetHomeNetwork3GPP2
nas.h, 1502
unpack_nas_GetHomeNetwork3GPP2_t, 1093
nameSize, 1095
pMCC, 1095
pMNC, 1095
pNID, 1095
pName, 1095
pNw2DescDisp, 1095
pNw2DescEnc, 1095
pNw2DescLen, 1095
pNw2MCC, 1095
pNw2MNC, 1095
pNw2Name, 1095
pSID, 1095
unpack_nas_GetHomeNetwork_t, 1095
mcc, 1096
mnc, 1096
name, 1096
nid, 1096
sid, 1096
unpack_nas_GetNetworkPreference
nas.h, 1502
unpack_nas_GetNetworkPreference_t, 1096
ActiveTechPref, 1097
Duration, 1097
PersistentTechPref, 1097
Tlvresult, 1097
unpack_nas_GetRFInfo
nas.h, 1503
unpack_nas_GetRFInfo_t, 1097
instancesSize, 1098
RFBandInfoElements, 1098
unpack_nas_GetServingNetwork
nas.h, 1503
unpack_nas_GetServingNetwork_t, 1098
CSDomain, 1100
DataCaps, 1100

DataCapsLen, 1100
MCC, 1100
MNC, 1100
Name, 1100
nameSize, 1100
PSDomain, 1100
RAN, 1100
RadioIfaces, 1100
RadioIfacesSize, 1100
RegistrationState, 1100
Roaming, 1100
unpack_nas_GetServingNetworkCapabilities
nas.h, 1504
unpack_nas_GetServingNetworkCapabilities_t, 1100
DataCaps, 1101
DataCapsLen, 1101
unpack_nas_GetSignalStrengths
nas.h, 1504
unpack_nas_GetSignalStrengths_t, 1101
len, 1101
radio, 1101
rssi, 1101
unpack_nas_InitiateDomainAttach
nas.h, 1504
unpack_nas_PerformNetworkScan
nas.h, 1505
unpack_nas_PerformNetworkScan_t, 1102
p3GppNetworkInfoInstances, 1102
p3GppNetworkInstanceSize, 1103
pPCSInstance, 1103
pPCSInstanceSize, 1103
pRATINstance, 1103
pRATInstanceSize, 1103
pScanResult, 1103
unpack_nas_SLQSConfigSigInfo
nas.h, 1508
unpack_nas_SLQSGetErrorRate
nas.h, 1508
unpack_nas_SLQSGetErrorRate_t, 1106
pCDMAFrameErrRate, 1107
pGSMBER, 1107
pHDRPackErrRate, 1107
pWCDMABER, 1107
unpack_nas_SLQSGetNetworkTime
nas.h, 1509
unpack_nas_SLQSGetNetworkTime_t, 1108
p3GPP2TimeInfo, 1108
p3GPPTimeInfo, 1108
unpack_nas_SLQSGetOperatorNameData
nas.h, 1509
unpack_nas_SLQSGetOperatorNameData_t, 1108
pNITZInformation, 1109
pOperatorNameString, 1109
pOperatorPLMNList, 1109
pPLMNNetworkName, 1109
pSrvcProviderName, 1109
unpack_nas_SLQSGetPLMNName
nas.h, 1510
unpack_nas_SLQSGetPLMNName_t, 1109
longName, 1109
longNameCI, 1109
longNameEn, 1110
longNameLen, 1110
longNameSB, 1110
shortName, 1110
shortNameCI, 1110
shortNameEn, 1110
shortNameLen, 1110
shortNameSB, 1110
spn, 1110
spnEncoding, 1110
spnLength, 1110
unpack_nas_SLQSGetServingSystem
nas.h, 1510
unpack_nas_SLQSGetServingSystem_t, 1110
BasestationID, 1113
BasestationLatitude, 1113
BasestationLongitude, 1113
CDMASystemInfoExt, 1113
CallBarStatus, 1113
CellID, 1113
ConcSvcInfo, 1113
CurrentPLMN, 1113
DTMInd, 1113
DataSrvCapabilities, 1113
DefaultRoamInd, 1113
DetailedSvcInfo, 1113
Gpp2TimeZone, 1113
GppNetworkDSTAdjustment, 1113
GppTimeZone, 1113
HdrPersonality, 1113
Lac, 1113
NetworkID, 1113
PRLInd, 1114
RoamIndicatorVal, 1114
RoamingIndicatorList, 1114
ServingSystem, 1114
SystemID, 1114
TrackAreaCode, 1114
unpack_nas_SLQSGetSignalStrength
nas.h, 1510
unpack_nas_SLQSGetSignalStrength_t, 1114
ecioList, 1116
ecioListLen, 1116
errorRateList, 1116
errorRateListLen, 1116
Io, 1116
ltersrp, 1116
ltesnr, 1116
rsrqInfo, 1116
rxSignalStrengthList, 1116
rxSignalStrengthListLen, 1116
signalStrengthReqMask, 1116
sinr, 1116
unpack_nas_SLQSGetSysInfo
nas.h, 1511

unpack_nas_SLQSGetSysInfo_t, 1116
pAddCDMASysInfo, 1118
pAddGSMSysInfo, 1118
pAddHDRSysInfo, 1118
pAddLTESysInfo, 1118
pCDMASrvStatusInfo, 1119
pCDMASysInfo, 1119
pGSMCallBarringSysInfo, 1119
pGSMCipherDomainSysInfo, 1119
pGSMSrvStatusInfo, 1119
pGSMSysInfo, 1119
pHDRSrvStatusInfo, 1119
pHDRSysInfo, 1119
pLTESrvStatusInfo, 1119
pLTESysInfo, 1119
pLTEVoiceSupportSysInfo, 1119
pWCDMASysInfo, 1119
unpack_nas_SLQSGetSysSelectionPref
nas.h, 1511
unpack_nas_SLQSGetSysSelectionPref_t, 1119
pBandPref, 1123
pEmerMode, 1123
pGWAcqOrderPref, 1123
pLTEBandPref, 1123
pModePref, 1123
pNetSelPref, 1123
pPRLPref, 1123
pRoamPref, 1123
pSrvDomainPref, 1123
unpack_nas_SLQSInitiateNetworkRegistration
nas.h, 1511
unpack_nas_SLQSNASSwiGetChannelLock
nas.h, 1515
unpack_nas_SLQSNASSwiGetChannelLock_t, 1130
unpack_nas_SLQSNASSwiSetChannelLock
nas.h, 1516
unpack_nas_SLQSNasConfigSigInfo2
nas.h, 1512
unpack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1512
unpack_nas_SLQSNasGet3GPP2Subscription_t, 1123
pDirNum, 1124
unpack_nas_SLQSNasGetCellLocationInfo
nas.h, 1512
unpack_nas_SLQSNasGetCellLocationInfo_t, 1124
pCDMAInfo, 1125
pLTEInfoInterfreq, 1125
pLTEInfoIntrafreq, 1125
pUMTSInfo, 1125
unpack_nas_SLQSNasGetHDRColorCode
nas.h, 1513
unpack_nas_SLQSNasGetHDRColorCode_t, 1126
pColorCode, 1126
unpack_nas_SLQSNasGetSigInfo
nas.h, 1513
unpack_nas_SLQSNasGetSigInfo_t, 1126
CDMASSInfo, 1127
GSMSSInfo, 1127
HDRSSInfo, 1127
LTESSInfo, 1127
unpack_nas_SLQSNasGetTxRxInfo
nas.h, 1513
unpack_nas_SLQSNasGetTxRxInfo_t, 1127
pRXChain0Info, 1127
pRXChain1Info, 1127
pTXInfo, 1127
unpack_nas_SLQSNasIndicationRegisterExt
nas.h, 1514
unpack_nas_SLQSNasNetworkTimeCallBack_ind
nas.h, 1514
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 1128
pDayltSavAdj, 1128
pRadioInterface, 1128
pTimeZone, 1128
universalTime, 1128
unpack_nas_SLQSNasSigInfoCallback_ind
nas.h, 1514
unpack_nas_SLQSNasSigInfoCallback_ind_t, 1128
pRscp, 1129
unpack_nas_SLQSNasSwiIndicationRegister
nas.h, 1515
unpack_nas_SLQSNasSwiModemStatus
nas.h, 1515
unpack_nas_SLQSNasSwiModemStatus_t, 1130
commonInfo, 1131
pLTEInfo, 1131
unpack_nas_SLQSNasSwiOTAMessageCallback_ind
nas.h, 1516
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 1131
Info, 1131
Tlvresult, 1131
unpack_nas_SLQSNasSysInfoCallback_ind
nas.h, 1516
unpack_nas_SLQSNasTimerCallback_ind
nas.h, 1517
unpack_nas_SLQSNasTimerCallback_ind_t, 1131
t3396_apn, 1132
t3396_plmn_id, 1132
t3396_val, 1132
unpack_nas_SLQSSetBandPreference
nas.h, 1517
unpack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1517
unpack_nas_SLQSSetSysSelectionPref
nas.h, 1517
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
nas.h, 1518
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 1132
Info, 1132
Tlvresult, 1132
unpack_nas_SLQSSwiGetHDRPersonality
nas.h, 1518
unpack_nas_SLQSSwiGetHDRPersonality_t, 1132

pCurrentPersonality, 1133
pPersonalityListLength, 1133
pProtocolSubtypeElement, 1133
unpack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1518
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 1133
pAppSubType, 1133
pCurrentPrsnlty, 1133
pPersonalityListLength, 1133
unpack_nas_SLQSSwiGetHRPDStats
nas.h, 1519
unpack_nas_SLQSSwiGetHRPDStats_t, 1134
pPilotSetData, 1134
pUATI, 1134
unpack_nas_SLQSSwiGetLteCQI
nas.h, 1519
unpack_nas_SLQSSwiGetLteCQI_t, 1134
ValidityCW0, 1135
ValidityCW1, 1135
unpack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1519
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 1135
pSccRxInfo, 1135
unpack_nas_SLQSSwiNetworkDebug
nas.h, 1520
unpack_nas_SLQSSwiNetworkDebug_t, 1135
pDataStatusDetail, 1136
pDeviceConfigDetail, 1136
pNetworkStat1x, 1136
pNetworkStatEVDO, 1136
pObjectVer, 1136
unpack_nas_SLQSSwiPSDetach
nas.h, 1520
unpack_nas_SLQSSysInfoCallback_ind_t, 1136
pGSMSysInfo, 1139
pHDRSysInfo, 1139
pLTESysInfo, 1139
pSysInfoNoChange, 1139
unpack_nas_SetACCOLC
nas.h, 1505
unpack_nas_SetCDMANetworkParameters
nas.h, 1505
unpack_nas_SetDataCapabilitiesCallback_ind
nas.h, 1506
unpack_nas_SetDataCapabilitiesCallback_ind_t, 1103
dataCaps, 1103
dataCapsSize, 1103
unpack_nas_SetEventReportInd
nas.h, 1506
unpack_nas_SetEventReportInd_t, 1103
RFTlv, 1104
RRTlv, 1104
SLQSSSTlv, 1104
SSTlv, 1104
unpack_nas_SetLURejectCallback
nas.h, 1506
unpack_nas_SetNasLTECphyCaIndCallback_ind
nas.h, 1506
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 1104
unpack_nas_SetNetworkPreference
nas.h, 1507
unpack_nas_SetNetworkPreference_t, 1105
Tlvresult, 1105
unpack_nas_SetRFInfoCallback
nas.h, 1507
unpack_nas_SetRoamingIndicatorCallback_ind
nas.h, 1507
unpack_nas_SetRoamingIndicatorCallback_ind_t, 1105
roaming, 1106
unpack_nas_SetServingSystemCallback_ind
nas.h, 1508
unpack_nas_SetServingSystemCallback_ind_t, 1106
SSInfo, 1106
Tlvresult, 1106
unpack_nas_SlqsGetLTECphyCAInfo
nas.h, 1509
unpack_nas_SlqsGetLTECphyCAInfo_t, 1107
LTECphyCaInfo, 1108
Tlvresult, 1108
unpack_omaDmConfigTlv_t, 1139
alertmsg, 1140
alertmsglength, 1140
state, 1140
userInputReq, 1140
userInputTimeout, 1140
unpack_omaDmFotaTlv_t, 1140
description, 1142
descriptionlength, 1142
fwdloadsize, 1142
fwloadComplete, 1142
namelength, 1142
package_name, 1142
sessionType, 1142
severity, 1142
state, 1142
updateCompleteStatus, 1142
userInputReq, 1142
userInputTimeout, 1142
version, 1142
versionlength, 1142
unpack_omaDmNotificationsTlv_t, 1142
notification, 1143
sessionStatus, 1143
unpack_pds_ForceXTRADownload
pds.h, 1532
unpack_pds_GetPDSDefaults
pds.h, 1532
unpack_pds_GetPDSDefaults_t, 1143
pAccuracy, 1144
pInterval, 1144
pOperation, 1144
pTimeout, 1144
unpack_pds_GetPDSState
pds.h, 1532
unpack_pds_GetPDSState_t, 1144

pEnabledStatus, 1144
pTrackingStatus, 1144
unpack_pds_GetPortAutomaticTracking
pds.h, 1533
unpack_pds_GetPortAutomaticTracking_t, 1145
pbAuto, 1145
unpack_pds_GetServiceAutomaticTracking
pds.h, 1533
unpack_pds_GetServiceAutomaticTracking_t, 1145
pbAuto, 1145
unpack_pds_GetXTRAAutomaticDownload
pds.h, 1533
unpack_pds_GetXTRAAutomaticDownload_t, 1145
pInterval, 1146
pbEnabled, 1146
unpack_pds_GetXTRANetwork
pds.h, 1534
unpack_pds_GetXTRANetwork_t, 1146
pPreference, 1146
unpack_pds_GetXTRAValidity
pds.h, 1534
unpack_pds_GetXTRAValidity_t, 1146
pDuration, 1147
pGPSWeek, 1147
pGPSWeekOffset, 1147
unpack_pds_PDSInjectTimeReference
pds.h, 1535
unpack_pds_ResetPDSData
pds.h, 1535
unpack_pds_SLQSGetAGPSConfig
pds.h, 1538
unpack_pds_SLQSGetAGPSConfig_t, 1149
pServerAddress, 1150
pServerPort, 1150
pServerURL, 1150
unpack_pds_SLQSGetGPSStateInfo
pds.h, 1538
unpack_pds_SLQSGetGPSStateInfo_t, 1150
Altitude, 1153
EngineState, 1153
HorizontalUncertainty, 1154
Iono_valid, 1154
Latitude, 1154
Longitude, 1154
Time_uncert_ms, 1154
ValidMask, 1154
VerticalUncertainty, 1154
unpack_pds_SLQSPDSDeterminePosition
pds.h, 1539
unpack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1539
unpack_pds_SLQSPDSInjectPositionData
pds.h, 1539
unpack_pds_SLQSSetAGPSConfig
pds.h, 1540
unpack_pds_SLQSSetPositionMethodState
pds.h, 1540
unpack_pds_SetEventReport_Ind
pds.h, 1535
unpack_pds_SetEventReport_Ind_t, 1147
dLatitude, 1148
dLongitude, 1148
has_PositionDataNMEA, 1149
has_SessionStatus, 1149
has_dLatitude, 1148
has_dLongitude, 1148
has_posSrc, 1149
posSrc, 1149
PositionDataNMEA, 1149
SessionStatus, 1149
unpack_pds_SetEventReportCallback
pds.h, 1536
unpack_pds_SetPDSDefaults
pds.h, 1536
unpack_pds_SetPDSState
pds.h, 1536
unpack_pds_SetPdsState_Ind
pds.h, 1536
unpack_pds_SetPdsState_Ind_t, 1149
EnabledStatus, 1149
TlvPresent, 1149
TrackingStatus, 1149
unpack_pds_SetPortAutomaticTracking
pds.h, 1537
unpack_pds_SetServiceAutomaticTracking
pds.h, 1537
unpack_pds_SetXTRAAutomaticDownload
pds.h, 1537
unpack_pds_SetXTRANetwork
pds.h, 1538
unpack_pds_StartPDSTrackingSessionExt
pds.h, 1540
unpack_pds_StopPDSTrackingSession
pds.h, 1541
unpack_qmi_t, 1154
msgid, 1155
type, 1155
xid, 1155
unpack_qos_IPv4Addr_t, 1155
addr, 1155
subnetMask, 1156
unpack_qos_IPv6Addr_t, 1156
addr, 1156
prefixLen, 1156
unpack_qos_IPv6TrafCls_t, 1156
mask, 1156
val, 1157
unpack_qos_Port_t, 1157
port, 1157
range, 1157
unpack_qos_QosFlowInfo_t, 1157
BearerID, 1159
is_RxQFlowGranted_Available, 1159
is_TxQFlowGranted_Available, 1159
NumRxFilters, 1159
NumTxFilters, 1159

QFlowState, 1159
RxQFilter, 1159
RxQFlowGranted, 1159
TxQFilter, 1159
TxQFlowGranted, 1159
unpack_qos_QosFlowInfoState_t, 1159
id, 1159
isNewFlow, 1159
state, 1160
unpack_qos_SLQSQosGetNetworkStatus
qos.h, 1976
unpack_qos_SLQSQosGetNetworkStatus_t, 1160
NWQoSStatus, 1160
unpack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 1977
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 1160
ambr_dl, 1161
ambr_dl_ext, 1161
ambr_dl_ext2, 1161
ambr_ul, 1161
ambr_ul_ext, 1161
ambr_ul_ext2, 1161
apnId, 1161
unpack_qos_SLQSQosSwiReadDataStats
qos.h, 1977
unpack_qos_SLQSQosSwiReadDataStats_t, 1161
apnId, 1162
numQosFlow, 1162
qosFlow, 1162
total_rx_bytes, 1162
total_rx_pkt, 1162
total_tx_bytes, 1162
total_tx_pkt, 1162
unpack_qos_SLQSSetQosEventCallback
qos.h, 1978
unpack_qos_SLQSSetQosEventCallback_ind
qos.h, 1978
unpack_qos_SLQSSetQosEventCallback_ind_t, 1163
NumFlows, 1163
QosFlowInfo, 1163
unpack_qos_SLQSSetQosNWStatusCallback_ind
qos.h, 1978
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 1163
status, 1163
unpack_qos_SLQSSetQosPriEventCallback_ind
qos.h, 1979
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 1164
event, 1164
unpack_qos_SLQSSetQosStatusCallback_ind
qos.h, 1980
unpack_qos_SLQSSetQosStatusCallback_ind_t, 1164
event, 1165
id, 1165
reason, 1165
status, 1165
unpack_qos_Tos_t, 1173
mask, 1174
val, 1174
unpack_qos_dataRate_t, 1155
dataRateMax, 1155
guaranteedRate, 1155
unpack_qos_pktErrRate_t, 1157
exponent, 1157
multiplier, 1157
unpack_qos_swiQosFilter_t, 1165
EspSpi, 1167
IPv4DstAddr, 1168
IPv4SrcAddr, 1168
IPv4Tos, 1168
IPv6DstAddr, 1168
IPv6Label, 1168
IPv6SrcAddr, 1168
IPv6TrafCls, 1168
Id, 1167
index, 1168
is_EspSpi_Available, 1168
is_IPv4DstAddr_Available, 1168
is_IPv4SrcAddr_Available, 1168
is_IPv4Tos_Available, 1168
is_IPv6DstAddr_Available, 1168
is_IPv6Label_Available, 1168
is_IPv6SrcAddr_Available, 1168
is_IPv6TrafCls_Available, 1168
is_Id_Available, 1168
is_NxtHdrProto_Available, 1168
is_Precedence_Available, 1168
is_TCPDstPort_Available, 1168
is_TCPSrcPort_Available, 1168
is_TranDstPort_Available, 1168
is_TranSrcPort_Available, 1168
is_UDPDstPort_Available, 1168
is_UDPSrcPort_Available, 1168
NxtHdrProto, 1168
Precedence, 1168
TCPDstPort, 1168
TCPSrcPort, 1169
TranDstPort, 1169
TranSrcPort, 1169
UDPDstPort, 1169
UDPSrcPort, 1169
version, 1169
unpack_qos_swiQosFlow_t, 1169
DataRate, 1172
index, 1172
is_DataRate_Available, 1172
is_Jitter_Available, 1172
is_Latency_Available, 1172
is_LteQci_Available, 1172
is_MaxAllowedPktSz_Available, 1172
is_MinPolicedPktSz_Available, 1172
is_PktErrRate_Available, 1172
is_ProfileId3GPP2_Available, 1172
is_TokenBucket_Available, 1172

is_TrafficClass_Available, 1172
is_val_3GPP2Pri_Available, 1172
is_val_3GPPImCn_Available, 1172
is_val_3GPPSigInd_Available, 1172
Jitter, 1172
Latency, 1172
LteQci, 1172
MaxAllowedPktSz, 1172
MinPolicedPktSz, 1172
PktErrRate, 1173
ProfileId3GPP2, 1173
TokenBucket, 1173
TrafficClass, 1173
val_3GPP2Pri, 1173
val_3GPPImCn, 1173
val_3GPPResResidualBER, 1173
val_3GPPSigInd, 1173
val_3GPPTraHdlPri, 1173
unpack_qos_tokenBucket_t, 1173
bucketSz, 1173
peakRate, 1173
tokenRate, 1173
unpack_result_code_only
common.h, 1404
unpack_rms_GetSMSWake
rms.h, 1981
unpack_rms_GetSMSWake_t, 1175
enabled, 1175
wake_mask, 1175
unpack_rms_SetSMSWake
rms.h, 1982
unpack_sar_SLQSGetRfSarState
sar.h, 1983
unpack_sar_SLQSGetRfSarState_t, 1176
pSarRFState, 1177
unpack_sar_SLQSSetRfSarState
sar.h, 1983
unpack_sms_GetSMSCAddress
sms.h, 1998
unpack_sms_GetSMSCAddress_t, 1177
addressSize, 1177
pSMSCAddress, 1177
pSMSCType, 1178
typeSize, 1178
unpack_sms_SLQSDeleteSMS
sms.h, 2000
unpack_sms_SLQSDeleteSMS_t, 1180
unpack_sms_SLQSGetIndicationRegister
sms.h, 2000
unpack_sms_SLQSGetIndicationRegister_t, 1180
pGetIndicationRegInfo, 1180
unpack_sms_SLQSGetMessageWaiting
sms.h, 2000
unpack_sms_SLQSGetMessageWaiting_t, 1180
pGetMsgWaitingInfoResp, 1180
unpack_sms_SLQSGetSMS
sms.h, 2001
unpack_sms_SLQSGetSMS_t, 1181
message, 1181
messageFormat, 1181
messageSize, 1181
messageTag, 1181
unpack_sms_SLQSGetSMSList
sms.h, 2001
unpack_sms_SLQSGetSMSList_t, 1182
messageList, 1182
messageListSize, 1182
unpack_sms_SLQSGetSmsBroadcastConfig
sms.h, 2001
unpack_sms_SLQSGetSmsBroadcastConfig_t, 1181
pBroadcastConfig, 1182
pCDMABroadcastConfig, 1182
unpack_sms_SLQSGetTransLayerInfo
sms.h, 2002
unpack_sms_SLQSGetTransLayerInfo_t, 1182
pGetTransLayerInfo, 1183
unpack_sms_SLQSGetTransNWRegInfo
sms.h, 2002
unpack_sms_SLQSGetTransNWRegInfo_t, 1183
unpack_sms_SLQSModifySMSStatus
sms.h, 2002
unpack_sms_SLQSModifySMSStatus_t, 1183
unpack_sms_SLQSSendAsyncSMS
sms.h, 2003
unpack_sms_SLQSSetIndicationRegister
sms.h, 2003
unpack_sms_SLQSSetSmsBroadcastActivation
sms.h, 2003
unpack_sms_SLQSSetSmsBroadcastConfig
sms.h, 2004
unpack_sms_SLQSSetSmsStorage
sms.h, 2004
unpack_sms_SLQSSmsGetMaxStorageSize
sms.h, 2004
unpack_sms_SLQSSmsGetMaxStorageSize_t, 1183
pMaxStorageSizeResp, 1184
unpack_sms_SLQSSmsGetMessageProtocol
sms.h, 2005
unpack_sms_SLQSSmsGetMessageProtocol_t, 1184
pMessageProtocol, 1184
unpack_sms_SLQSSmsSetRoutes
sms.h, 2005
unpack_sms_SLQSSwiGetSMSStorage
sms.h, 2005
unpack_sms_SLQSSwiGetSMSStorage_t, 1184
pSmsStorage, 1185
unpack_sms_SLQSWmsMemoryFullCallBack_ind
sms.h, 2006
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 1185
messageMode, 1185
storageType, 1185
unpack_sms_SaveSMS
sms.h, 1998
unpack_sms_SaveSMS_t, 1178
pMessageIndex, 1178

unpack_sms_SendSMS
sms.h, 1998
unpack_sms_SendSMS_t, 1178
messageFailureCode, 1178
messageID, 1178
unpack_sms_SetNewSMSCallback
sms.h, 1999
unpack_sms_SetNewSMSCallback_ind
sms.h, 1999
unpack_sms_SetNewSMSCallback_ind_t, 1178
ETWSTlv, 1179
IMSTlv, 1179
MMTlv, 1179
NewMMTlv, 1179
SMSCTlv, 1179
TRMessageTlv, 1180
unpack_sms_SetNewSMSCallback_t, 1180
unpack_sms_SetSMSCAddress
sms.h, 1999
unpack_swiloc_SwiLocGetAutoStart
swiloc.h, 2008
unpack_swiloc_SwiLocGetAutoStart_t, 1185
fix_rate, 1186
fix_rate_reported, 1187
fix_type, 1187
fix_type_reported, 1187
function, 1187
function_reported, 1187
max_dist, 1187
max_dist_reported, 1187
max_time, 1187
max_time_reported, 1187
unpack_swiloc_SwiLocSetAutoStart
swiloc.h, 2009
unpack_swioma_SLQSOMADMAlertCallback
swioma.h, 2014
unpack_swioma_SLQSOMADMAlertCallback_ind
swioma.h, 2015
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 1187
eventType, 1187
unpack_swioma_SLQSOMADMCancelSession
swioma.h, 2015
unpack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 2016
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1188
Date, 1190
DateLength, 1190
PkgDescLength, 1190
PkgDescription, 1190
PkgName, 1190
PkgNameLength, 1190
RetryCount, 1190
SessionState, 1190
SessionType, 1190
Severity, 1190
Source, 1190
SourceLength, 1190
Status, 1190
Time, 1190
TimeLength, 1190
UpdateCompleteStatus, 1190
unpack_swioma_SLQSOMADMGetSettings
swioma.h, 2016
unpack_swioma_SLQSOMADMGetSettings_t, 1190
Autosdm, 1192
FOTAdownload, 1192
FwAutoCheck, 1192
unpack_swioma_SLQSOMADMSendSelection
swioma.h, 2017
unpack_swioma_SLQSOMADMSetSettings
swioma.h, 2017
unpack_swioma_SLQSOMADMStartSession
swioma.h, 2017
unpack_swioma_SLQSOMADMStartSession_t, 1192
FwAvailability, 1192
unpack_uim_ChangePin
uim.h, 2029
unpack_uim_ChangePin_t, 1192
pEncryptedPIN1, 1193
pIndicationToken, 1193
pRemainingRetries, 1193
Tlvresult, 1193
unpack_uim_GetCardStatus
uim.h, 2029
unpack_uim_GetCardStatus_t, 1193
pCardStatus, 1194
pHotSwapStatus, 1194
Tlvresult, 1194
unpack_uim_ReadTransparent
uim.h, 2029
unpack_uim_ReadTransparent_t, 1194
pCardResult, 1194
pEncryptedData, 1194
pIndicationToken, 1194
pReadResult, 1194
Tlvresult, 1195
unpack_uim_SLQSUIMAuthenticate
uim.h, 2030
unpack_uim_SLQSUIMAuthenticate_t, 1196
pAuthenticateResult, 1196
pCardResult, 1196
pIndicationToken, 1196
unpack_uim_SLQSUIMDepersonalization
uim.h, 2031
unpack_uim_SLQSUIMDepersonalization_t, 1197
pRemainingRetries, 1197
unpack_uim_SLQSUIMEventRegister
uim.h, 2031
unpack_uim_SLQSUIMEventRegister_t, 1197
eventMask, 1197
unpack_uim_SLQSUIMGetConfiguration
uim.h, 2031
unpack_uim_SLQSUIMGetConfiguration_t, 1197
pAutoSelection, 1198
pHaltSubscription, 1198

pPersonalizationStatus, 1198
unpack_uim_SLQSUIMGetFileAttributes
uim.h, 2032
unpack_uim_SLQSUIMGetFileAttributes_t, 1198
pCardResult, 1199
pFileAttributes, 1199
pIndicationToken, 1199
unpack_uim_SLQSUIMGetSlotsStatus
uim.h, 2032
unpack_uim_SLQSUIMGetSlotsStatus_t, 1199
pNumberOfPhySlot, 1199
pUimSlotsStatus, 1199
unpack_uim_SLQSUIMPowerDown
uim.h, 2033
unpack_uim_SLQSUIMPowerUp
uim.h, 2033
unpack_uim_SLQSUIMRefreshComplete
uim.h, 2033
unpack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 2033
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 1199
pRefreshEvent, 1200
unpack_uim_SLQSUIMRefreshOK
uim.h, 2034
unpack_uim_SLQSUIMRefreshRegister
uim.h, 2034
unpack_uim_SLQSUIMReset
uim.h, 2034
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind
uim.h, 2035
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t, 1200
unpack_uim_SLQSUIMSwitchSlot
uim.h, 2035
unpack_uim_SetPinProtection
uim.h, 2030
unpack_uim_SetPinProtection_t, 1195
pEncryptedPIN1, 1195
pIndicationToken, 1195
pRemainingRetries, 1195
Tlvresult, 1195
unpack_uim_SetUimSlotStatusChangeCallback_ind
uim.h, 2030
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 1195
bNumberOfPhySlots, 1196
slotsstatusChange, 1196
unpack_uim_UnblockPin
uim.h, 2036
unpack_uim_UnblockPin_t, 1200
pEncryptedPIN1, 1201
pIndicationToken, 1201
pRemainingRetries, 1201
Tlvresult, 1201
unpack_uim_VerifyPin
uim.h, 2036
unpack_uim_VerifyPin_t, 1201
pEncryptedPIN1, 1201
pIndicationToken, 1201
pRemainingRetries, 1201
Tlvresult, 1201
unpack_voice_AnswerUSSD
voice.h, 2053
unpack_voice_CancelUSSD
voice.h, 2053
unpack_voice_OriginateUSSD
voice.h, 2053
unpack_voice_SLQSOriginateUSSD
voice.h, 2054
unpack_voice_SLQSOriginateUSSD_t, 1201
pAlphaIDInfo, 1202
pCCSuppsType, 1202
pCallId, 1202
pCcResultType, 1202
pUSSDInfo, 1202
pfailureCause, 1202
unpack_voice_SLQSVoiceALSSelectLine
voice.h, 2054
unpack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 2054
unpack_voice_SLQSVoiceAnswerCall
voice.h, 2055
unpack_voice_SLQSVoiceAnswerCall_t, 1202
pCallId, 1202
unpack_voice_SLQSVoiceBindSubscription
voice.h, 2055
unpack_voice_SLQSVoiceBurstDTMF
voice.h, 2055
unpack_voice_SLQSVoiceBurstDTMF_t, 1202
pCallID, 1203
unpack_voice_SLQSVoiceDialCall
voice.h, 2056
unpack_voice_SLQSVoiceDialCall_t, 1203
pAlphaIDInfo, 1203
pCCResultType, 1204
pCCSUPSType, 1204
pCallID, 1203
unpack_voice_SLQSVoiceEndCall
voice.h, 2056
unpack_voice_SLQSVoiceEndCall_t, 1204
pCallId, 1204
unpack_voice_SLQSVoiceGetAllCallInfo
voice.h, 2056
unpack_voice_SLQSVoiceGetAllCallInfo_t, 1204
pArrAlertingPattern, 1206
pArrAlertingType, 1206
pArrAlphaID, 1206
pArrCallEndReason, 1206
pArrCallInfo, 1206
pArrCalledPartyNum, 1206
pArrConnectPartyNum, 1206
pArrDiagInfo, 1206
pArrRedirPartyNum, 1206
pArrRemotePartyName, 1206
pArrRemotePartyNum, 1206
pArrSvcOption, 1206

pArrUUSInfo, 1207
pVoicePrivacy, 1207
unpack_voice_SLQSVoiceGetCLIP
voice.h, 2058
unpack_voice_SLQSVoiceGetCLIP_t, 1214
pAlphaIDInfo, 1215
pCCResType, 1215
pCallID, 1215
pFailCause, 1215
unpack_voice_SLQSVoiceGetCLIR
voice.h, 2059
unpack_voice_SLQSVoiceGetCLIR_t, 1215
pAlphaIDInfo, 1216
pCCResType, 1216
pCallID, 1216
pFailCause, 1216
unpack_voice_SLQSVoiceGetCNAP
voice.h, 2059
unpack_voice_SLQSVoiceGetCNAP_t, 1216
pAlphaIDInfo, 1217
pCCResType, 1217
pCallID, 1217
pFailCause, 1217
unpack_voice_SLQSVoiceGetCOLP
voice.h, 2059
unpack_voice_SLQSVoiceGetCOLP_t, 1217
pAlphaIDInfo, 1218
pCCResType, 1218
pCallID, 1218
pFailCause, 1219
unpack_voice_SLQSVoiceGetCOLR
voice.h, 2060
unpack_voice_SLQSVoiceGetCOLR_t, 1219
pAlphaIDInfo, 1220
pCCResType, 1220
pCallID, 1220
pFailCause, 1220
unpack_voice_SLQSVoiceGetCallBarring
voice.h, 2057
unpack_voice_SLQSVoiceGetCallBarring_t, 1207
pAlphaIDInfo, 1208
pCCResType, 1208
pCallID, 1208
pFailCause, 1208
pSvcClass, 1208
unpack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 2057
unpack_voice_SLQSVoiceGetCallForwardingStatus_t, 1208
pAlphaIDInfo, 1209
pCCResType, 1209
pCallID, 1209
pFailCause, 1209
unpack_voice_SLQSVoiceGetCallInfo
voice.h, 2058
unpack_voice_SLQSVoiceGetCallInfo_t, 1209
pAlertType, 1212
pAlertingPattern, 1212
pAlphaIDInfo, 1212
pCallInfo, 1212
pConnectNumInfo, 1212
pDiagInfo, 1212
pOTASPStatus, 1212
pRemotePartyName, 1212
pRemotePartyNum, 1212
pSrvOpt, 1212
pUUSInfo, 1212
pVoicePrivacy, 1212
unpack_voice_SLQSVoiceGetCallWaiting
voice.h, 2058
unpack_voice_SLQSVoiceGetCallWaiting_t, 1212
pAlphaIDInfo, 1213
pCCResType, 1213
pCallID, 1213
pFailCause, 1213
pSvcClass, 1213
unpack_voice_SLQSVoiceGetConfig
voice.h, 2060
unpack_voice_SLQSVoiceGetConfig_t, 1220
pAirTimerCnt, 1221
pAutoAnswerStat, 1221
pCurAMRConfig, 1221
pCurPrefVoiceSO, 1221
pCurVoiceDomainPref, 1221
pCurVoicePrivacyPref, 1222
pCurrTTYMode, 1221
pRoamTimerCnt, 1222
unpack_voice_SLQSVoiceIndicationRegister
voice.h, 2060
unpack_voice_SLQSVoiceManageCalls
voice.h, 2061
unpack_voice_SLQSVoiceManageCalls_t, 1222
pFailCause, 1222
unpack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 2061
unpack_voice_SLQSVoiceSendFlash
voice.h, 2061
unpack_voice_SLQSVoiceSendFlash_t, 1222
pCallID, 1222
unpack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 2062
unpack_voice_SLQSVoiceSetCallBarringPassword_t, 1222
pAlphaIDInfo, 1223
pCCResType, 1223
pCallID, 1223
pFailCause, 1224
unpack_voice_SLQSVoiceSetConfig
voice.h, 2062
unpack_voice_SLQSVoiceSetConfig_t, 1224
pAirTimerStatus, 1225
pAutoAnsStatus, 1225
pPrefVoiceSOStatus, 1225
pRoamTimerStatus, 1225
pTTYConfigStatus, 1225
pVoiceDomainPrefStatus, 1225

unpack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 2062
unpack_voice_SLQSVoiceSetSUPSService
voice.h, 2063
unpack_voice_SLQSVoiceSetSUPSService_t, 1225
pCallID, 1226
pFailCause, 1226
unpack_voice_SLQSVoiceStartContDTMF
voice.h, 2063
unpack_voice_SLQSVoiceStartContDTMF_t, 1226
pCallID, 1227
unpack_voice_SLQSVoiceStopContDTMF
voice.h, 2063
unpack_voice_SLQSVoiceStopContDTMF_t, 1227
callID, 1227
unpack_wds_DHCPv4ClientLease_ind
wds.h, 2089
unpack_wds_DHCPv4ClientLease_ind_t, 1227
DHCPv4LeaseOptTlv, 1228
DHCPv4LeaseStateTlv, 1228
IPv4AddrTlv, 1228
ProfileIdTlv, 1228
unpack_wds_DHCPv4ClientLeaseChange
wds.h, 2089
unpack_wds_GetAutoconnect
wds.h, 2090
unpack_wds_GetAutoconnect_t, 1228
psetting, 1228
unpack_wds_GetByteTotals
wds.h, 2090
unpack_wds_GetByteTotals_t, 1228
pRXTotalBytes, 1229
pTXTotalBytes, 1229
unpack_wds_GetConnectionRate
wds.h, 2090
unpack_wds_GetConnectionRate_t, 1229
currentChannelRXRate, 1229
currentChannelTXRate, 1229
maxChannelRXRate, 1229
maxChannelTXRate, 1229
unpack_wds_GetDataBearerTechnology
wds.h, 2091
unpack_wds_GetDataBearerTechnology_t, 1229
pDataBearer, 1230
unpack_wds_GetDefaultProfile
wds.h, 2091
unpack_wds_GetDefaultProfile_t, 1230
apnname, 1232
apnsize, 1232
auth, 1232
ipaddr, 1232
ipaddrv6, 1232
name, 1232
namesize, 1232
pdptype, 1232
pridns, 1232
pridnsv6, 1232
secdns, 1232
secdnsv6, 1232
username, 1232
usersize, 1232
unpack_wds_GetDefaultProfileNum
wds.h, 2091
unpack_wds_GetDefaultProfileNum_t, 1232
index, 1233
unpack_wds_GetDormancyState
wds.h, 2092
unpack_wds_GetDormancyState_t, 1233
dormancyState, 1233
unpack_wds_GetLastMobileIPError
wds.h, 2092
unpack_wds_GetLastMobileIPError_t, 1233
error, 1233
unpack_wds_GetMobileIP
wds.h, 2093
unpack_wds_GetMobileIP_t, 1234
mipMode, 1234
unpack_wds_GetMobileIPProfile
wds.h, 2093
unpack_wds_GetMobileIPProfile_t, 1234
AAASPI, 1235
AAAState, 1235
address, 1235
enabled, 1235
HASPI, 1235
HAState, 1236
NAI, 1236
naiSize, 1236
primaryHA, 1236
revTunneling, 1236
secondaryHA, 1236
unpack_wds_GetPacketStatistics
wds.h, 2093
unpack_wds_GetPacketStatistics_t, 1236
pRXDroppedCount, 1237
pRXOKBytesLastCall, 1237
pRXOkBytesCount, 1237
pRXPacketErrors, 1237
pRXPacketOverflows, 1237
pRXPacketSuccesses, 1237
pTXDroppedCount, 1237
pTXOKBytesLastCall, 1237
pTXOkBytesCount, 1237
pTXPacketErrors, 1237
pTXPacketOverflows, 1237
pTXPacketSuccesses, 1237
unpack_wds_GetPacketStatus
wds.h, 2094
unpack_wds_GetPacketStatus_t, 1237
rXDroppedCount, 1239
rXOKBytesLastCall, 1239
rXOkBytesCount, 1239
rXPacketErrors, 1239
rXPacketOverflows, 1239
rXPacketSuccesses, 1239
tXDroppedCount, 1239

tXOKBytesLastCall, 1239
tXOkBytesCount, 1239
tXPacketErrors, 1239
tXPacketOverflows, 1239
tXPacketSuccesses, 1239
unpack_wds_GetSessionDuration
wds.h, 2094
unpack_wds_GetSessionDuration_t, 1239
callDuration, 1239
unpack_wds_GetSessionState
wds.h, 2094
unpack_wds_GetSessionState_t, 1240
connectionStatus, 1240
unpack_wds_RMSetTransferStatistics
wds.h, 2095
unpack_wds_RMSetTransferStatistics_t, 1240
unpack_wds_RMTransferStatistics_ind
wds.h, 2095
unpack_wds_SLQSCreateProfile
wds.h, 2097
unpack_wds_SLQSCreateProfile_t, 1240
pCreateProfileOut, 1241
pProfileID, 1241
Tlvresult, 1241
unpack_wds_SLQSDeleteProfile
wds.h, 2098
unpack_wds_SLQSDeleteProfile_t, 1241
extendedErrorCode, 1241
unpack_wds_SLQSGet3GPPConfigItem
wds.h, 2098
unpack_wds_SLQSGet3GPPConfigItem_t, 1241
_3gppRelease, 1242
defaultPDNEnabled, 1242
LTEAttachProfile, 1242
profileList, 1243
unpack_wds_SLQSGetCurrDataSystemStat
wds.h, 2098
unpack_wds_SLQSGetCurrDataSystemStat_t, 1243
currNetworkInfo, 1243
networkInfoLen, 1243
prefNetwork, 1243
unpack_wds_SLQSGetCurrentChannelRate
wds.h, 2099
unpack_wds_SLQSGetCurrentChannelRate_t, 1243
max_channel_rx_rate, 1244
max_channel_tx_rate, 1244
unpack_wds_SLQSGetDUNCallInfo
wds.h, 2099
unpack_wds_SLQSGetDUNCallInfo_t, 1245
callEndReason, 1247
channelRate, 1247
connectionStatus, 1247
dataBearerTech, 1247
dormancyStatus, 1247
lastCallDataBearerTech, 1247
mdmCallDurationActive, 1247
rxOKBytesCount, 1247
txOKBytesCount, 1248
unpack_wds_SLQSGetDataBearerTechnology
wds.h, 2099
unpack_wds_SLQSGetDataBearerTechnology_t, 1244
curDataBearerTechnology, 1245
dataBearerMask, 1245
lastCallDataBearerTechnology, 1245
unpack_wds_SLQSGetProfileSettings
wds.h, 2100
unpack_wds_SLQSGetProfileSettings_t, 1248
pProfileSettings, 1248
ProfileType, 1248
Tlvresult, 1248
unpack_wds_SLQSGetRuntimeSettings
wds.h, 2100
unpack_wds_SLQSGetRuntimeSettings_t, 1248
APNName, 1249
Authentication, 1250
DomainList, 1250
GPRSGrantedQoS, 1250
GWAddressV4, 1250
IMCNflag, 1250
IPFamilyPreference, 1250
IPV6AddrInfo, 1250
IPV6GWAddrInfo, 1250
IPv4, 1250
Mtu, 1250
PDPType, 1250
PrimaryDNSV4, 1250
PrimaryDNSV6, 1250
ProfileID, 1250
ProfileName, 1250
SecondaryDNSV4, 1250
SecondaryDNSV6, 1250
ServerAddrList, 1250
SubnetMaskV4, 1250
Technology, 1250
UMTSGrantedQoS, 1250
Username, 1250
unpack_wds_SLQSModifyProfile
wds.h, 2100
unpack_wds_SLQSModifyProfile_t, 1250
pExtErrorCode, 1251
unpack_wds_SLQSResetPacketStatics
wds.h, 2101
unpack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 2102
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 1254
pHwConfig, 1254
unpack_wds_SLQSSGetLoopback
wds.h, 2103
unpack_wds_SLQSSGetLoopback_t, 1255
ByteLoopbackMode, 1255
ByteLoopbackMultiplier, 1255
unpack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 2103
unpack_wds_SLQSSSetLoopback
wds.h, 2103
unpack_wds_SLQSSet3GPPConfigItem

wds.h, 2101
unpack_wds_SLQSSetIPFamilyPreference
wds.h, 2101
unpack_wds_SLQSSetIPFamilyPreference_t, 1251
TlvResult, 1251
unpack_wds_SLQSSetPacketSrvStatusCallback
wds.h, 2101
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1251
bearerID, 1252
conn_status, 1252
ipFamily, 1252
reconfigReqd, 1252
sessionEndReason, 1252
techName, 1252
verboseSessnEndReason, 1252
verboseSessnEndReasonType, 1252
unpack_wds_SLQSSetWdsEventCallback
wds.h, 2102
unpack_wds_SLQSSetWdsEventCallback_ind
wds.h, 2102
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1252
currDBTechAvail, 1253
currNWInfo, 1253
dBTechAvail, 1253
dBTechnology, 1253
dBtechExtAvail, 1253
dBtechnologyExt, 1253
dataSysStatAvail, 1253
dormancyStatAvail, 1254
dormancyStatus, 1254
mipStatus, 1254
mipstatAvail, 1254
netInfoLen, 1254
prefNetwork, 1254
ratMask, 1254
rx_bytes, 1254
rx_pkts, 1254
soMask, 1254
tx_bytes, 1254
tx_pkts, 1254
xferStatAvail, 1254
unpack_wds_SLQSStartDataSession
wds.h, 2104
unpack_wds_SLQSStartDataSession_t, 1255
pFailureReason, 1256
pVerboseFailReasonType, 1256
pVerboseFailureReason, 1256
psid, 1256
unpack_wds_SLQSStopDataSession
wds.h, 2104
unpack_wds_SLQSWdsGoActive
wds.h, 2104
unpack_wds_SLQSWdsGoDormant
wds.h, 2105
unpack_wds_SLQSWdsSetEventReport
wds.h, 2105
unpack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 2105
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1256
apnName, 1257
bearerId, 1257
contextId, 1257
ipv4Address, 1257
ipv4GWAddress, 1257
ipv6Address, 1258
ipv6GWAddress, 1258
unpack_wds_SetAutoconnect
wds.h, 2095
unpack_wds_SetDefaultProfile
wds.h, 2096
unpack_wds_SetDefaultProfileNum
wds.h, 2096
unpack_wds_SetMobileIP
wds.h, 2096
unpack_wds_SetMobileIPParameters
wds.h, 2097
unpack_wds_SetMobileIPProfile
wds.h, 2097
unpack_wds_SetMobileIPProfile_t, 1240
UnpackQmiProfileInfo
wds.h, 2070
unpackWdsProfileParam, 1258
SlqsProfile3GPP, 1259
SlqsProfile3GPP2, 1259
upLink
NSSAudioCtrl, 579
UpdateCompleteStatus
unpack_swioma_SLQSOMADMGetSessionInfo_t, 1190
updateCompleteStatus
omaDmFotaTlv, 585
unpack_omaDmFotaTlv_t, 1142
upgrade_mc77xx_fw
qaGobiApiFms.h, 1700
UpgradeFirmware2k
qaGobiApiFms.h, 1700
upinRetries
slotInf, 868
slotInfo, 870
uim_slotInfo, 988
upinState
slotInf, 869
slotInfo, 870
uim_slotInfo, 988
UpkQmiCbkCatEventReportInd
qaCbkCatEventReportInd.h, 1543
UpkQmiCbkSwiOmaDmEventReportInd
qaCbkSwiOmaDmEventReportInd.h, 1544
UpkQmiCbkSwiOmaDmEventReportIndExt
qaCbkSwiOmaDmEventReportInd.h, 1544
UpkQmiNasGetRFBandInfo
qaNasGetRFBandInfo.h, 1964
UpkQmiNasPerformNetworkScan
qaNasPerformNetworkScan.h, 1965

upukRetries
slotInf, 869
slotInfo, 870
uim_slotInfo, 988
urlAddr, 1259
address, 1259
TlvPresent, 1259
usageMask
loc_sensorDataUsage, 372
sensorDataUsage_s, 832
t_sensor, 944
User Identity Module Service (UIM), 52
userData
SMSAsyncRawSend_s, 901
userInputReq
omaDmConfigTlv, 580
omaDmConfigTlvExt, 583
omaDmFotaTlv, 585
unpack_omaDmConfigTlv_t, 1140
unpack_omaDmFotaTlv_t, 1142
userInputTimeout
omaDmConfigTlv, 580
omaDmConfigTlvExt, 583
omaDmFotaTlv, 585
omaDmFotaTlvExt, 587
unpack_omaDmConfigTlv_t, 1140
unpack_omaDmFotaTlv_t, 1142
Username
unpack_wds_SLQSGetRuntimeSettings_t, 1250
username
unpack_wds_GetDefaultProfile_t, 1232
usersize
unpack_wds_GetDefaultProfile_t, 1232
ussDCS
pack_voice_SLQSOriginateUSSD_t, 687
USSInfo, 1264
voice_USSInfo, 1302
ussData
pack_voice_SLQSOriginateUSSD_t, 687
USSInfo, 1264
voice_USSInfo, 1301
ussLen
pack_voice_SLQSOriginateUSSD_t, 687
USSInfo, 1264
voice_USSInfo, 1302
uusInfo
allCallsUUSInfo, 118
voice_allCallsUUSInfo, 1268

v4sessionId
qaQmiInterfaceInfo, 765
ssdatasession_params, 921
WdsRunTimeSettings, 1395
v6sessionId
qaQmiInterfaceInfo, 765
ssdatasession_params, 921
WdsRunTimeSettings, 1395
VOICE_SUPS_SRV_CLASS_DATA
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_FAX
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_NONE
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_PACKETACCESS
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_PADACCESS
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_SMS
qaGobiApiVoice.h, 1903
VOICE_SUPS_SRV_CLASS_VOICE
qaGobiApiVoice.h, 1903
VDOP
loc_precisionDilution, 369
precisionDilution_s, 748
VOICE_SRV
qaGobiApiCbk.h, 1559
val
IPv6TrafCls, 332
Tos, 959
unpack_qos_IPv6TrafCls_t, 1157
unpack_qos_Tos_t, 1174
val_3GPP2Pri
unpack_qos_swiQosFlow_t, 1173
val_3GPPImCn
unpack_qos_swiQosFlow_t, 1173
val_3GPPResResidualBER
unpack_qos_swiQosFlow_t, 1173
val_3GPPSigInd
unpack_qos_swiQosFlow_t, 1173
val_3GPPTraHdlPri
unpack_qos_swiQosFlow_t, 1173
ValidMask
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
validMask
loc_satelliteInfo, 371
satelliteInfo, 827
ValidateSPC
qaGobiApiDms.h, 1681
ValidityCW0
LteCQIParm, 389
unpack_nas_SLQSSwiGetLteCQI_t, 1135
ValidityCW1
LteCQIParm, 389
unpack_nas_SLQSSwiGetLteCQI_t, 1135
Value
GetM2MSpkrGainResp, 278
SetM2MSpkrGainReq, 854
value
pack_dms_UIMSetPINProtection_t, 603
pack_dms_UIMVerifyPIN_t, 604
value_length
custSettingInfo, 208

DMScustSettingInfo, 227
pack_dms_SetCustFeaturesV2_t, 595
setCustomSettingV2, 844
verbFailReason
ssdatasession_params, 921
verbFailReasonType
ssdatasession_params, 921
verboseSessnEndReason
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1252
verboseSessnEndReasonType
_packetSrvStatus, 70
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1252
verifyLeft
personalizationStatus, 737
remainingRetries, 809
uim_personalizationStatus, 980
uim_remainingRetries, 984
verifyPIN
pack_uim_VerifyPin_t, 686
UIMVerifyPinReq, 1012
verifyRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 1065
unpack_dms_UIMSetControlKeyProtection_t, 1068
unpack_dms_UIMSetPINProtection_t, 1068
verifyUIMPIN, 1266
pinID, 1266
pinLen, 1266
pinVal, 1266
version
omaDmFotaTlv, 585
omaDmFotaTlvExt, 587
swiQosFilter, 935
unpack_omaDmFotaTlv_t, 1142
unpack_qos_swiQosFilter_t, 1169
versionString
pack_dms_SLQSSwiSetOSInfo_t, 600
unpack_dms_SLQSSwiGetOSInfo_t, 1062
versionlength
omaDmFotaTlv, 585
omaDmFotaTlvExt, 587
unpack_omaDmFotaTlv_t, 1142
vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 616
vertReliability
pack_loc_SLQSLOCInjectPosition_t, 616
vertUnc
pack_loc_SLQSLOCInjectPosition_t, 616
VerticalUncertainty
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
VirtStream
nas_protocolSubtypeElement, 491
protocolSubtypeElement, 763
Voice Service (VOICE), 50
voice.h
LITE_VOICE_SUPS_SRV_CLASS_DATA, 2041
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC, 2041
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC, 2041
LITE_VOICE_SUPS_SRV_CLASS_FAX, 2041
LITE_VOICE_SUPS_SRV_CLASS_NONE, 2041
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS, 2041
LITE_VOICE_SUPS_SRV_CLASS_PADACCESS, 2041
LITE_VOICE_SUPS_SRV_CLASS_SMS, 2041
LITE_VOICE_SUPS_SRV_CLASS_VOICE, 2041
voice.h, 2036
liteServiceClassInformation, 2041
MAXVOICEUSSDLENGTH, 2041
pack_voice_AnswerUSSD, 2041
pack_voice_CancelUSSD, 2041
pack_voice_OriginateUSSD, 2042
pack_voice_SLQSOriginateUSSD, 2042
pack_voice_SLQSVoiceALSSelectLine, 2042
pack_voice_SLQSVoiceALSSetLineSwitching, 2043
pack_voice_SLQSVoiceAnswerCall, 2043
pack_voice_SLQSVoiceBindSubscription, 2044
pack_voice_SLQSVoiceBurstDTMF, 2044
pack_voice_SLQSVoiceDialCall, 2044
pack_voice_SLQSVoiceEndCall, 2045
pack_voice_SLQSVoiceGetAllCallInfo, 2045
pack_voice_SLQSVoiceGetCLIP, 2047
pack_voice_SLQSVoiceGetCLIR, 2047
pack_voice_SLQSVoiceGetCNAP, 2048
pack_voice_SLQSVoiceGetCOLP, 2048
pack_voice_SLQSVoiceGetCOLR, 2048
pack_voice_SLQSVoiceGetCallBarring, 2045
pack_voice_SLQSVoiceGetCallForwardingStatus, 2046
pack_voice_SLQSVoiceGetCallInfo, 2046
pack_voice_SLQSVoiceGetCallWaiting, 2047
pack_voice_SLQSVoiceGetConfig, 2049
pack_voice_SLQSVoiceIndicationRegister, 2049
pack_voice_SLQSVoiceManageCalls, 2049
pack_voice_SLQSVoiceOrigUSSDNoWait, 2050
pack_voice_SLQSVoiceSendFlash, 2050
pack_voice_SLQSVoiceSetCallBarringPassword, 2050
pack_voice_SLQSVoiceSetConfig, 2051
pack_voice_SLQSVoiceSetPreferredPrivacy, 2051
pack_voice_SLQSVoiceSetSUPSService, 2052
pack_voice_SLQSVoiceStartContDTMF, 2052
pack_voice_SLQSVoiceStopContDTMF, 2052
unpack_voice_AnswerUSSD, 2053
unpack_voice_CancelUSSD, 2053
unpack_voice_OriginateUSSD, 2053
unpack_voice_SLQSOriginateUSSD, 2054
unpack_voice_SLQSVoiceALSSelectLine, 2054
unpack_voice_SLQSVoiceALSSetLineSwitching, 2054

unpack_voice_SLQSVoiceAnswerCall, 2055
unpack_voice_SLQSVoiceBindSubscription, 2055
unpack_voice_SLQSVoiceBurstDTMF, 2055
unpack_voice_SLQSVoiceDialCall, 2056
unpack_voice_SLQSVoiceEndCall, 2056
unpack_voice_SLQSVoiceGetAllCallInfo, 2056
unpack_voice_SLQSVoiceGetCLIP, 2058
unpack_voice_SLQSVoiceGetCLIR, 2059
unpack_voice_SLQSVoiceGetCNAP, 2059
unpack_voice_SLQSVoiceGetCOLP, 2059
unpack_voice_SLQSVoiceGetCOLR, 2060
unpack_voice_SLQSVoiceGetCallBarring, 2057
unpack_voice_SLQSVoiceGetCallForwarding-Status, 2057
unpack_voice_SLQSVoiceGetCallInfo, 2058
unpack_voice_SLQSVoiceGetCallWaiting, 2058
unpack_voice_SLQSVoiceGetConfig, 2060
unpack_voice_SLQSVoiceIndicationRegister, 2060
unpack_voice_SLQSVoiceManageCalls, 2061
unpack_voice_SLQSVoiceOrigUSSDNoWait, 2061
unpack_voice_SLQSVoiceSendFlash, 2061
unpack_voice_SLQSVoiceSetCallBarringPassword, 2062
unpack_voice_SLQSVoiceSetConfig, 2062
unpack_voice_SLQSVoiceSetPreferredPrivacy, 2062
unpack_voice_SLQSVoiceSetSUPSService, 2063
unpack_voice_SLQSVoiceStartContDTMF, 2063
unpack_voice_SLQSVoiceStopContDTMF, 2063
voice_CLIPResp, 1284
ActiveStatus, 1285
ProvisionStatus, 1285
voice_CLIRResp, 1285
ActiveStatus, 1286
ProvisionStatus, 1286
voice_CNAPResp, 1286
ActiveStatus, 1286
ProvisionStatus, 1286
voice_COLPResp, 1287
ActiveStatus, 1287
ProvisionStatus, 1287
voice_COLRResp, 1287
ActiveStatus, 1288
ProvisionStatus, 1288
voice_CUGInfo, 1290
CUGIndex, 1290
SuppOA, 1290
SuppPrefCUG, 1290
voice_DTMFLengths, 1292
DTMFInterdigitInterval, 1292
DTMFPulseWidth, 1292
voice_USSInfo, 1301
ussDCS, 1302
ussData, 1301
ussLen, 1302
voice_UUSInfo, 1302
UUSData, 1303
UUSDatalen, 1303
UUSDcs, 1303
UUSType, 1303
voice_airTimer, 1266
airTimerValue, 1267
namID, 1267
voice_allCallsAlphaIDInfo, 1267
AlphaIDInfo, 1267
callID, 1267
voice_allCallsDiagInfo, 1267
callID, 1268
DiagInfo, 1268
voice_allCallsUUSInfo, 1268
callID, 1268
uusInfo, 1268
voice_alphaIDInfo, 1268
alphaDcs, 1269
alphaLen, 1269
alphaText, 1269
voice_arrAlertingPattern, 1269
alertingPattern, 1270
callID, 1270
numInstances, 1270
voice_arrAlertingType, 1270
AlertingType, 1271
callID, 1271
numInstances, 1271
voice_arrAlphaID, 1271
allCallsAlphaIDInfoArr, 1271
numInstances, 1271
voice_arrCallEndReason, 1272
callEndReason, 1272
callID, 1272
numInstances, 1272
voice_arrCallInfo, 1272
getAllCallInfo, 1273
numInstances, 1273
voice_arrCalledPartyNum, 1271
CalledPartyNum, 1272
numInstances, 1272
voice_arrConnectPartyNum, 1273
ConnectedPartyNum, 1273
numInstances, 1273
voice_arrDiagInfo, 1274
DiagInfo, 1274
numInstances, 1274
voice_arrRedirPartyNum, 1274
numInstances, 1274
RedirPartyNum, 1275
voice_arrRemotePartyName, 1275
GetAllCallRmtPtyName, 1275
numInstances, 1275
voice_arrRemotePartyNum, 1275
numInstances, 1276
RmtPtyNum, 1276
voice_arrSvcOption, 1276
callID, 1276
numInstances, 1276
srvOption, 1276

voice_arrUUSInfo, 1276
AllCallsUUSInfo, 1277
numInstances, 1277
voice_burstDTMFInfo, 1277
digitCnt, 1277
pCallID, 1277
pDigitBuff, 1277
voice_callFWExtInfo, 1279
noReplyTimer, 1281
numLen, 1281
numPlan, 1281
numType, 1281
number, 1281
PI, 1281
SI, 1281
SvcClass, 1281
SvcStatus, 1281
voice_callFWInfo, 1281
noReplyTimer, 1282
numLen, 1282
number, 1282
SvcClass, 1282
SvcStatus, 1282
voice_callFwdTypeAndPlan, 1278
numberPlan, 1279
numberType, 1279
voice_callInfo, 1282
callID, 1283
callState, 1283
callType, 1284
direction, 1284
mode, 1284
voice_calledPartySubAdd, 1278
extBit, 1278
oddEvenInd, 1278
subAddr, 1278
subAddrLen, 1278
subAddrType, 1278
voice_ccSUPSType, 1284
reason, 1284
svcType, 1284
voice_connectNumInfo, 1288
callerID, 1289
callerIDLen, 1289
numPlan, 1289
numPresInd, 1289
numType, 1290
screeningInd, 1290
voice_curAMRConfig, 1290
gsmAmrStat, 1291
wcdmaAmrStat, 1291
voice_diagInfo, 1291
diagInfoLen, 1291
diagnosticInfo, 1291
voice_getAllCallInformation, 1292
ALS, 1293
Callinfo, 1293
isEmpty, 1293
voice_getAllCallRmtPtyName, 1293
callID, 1293
RemotePartyName, 1293
voice_getAllCallRmtPtyNum, 1293
callID, 1294
RemotePartyNum, 1294
voice_getCallFWExtInfo, 1294
CallFWExtInfo, 1294
numInstances, 1294
voice_getCallFWInfo, 1295
CallFWInfo, 1295
numInstances, 1295
voice_peerNumberInfo, 1295
callID, 1297
numLen, 1297
numPI, 1297
numPlan, 1297
numSI, 1297
numType, 1297
number, 1297
voice_prefVoiceSO, 1297
evrcCapability, 1299
homeOrigVoiceSO, 1299
homePageVoiceSO, 1299
namID, 1299
roamOrigVoiceSO, 1299
voice_remotePartyName, 1299
callerName, 1299
codingScheme, 1300
nameLen, 1300
namePI, 1300
voice_remotePartyNum, 1300
numLen, 1300
presentationInd, 1300
remPartyNumber, 1300
voice_roamTimer, 1300
namID, 1301
roamTimerValue, 1301
voiceALSSelectLineInfo, 1303
lineValue, 1303
voiceALSSetLineSwitchInfo, 1303
switchOption, 1303
voiceAnswerCall, 1304
pCallId, 1304
voiceBindSubscriptionInfo, 1304
subsType, 1304
voiceBurstDTMFInfo, 1304
BurstDTMFInfo, 1305
pBurstDTMFLengths, 1305
voiceCallInfoReq, 1305
callID, 1305
voiceCallInfoResp, 1305
pAlertType, 1308
pAlertingPattern, 1308
pAlphaIDInfo, 1308
pCallInfo, 1308
pConnectNumInfo, 1308
pDiagInfo, 1308

pOTASPStatus, 1308
pRemotePartyName, 1308
pRemotePartyNum, 1308
pSrvOpt, 1308
pUUSInfo, 1308
pVoicePrivacy, 1308
voiceCallRequestParams, 1308
callNumber, 1310
pCLIRType, 1310
pCUGInfo, 1310
pCallPartySubAdd, 1310
pCallType, 1310
pEmergencyCategory, 1310
pSvcType, 1310
pUUSInFo, 1310
voiceCallResponseParams, 1310
pAlphaIDInfo, 1311
pCCResultType, 1311
pCCSUPSType, 1311
pCallID, 1311
voiceContDTMFinfo, 1311
DTMFdigit, 1311
pCallID, 1311
voiceDTMFEventInfo, 1312
DTMFInformation, 1312
pOffLength, 1312
pOnLength, 1312
voiceFlashInfo, 1312
pCallID, 1313
pFlashPayLd, 1313
pFlashType, 1313
voiceGetAllCallInfo, 1313
pArrAlertingPattern, 1315
pArrAlertingType, 1315
pArrAlphaID, 1315
pArrCallEndReason, 1315
pArrCallInfo, 1315
pArrCalledPartyNum, 1315
pArrConnectPartyNum, 1315
pArrDiagInfo, 1315
pArrRedirPartyNum, 1315
pArrRemotePartyName, 1315
pArrRemotePartyNum, 1315
pArrSvcOption, 1315
pArrUUSInfo, 1316
pOTASPStatus, 1316
pVoicePrivacy, 1316
voiceGetCLIPResp, 1321
pAlphaIDInfo, 1322
pCCResType, 1322
pCCSUPSType, 1322
pCLIPResp, 1322
pCallID, 1322
pFailCause, 1322
voiceGetCLIRResp, 1322
pAlphaIDInfo, 1323
pCCResType, 1324
pCCSUPSType, 1324
pCLIRResp, 1324
pCallID, 1323
pFailCause, 1324
voiceGetCNAPResp, 1324
pAlphaIDInfo, 1325
pCCResType, 1325
pCCSUPSType, 1325
pCNAPResp, 1325
pCallID, 1325
pFailCause, 1325
voiceGetCOLPResp, 1325
pAlphaIDInfo, 1326
pCCResType, 1326
pCCSUPSType, 1326
pCOLPResp, 1326
pCallID, 1326
pFailCause, 1326
voiceGetCOLRResp, 1326
pAlphaIDInfo, 1327
pCCResType, 1327
pCCSUPSType, 1327
pCOLRResp, 1327
pCallID, 1327
pFailCause, 1327
voiceGetCallBarringReq, 1316
pSvcClass, 1316
reason, 1316
voiceGetCallBarringResp, 1316
pAlphaIDInfo, 1317
pCCResType, 1318
pCCSUPSType, 1318
pCallID, 1318
pFailCause, 1318
pSvcClass, 1318
voiceGetCallFWReq, 1318
pSvcClass, 1318
Reason, 1318
voiceGetCallFWResp, 1318
pAlphaIDInfo, 1320
pCCResType, 1320
pCCSUPSType, 1320
pCallID, 1320
pFailCause, 1320
pGetCallFWExtInfo, 1320
pGetCallFWInfo, 1320
voiceGetCallWaitInfo, 1320
pAlphaIDInfo, 1321
pCCResType, 1321
pCCSUPSType, 1321
pCallID, 1321
pFailCause, 1321
pSvcClass, 1321
voiceGetConfigReq, 1327
pAMRStatus, 1329
pAirTimer, 1329
pAutoAnswer, 1329
pNamID, 1329
pPrefVoicePrivacy, 1329

pPrefVoiceSO, 1329
pRoamTimer, 1329
pTTYMode, 1329
pVoiceDomainPref, 1329
voiceGetConfigResp, 1329
pAirTimerCnt, 1331
pAutoAnswerStat, 1331
pCurAMRConfig, 1331
pCurPrefVoiceSO, 1331
pCurVoiceDomainPref, 1331
pCurVoicePrivacyPref, 1331
pCurrTTYMode, 1331
pRoamTimerCnt, 1331
voiceIndicationRegisterInfo, 1331
pRegDTMFEvents, 1332
pRegVoicePrivacyEvents, 1332
pSuppsNotifEvents, 1332
voiceInfoRec, 1332
callID, 1334
pCLIRCause, 1334
pCallWaitInd, 1334
pCalledPartyInfo, 1334
pCallerIDInfo, 1334
pCallerNameInfo, 1334
pCallingPartyInfo, 1334
pConnectNumInfo, 1334
pDispInfo, 1334
pExtDispInfo, 1334
pExtDispRecInfo, 1334
pLineCtrlInfo, 1334
pNSSAudioCtrl, 1334
pNSSRelease, 1334
pRedirNumInfo, 1334
pSignalInfo, 1334
voiceManageCallsReq, 1334
pCallID, 1335
SUPSType, 1335
voiceManageCallsResp, 1335
pFailCause, 1336
VoiceNumber
unpack_dms_GetVoiceNumber_t, 1046
voiceNumberSize
unpack_dms_GetVoiceNumber_t, 1046
voiceOTASPStatusInfo, 1336
callID, 1337
OTASPStatus, 1337
voiceOrigUSSDNoWaitInfo, 1336
USSInformation, 1336
voicePrivacy
voicePrivacyInfo, 1338
voicePrivacyInfo, 1337
callID, 1338
voicePrivacy, 1338
voiceSUPSInfo, 1350
pAlphaIDInfo, 1352
pCLIPstatus, 1352
pCLIRstatus, 1352
pCNAPstatus, 1352
pCOLPstatus, 1352
pCOLRstatus, 1352
pCallBarPasswd, 1352
pCallFWNum, 1352
pCallFWTimerVal, 1352
pCallFwdInfo, 1352
pCallID, 1352
pDataSrc, 1352
pFailCause, 1352
pNewPwdData, 1352
pReason, 1352
pSvcClass, 1352
pUSSInfo, 1352
SUPSInformation, 1352
voiceSUPSNotification, 1352
callID, 1354
notifType, 1354
pCUGIndex, 1354
pECTNum, 1354
voiceSetAllCallStatusCbkInfo, 1338
arrCallInfomation, 1340
pArrAlertingPattern, 1340
pArrAlertingType, 1340
pArrAlphaID, 1340
pArrCallEndReason, 1340
pArrCalledPartyNum, 1340
pArrConnectPartyNum, 1340
pArrDiagInfo, 1340
pArrRedirPartyNum, 1340
pArrRemotePartyName, 1340
pArrRemotePartyNum, 1340
pArrSvcOption, 1340
voiceSetCallBarringPwdInfo, 1340
newPasswd, 1341
newPasswdAgain, 1341
oldPasswd, 1341
Reason, 1341
voiceSetCallBarringPwdResp, 1341
pAlphaIDInfo, 1342
pCCResType, 1342
pCCSUPSType, 1342
pCallID, 1342
pFailCause, 1342
voiceSetConfigReq, 1342
pAirTimerConfig, 1343
pAutoAnswer, 1344
pPrefVoiceDomain, 1344
pPrefVoiceSO, 1344
pRoamTimerConfig, 1344
pTTYMode, 1344
voiceSetConfigResp, 1344
pAirTimerStatus, 1345
pAutoAnsStatus, 1345
pPrefVoiceSOStatus, 1345
pRoamTimerStatus, 1345
pTTYConfigStatus, 1345
pVoiceDomainPrefStatus, 1345
voiceSetPrefPrivacy, 1345

privacyPref, 1346
voiceSetSUPSServiceReq, 1346
pCallBarringPasswd, 1348
pCallForwardingNumber, 1348
pCallFwdTypeAndPlan, 1348
pServiceClass, 1348
pTimerVal, 1348
reason, 1348
voiceSvc, 1348
voiceSetSUPSServiceResp, 1348
pAlphaIDInfo, 1349
pCCResultType, 1349
pCCSUPSType, 1349
pCallID, 1349
pFailCause, 1349
voiceStopContDTMFinfo, 1349
callID, 1350
voiceSvc
pack_voice_SLQSVoiceSetSUPSService_t, 703
voiceSetSUPSServiceReq, 1348
VolValue
SetAudioVolTLBConfigReq, 843
Volume
GetAudioProfileResp, 258
GetAudioVolTLBConfigReq, 259
GetM2MAudioProfileResp, 275
SetAudioProfileReq, 842
SetAudioVolTLBConfigReq, 843
voteForInit
registerRefresh, 809
uim_registerRefresh, 984

WAMSParaChanged
_SwiAvmsEventInd, 100
wAlertMsgLength
_QmiSwiAvmsEventReportConfigResp, 74
WCDMACellInfo
lteWcdmaCellInfo, 409
nas_lteWcdmaCellInfo, 471
WCDMAECIOThresh, 1355
pWCDMAECIOThreshList, 1355
WCDMAECIOThreshListLen, 1355
WCDMAECIOThreshListLen
nas_WCDMAECIOThresh, 523
WCDMAECIOThresh, 1355
WCDMAInfoLTENeighborCell, 1355
UMTSLTENbrCell, 1356
umtsLTENbrCellLen, 1356
wcdmaRRCState, 1356
WCDMARSSIThresh, 1360
pWCDMARSSIThreshList, 1361
WCDMARSSIThreshListLen, 1361
WCDMARSSIThreshListLen
nas_WCDMARSSIThresh, 525
WCDMARSSIThresh, 1361
WCDMASSInfo
unpack_nas_SLQSNasGetSigInfo_t, 1127
WCDMASysInfo, 1361
cellId, 1363
cellIdValid, 1364
hsCallStatus, 1364
hsCallStatusValid, 1364
hsInd, 1364
hsIndValid, 1364
lac, 1364
lacValid, 1364
MCC, 1364
MNC, 1364
networkIdValid, 1364
psc, 1364
pscValid, 1364
regRejectInfoValid, 1364
rejCause, 1364
rejectSrvDomain, 1364
sysInfoWCDMA, 1364
WDS_IsGobiDevice
qaGobiApiWds.h, 1963
WDS_SRV
qaGobiApiCbk.h, 1559
WDSGetLoopbackData, 1389
ByteLoopbackMode, 1390
ByteLoopbackMultiplier, 1390
WDSSWICurrentChannelRates, 1397
current_channel_rx_rate, 1398
current_channel_tx_rate, 1398
max_channel_rx_rate, 1398
max_channel_tx_rate, 1398
WDSSetLoopbackData, 1397
pLoopbackMode, 1397
pLoopbackMultiplier, 1397
wDescriptionLength
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
wErrorCode
_QmiSwiAvmsEventReportDataSessionStatus-Resp, 75
wHTTPStatus
_QmiSwiAvmsEventReportHTTPStatusResp, 76
wNameLength
_QmiSwiAvmsEventReportBinaryUpdateSession-InfoResp, 73
WORD
SwiDataTypes.h, 2007
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, 660
unpack_rms_GetSMSWake_t, 1175
wcdmaAmrStat
curAMRConfig, 196
voice_curAMRConfig, 1291
wcdmaCellInfo, 1354
cpich_ecno, 1354
cpich_rscp, 1354
psc, 1354
srxlev, 1355
wcdmaLongMsgDecodingParams, 1356
Date, 1357
pIsUDHPresent, 1357
pMessage, 1357
pPartNum, 1358
pReferenceNum, 1358
pScAddr, 1358
pScAddrLength, 1358
pSenderAddr, 1358
pSenderAddrLength, 1358
pTextMsg, 1358
pTextMsgLength, 1358
pTotalNum, 1358
Time, 1358
wcdmaMsgDecodingParams, 1358
Date, 1359
pMessage, 1359
pScAddr, 1359
pScAddrLength, 1359
pSenderAddr, 1359
pSenderAddrLength, 1359
pTextMsg, 1359
pTextMsgLength, 1359
Time, 1359
wcdmaMsgEncodingParams, 1359
alphabet, 1360
messageSize, 1360
pDestAddr, 1360
pPDUMessage, 1360
pTextMsg, 1360
wcdmaRRCState
nas_WCDMAInfoLTENeighborCell, 524
WCDMAInfoLTENeighborCell, 1356
wcdmaUARFCN, 1364
status, 1365
uarfcn, 1365
wds.h
QMI_LITE_WDS_CURRENT_CALL_DB_MASK, 2070
QMI_LITE_WDS_LAST_CALL_DB_MASK, 2070
wds.h, 2064
BYT_STAT_STAT_MASK, 2069
liteQmiDataBearerMasks, 2070
PACK_WDS_IPV4, 2069
PACK_WDS_IPV6, 2069
pack_wds_DHCPv4ClientLeaseChange, 2070
pack_wds_GetAutoconnect, 2070
pack_wds_GetByteTotals, 2071
pack_wds_GetConnectionRate, 2071
pack_wds_GetDataBearerTechnology, 2072
pack_wds_GetDefaultProfile, 2072
pack_wds_GetDefaultProfileNum, 2072
pack_wds_GetDormancyState, 2073
pack_wds_GetLastMobileIPError, 2073
pack_wds_GetMobileIP, 2074
pack_wds_GetMobileIPProfile, 2074
pack_wds_GetPacketStatistics, 2075
pack_wds_GetPacketStatus, 2075
pack_wds_GetSessionDuration, 2075
pack_wds_GetSessionState, 2076
pack_wds_RMSetTransferStatistics, 2076
pack_wds_SLQSCreateProfile, 2079
pack_wds_SLQSDeleteProfile, 2080
pack_wds_SLQSGet3GPPConfigItem, 2080
pack_wds_SLQSGetCurrDataSystemStat, 2080
pack_wds_SLQSGetCurrentChannelRate, 2081
pack_wds_SLQSGetDUNCallInfo, 2082
pack_wds_SLQSGetDataBearerTechnology, 2081
pack_wds_SLQSGetProfileSettings, 2082
pack_wds_SLQSGetRuntimeSettings, 2083
pack_wds_SLQSModifyProfile, 2083
pack_wds_SLQSResetPacketStatics, 2083
pack_wds_SLQSSGetDHCPv4ClientConfig, 2085
pack_wds_SLQSSGetLoopback, 2085
pack_wds_SLQSSSetDHCPv4ClientConfig, 2086
pack_wds_SLQSSSetLoopback, 2086
pack_wds_SLQSSet3GPPConfigItem, 2084
pack_wds_SLQSSetIPFamilyPreference, 2084
pack_wds_SLQSSetWdsEventCallback, 2085
pack_wds_SLQSStartDataSession, 2087
pack_wds_SLQSStopDataSession, 2087
pack_wds_SLQSWdsGoActive, 2087
pack_wds_SLQSWdsGoDormant, 2088
pack_wds_SLQSWdsSetEventReport, 2088
pack_wds_SLQSWdsSwiPDPRuntimeSettings, 2089
pack_wds_SetAutoconnect, 2077
pack_wds_SetDefaultProfile, 2077
pack_wds_SetDefaultProfileNum, 2077
pack_wds_SetMobileIP, 2078
pack_wds_SetMobileIPParameters, 2078
pack_wds_SetMobileIPProfile, 2079
unpack_wds_DHCPv4ClientLease_ind, 2089
unpack_wds_DHCPv4ClientLeaseChange, 2089
unpack_wds_GetAutoconnect, 2090
unpack_wds_GetByteTotals, 2090
unpack_wds_GetConnectionRate, 2090
unpack_wds_GetDataBearerTechnology, 2091
unpack_wds_GetDefaultProfile, 2091
unpack_wds_GetDefaultProfileNum, 2091
unpack_wds_GetDormancyState, 2092
unpack_wds_GetLastMobileIPError, 2092

unpack_wds_GetMobileIP, 2093
unpack_wds_GetMobileIPProfile, 2093
unpack_wds_GetPacketStatistics, 2093
unpack_wds_GetPacketStatus, 2094
unpack_wds_GetSessionDuration, 2094
unpack_wds_GetSessionState, 2094
unpack_wds_RMSetTransferStatistics, 2095
unpack_wds_RMTransferStatistics_ind, 2095
unpack_wds_SLQSCreateProfile, 2097
unpack_wds_SLQSDeleteProfile, 2098
unpack_wds_SLQSGet3GPPConfigItem, 2098
unpack_wds_SLQSGetCurrDataSystemStat, 2098
unpack_wds_SLQSGetCurrentChannelRate, 2099
unpack_wds_SLQSGetDUNCallInfo, 2099
unpack_wds_SLQSGetDataBearerTechnology, 2099
unpack_wds_SLQSGetProfileSettings, 2100
unpack_wds_SLQSGetRuntimeSettings, 2100
unpack_wds_SLQSModifyProfile, 2100
unpack_wds_SLQSResetPacketStatics, 2101
unpack_wds_SLQSSGetDHCPv4ClientConfig, 2102
unpack_wds_SLQSSGetLoopback, 2103
unpack_wds_SLQSSSetDHCPv4ClientConfig, 2103
unpack_wds_SLQSSSetLoopback, 2103
unpack_wds_SLQSSet3GPPConfigItem, 2101
unpack_wds_SLQSSetIPFamilyPreference, 2101
unpack_wds_SLQSSetPacketSrvStatusCallback, 2101
unpack_wds_SLQSSetWdsEventCallback, 2102
unpack_wds_SLQSSetWdsEventCallback_ind, 2102
unpack_wds_SLQSStartDataSession, 2104
unpack_wds_SLQSStopDataSession, 2104
unpack_wds_SLQSWdsGoActive, 2104
unpack_wds_SLQSWdsGoDormant, 2105
unpack_wds_SLQSWdsSetEventReport, 2105
unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 2105
unpack_wds_SetAutoconnect, 2095
unpack_wds_SetDefaultProfile, 2096
unpack_wds_SetDefaultProfileNum, 2096
unpack_wds_SetMobileIP, 2096
unpack_wds_SetMobileIPParameters, 2097
unpack_wds_SetMobileIPProfile, 2097
UnpackQmiProfileInfo, 2070
wds_DHCPLeaseOptTlv, 1367
numOpt, 1367
optList, 1367
optListData, 1367
TlvPresent, 1367
wds_DHCPLeaseStateTlv, 1368
leaseState, 1368
TlvPresent, 1368
wds_DHCPOpt, 1368
optCode, 1369
optValLen, 1369
pOptVal, 1369
wds_DHCPProfileIdTlv, 1369
profileId, 1369
profileType, 1369
TlvPresent, 1369
wds_DHCPv4HWConfig, 1369
chaddr, 1370
chaddrLen, 1370
hwType, 1370
wds_DHCPv4Option, 1370
optCode, 1370
optVal, 1370
optValLen, 1370
wds_DHCPv4OptionList, 1371
numOpt, 1371
pOptList, 1371
wds_DHCPv4ProfileId, 1371
profileId, 1371
profileType, 1371
wds_DataULongLongTlv, 1366
TlvPresent, 1366
ullData, 1366
wds_DataULongTlv, 1367
TlvPresent, 1367
ulData, 1367
wds_Domain, 1372
domainLen, 1372
domainName, 1372
wds_DomainNameList, 1372
domain, 1372
numInstances, 1372
wds_GPRSQoS, 1373
delayClass, 1373
meanThroughputClass, 1373
peakThroughputClass, 1373
precedenceClass, 1373
reliabilityClass, 1373
wds_IPV6AddressInfo, 1374
IPAddressV6, 1374
IPV6PrefixLen, 1374
wds_IPV6GWAddressInfo, 1374
gwAddressV6, 1375
gwV6PrefixLen, 1375
wds_IPv4AdTlv, 1373
IPv4Addr, 1374
TlvPresent, 1374
wds_PCSCFFQDNAddress, 1375
fqdnAddr, 1375
fqdnLen, 1375
wds_PCSCFFQDNAddressList, 1375
numInstances, 1376
pcsfFQDNAddress, 1376
wds_PCSCFIPv4ServerAddressList, 1376
numInstances, 1376
pscsfIPv4Addr, 1376
wds_ProfileIdentifier, 1376
profileIndex, 1377
profileType, 1377

wds_TrStatInd, 1377
statsMask, 1378
statsPeriod, 1378
wds_UMTSMinQoS, 1378
deliveryErrSDU, 1380
grntDownlinkBitrate, 1380
grntUplinkBitrate, 1380
maxDownlinkBitrate, 1380
maxSDUSize, 1380
maxUplinkBitrate, 1380
qosDeliveryOrder, 1380
resBerRatio, 1380
sduErrorRatio, 1380
trafficClass, 1380
trafficPriority, 1380
transferDelay, 1380
wds_currNetworkInfo, 1365
NetworkType, 1366
RATMask, 1366
SOMask, 1366
wds_profileInfo, 1377
SlqsProfile3GPP, 1377
SlqsProfile3GPP2, 1377
WdsByteTotals, 1380
ByteTotalsElmntsV4, 1381
ByteTotalsElmntsV6, 1381
pV4sessionId, 1381
pV6sessionId, 1381
WdsByteTotalsElmnts, 1381
pRXTotalBytes, 1382
pTXTotalBytes, 1382
WdsClientLeaseChange, 1382
pEnableNotification, 1382
WdsConnectionRate, 1382
ConnRateElmntsV4, 1383
ConnRateElmntsV6, 1383
pV4sessionId, 1383
pV6sessionId, 1383
WdsConnectionRateElmnts, 1383
pCurrentChannelRXRate, 1384
pCurrentChannelTXRate, 1384
pMaxChannelRXRate, 1384
pMaxChannelTXRate, 1384
WdsDHCPv4ClientLeaseInd, 1384
pIPv4Addr, 1384
pLeaseState, 1384
pOptList, 1384
pProfileId, 1385
WdsDHCPv4Config, 1385
pHwConfig, 1385
pProfileId, 1385
pRequestOptionList, 1385
WdsDHCPv4HWConfig, 1386
chaddr, 1386
chaddrLen, 1386
hwType, 1386
WdsDHCPv4Option, 1386
optCode, 1387
optVal, 1387
optValLen, 1387
WdsDHCPv4OptionList, 1388
numOpt, 1388
pOptList, 1388
WdsDHCPv4ProfileId, 1389
profileId, 1389
profileType, 1389
wdsDhcpv4HwConfig, 1385
chaddr, 1386
chaddrLen, 1386
hwType, 1386
wdsDhcpv4Option, 1387
optCode, 1387
optVal, 1387
optValLen, 1387
wdsDhcpv4OptionList, 1388
numOpt, 1388
pOptList, 1388
wdsDhcpv4ProfileId, 1388
profileId, 1389
profileType, 1389
WdsIpAddressInfoReq, 1390
ip, 1391
pv4sessionId, 1391
pv6sessionId, 1391
WdsPktStatisticsElmnts, 1391
pRXDroppedCount, 1392
pRXOKBytesLastCall, 1392
pRXOkBytesCount, 1392
pRXPacketErrors, 1392
pRXPacketOverflows, 1392
pRXPacketSuccesses, 1392
pTXDroppedCount, 1392
pTXOKBytesLastCall, 1392
pTXOkBytesCount, 1392
pTXPacketErrors, 1392
pTXPacketOverflows, 1392
pTXPacketSuccesses, 1392
WdsPktStatisticsReq, 1393
pStatMask, 1393
WdsPktStatisticsResp, 1393
pV4sessionId, 1394
pV6sessionId, 1394
PktStatElmntsV4, 1394
PktStatElmntsV6, 1394
WdsProfileParam, 1394
SlqsProfile3GPP, 1394
SlqsProfile3GPP2, 1394
WdsRunTimeSettings, 1394
rts, 1395
v4sessionId, 1395
v6sessionId, 1395
wdsSetEventReportReq, 1395
pCurrChannelRateInd, 1396
pCurrDataBearerTechInd, 1396
pCurrPrefDataSysInd, 1396
pDataBearerTechInd, 1396

pDataCallStatusChangeInd, 1397
pDataSystemStatusChangeInd, 1397
pDormancyStatusInd, 1397
pEVDOPageMonPerChangeInd, 1397
pMIPStatusInd, 1397
pTransferStatInd, 1397
Wireless Data Service (WDS), 38

xAxis
sensorData, 830
sensorData_t, 831
xferStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 1254
xid
pack_loc_SLQSLOCGetBestAvailPos_t, 610
pack_qmi_t, 658
unpack_qmi_t, 1155
xtra_start_gps_minutes
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
xtra_start_gps_week
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154
xtra_valid_duration_hours
GPSStateInfo, 290
unpack_pds_SLQSGetGPSStateInfo_t, 1154

yAxis
sensorData, 830
sensorData_t, 831
year
nas_timeInfo, 516
nas_UniversalTime, 522
timeInfo, 955
UniversalTime, 1023

zAxis
sensorData, 830
sensorData_t, 831